On the Exact Rate-Memory Trade-off for Multi-access Coded Caching with Uncoded Placement

Kota Srinivas Reddy and Nikhil Karamchandani
Department of Electrical Engineering
Indian Institute of Technology, Bombay
Email: ksreddy@ee.iitb.ac.in, nikhilk@ee.iitb.ac.in

Abstract—We study a cache-aided content delivery network consisting of a central server which hosts a catalog of $N$ files, and $K$ caches each with limited memory $M$ which store content related to the files. There are $K$ users, each of which requests a file from the catalog, and has access to the data stored in $L \geq 1$ neighboring caches (with a cyclic wrap-around). The server transmits a common message to all the users, so that each of them can recover their requested file. This setup was recently studied in [1], where a coloring-based placement and coded-delivery policy was proposed and the required server transmission size was shown to be order-optimal with respect to information-theoretic bounds. We propose an alternate index coding-based placement and delivery scheme for this setup, which performs better than the previously proposed strategy. Furthermore, for multiple special cases including the $(N, K \leq 4, L)$ and $(N, K, L = K - 1)$-setups, we show that the scheme is exactly optimal under the restriction of uncoded placement. This extends other recent work [2], [3] which studies exact optimality for the single cache-access case ($L = 1$) to the multi cache-access case ($L > 1$).

I. INTRODUCTION

The rapid increase in usage of smart devices has lead to an unprecedented growth in internet traffic. A recent study [4] shows that data traffic from Video on Demand (VoD) services will increase exponentially in the forthcoming years. One way to meet this rise in demand is by prefetching and caching some of the data locally. Motivated by this, we study a cache-aided content delivery network (CCDN) as shown in Figure 1.

We consider a CCDN which consists of a central server which stores the entire content catalog, users which demand files from the catalog, and local caches with limited storage capabilities. In each time-slot, users reveal their demands and are served by providing them access to a subset of the caches and a common transmitted message from the central server. In this paper, we assume that each user is connected to $L$ neighboring caches with a cyclic wrap-around as shown in Figure 1. This model has been studied previously in [1], [5], [6] and is motivated by the possibility of enabling caches in small cells of the emerging heterogeneous wireless networks (HetNet) architecture for 5G cellular networks. In this CCDN model, the main challenges are (i) what to store in each cache (Placement Policy), and (ii) how to serve the user demands (Delivery Policy), with the goal of minimizing the server transmission rate.

The seminal work of [5], [7] studied a setup similar to the one shown in Figure 1, with $L = 1$, i.e., each user has access to a unique cache. They proposed a (uncoded) placement and (coded) delivery policy for this setup, and demonstrated that the required server transmission rate of their scheme is order-optimal with respect to information-theoretic lower bounds, which assume no restriction on either the placement or the delivery policy employed. Recent work [2], [3] has exploited connections of this problem to the well-studied index coding problem [8] to derive better lower bounds under the assumption of uncoded placement, and in fact proved that the policy proposed in [5] is exactly optimal (and not just order-optimal) amongst all schemes which use uncoded placement.

[1] proposed the general setting in Figure 1 and devised a coloring-based placement and delivery policy, for which they showed that the server transmission rate is orderwise optimal with respect to the information-theoretic lower bound. In this paper, we propose a new index coding-based delivery policy which achieves a lower server transmission rate than the scheme proposed in [1]. Furthermore, in the same spirit as [2], [3], we derive index coding-based lower bounds for
several example setups and show that our proposed scheme is exactly optimal amongst all uncoded placement policies for these setups. We present some generalizations of these results and also discuss the challenges involved in further extensions.

### II. SETTING

We study a cache system consisting of a central server, $K$ caches, and $N$ users, see Figure 1. We assume

- the central server stores $N$ files $F_1, F_2, ..., F_N$.
- each cache can store $M$ units of data.
- each user is connected to $L$ caches with a cyclic wraparound (for symmetry) as shown in Figure 1, and has access to the data stored in those caches.
- each user requests one out of the $N$ files, and
- there is an error-free shared communication link between the central server and the users.

The system operates in two phases: a placement phase, and a delivery phase as described below.

1) Placement Phase: In the placement phase, each cache (Cache $i$) stores content $\{Z_i = \phi_i(F_1, F_2, ..., F_N)\}$ related to the $N$ files. We assume uncoded placement phase i.e., we are allowed to split the files into parts and store the file parts, but coding across the file parts is not allowed. A major advantage of uncoded storage is that it can handle asynchronous demands without increasing the communication rate [2]. The placement phase occurs before the users reveal their requests.

After the placement phase, each user (User $j$) reveals the index of its requested file $(d_j)$, chosen arbitrarily from $(1, 2, ..., N)$. We will refer to $(d_1, d_2, ..., d_K)$ as the the request profile.

2) Delivery Phase: In the delivery phase, depending on the request profile $(d_1, d_2, ..., d_K)$ and the contents of the caches $(Z_1, Z_2, ..., Z_K)$, the central server broadcasts a coded message $X_{d_1, d_2, ..., d_K}$ of size $R$ units. By using the broadcasted message $X_{d_1, d_2, ..., d_K}$ and the cache content at the accessible caches $(Z_{d_1+1}, Z_{d_2+1}, ..., Z_{d_K+1})$, each user (User $j$) aims to reconstruct its requested file $(F_{d_j})$.

A memory-rate pair $(M, R)$ is said to be achievable for a system with cache memory $M$, if every user is able to recover its requested file for any valid request profile. Finally, we define $R^*(M)$ (optimal rate-memory trade-off) as the smallest rate $R$ such that $(M, R)$ is achievable.

As mentioned in Section II, [1] studied this setup with multi-cache access ($L > 1$) and proposed a coloring-based achievability scheme which builds on the coded delivery ideas presented in [5], [7] for single cache access ($L = 1$). For the general setup\(^2\), the server transmission rate $R(M)$ for this scheme is given by

$$R(M) = \frac{K - \frac{KLM}{N}}{1 + \frac{KM}{N}}, \quad M \in \left\{0, \frac{N}{K}, \frac{2N}{K}, ..., \frac{N}{L}\right\}.$$ 

For general $0 \leq M \leq N/L$, the lower convex envelope of these points is achievable via memory-sharing. [1] proved that the above transmission rate is order-wise optimal with respect to the information-theoretic lower bound. Incidentally, the proposed scheme used uncoded placement and coding is used only in the server broadcast message.

**Goal:** In this work, our goal is to study the optimal rate-memory trade-off $(R^*(M))$ under the restriction of uncoded placement. Given recent results which use lower bounds on the related index coding problem to exactly characterize $R^*(M)$ under this restriction for the single-cache access case ($L = 1$), we aim to extend this methodology to the general case of $L > 1$.

The paper is organized as follows. Section III presents some preliminaries on the index coding problem. Section IV discusses the $(N = 4, K = 4, L = 2)$-CCDN setup in detail and derive $R^*(M)$ under the restriction of uncoded placement in this case. This involves deriving improved achievable rates and lower bounds, as compared to those available from [1]. Section V extends the scheme to the general $(N, K, L)$ setup, presents several scenarios where the scheme is optimal, and also briefly discusses the challenges involved in characterizing the optimal rate-memory trade-off for the general setup.

### III. PRELIMINARIES

Our results are based on a mapping of our setup to that of the well-studied index coding problem [8]. As in our setup, the index coding problem consists of a server hosting a catalogue of say $n$ files. There are $n$ users, where each user $i$ wants a different file, say File $i$, and has access to a subset $J_i$ of the remaining files. Given the request profile and the side information profile, the server broadcasts a message to all the users so that each user is able to recover its requested file. The goal is to characterize the minimum required size of the server transmission for any given instance of the index coding problem. We will use the intuition that if we fix the user demands and the cache contents in our CCDN model, then the problem of minimizing the server transmission size corresponds to an instance of the index coding problem.

The $n$ user-index coding problem can be equivalently represented by a directed graph $G$ with $n$ nodes, where each node represents a unique user request, and an edge from Node $i$ to Node $j$ exists if user $i$ has access to File $j$. The following lemma from [9] provides a lower bound on the server transmission size as a function of this side information graph $G$.

**Lemma 1:** Consider an $n$ user-index coding problem [8] represented by the side information graph $G$. Let $M_i$ be the file requested by User $i$ and let $S(G)$ be the optimal server transmission size. Then, for any subset $J \subseteq \{1, 2, ..., n\}$ such that the subgraph of $G$ induced by the vertices in $J$ does not contain a directed cycle, we have

$$S(G) \geq \sum_{i \in J} |M_i|,$$

where $|M_i|$ indicates the size of file $M_i$.

As mentioned before, recent work [2], [3] has considered the special case of the CCDN model described in Section II cor-
responding to single cache access \((L = 1)\). The above lemma was a main ingredient there in the derivation of the optimal server transmission rate under uncoded placement. Following along similar lines, we will employ the above lemma to derive lower bounds on the optimal server transmission rate required for our CCDN setup with multi-cache access \((L > 1)\).

IV. \((N = 4, K = 4, L = 2)\)-CCDN

Consider an example with \(N = 4\) files, \(K = 4\) users / caches and each user connected to \(L = 2\) caches, as shown in Figure 2. We explain the proofs for this setup in detail, the remaining ones are discussed in short.

**Achievability:** We discuss the achievability for 3 memory points \(M = \{0, 1, 2\}\). At the remaining points, achievable rate is obtained by memory sharing. Let the 4 files be denoted by \(F_1 = A, F_2 = B, F_3 = C\), and \(F_4 = D\).

- \(M = 0\)
  \[ R(0) = 4 \text{ units}. \]
- \(M = 1\)

**Placement Phase:**
Divide each file into 4 parts and store the 1st part of each file in Cache 1, the 2nd part of each file in Cache 2, and so on as shown in Figure 2. Observe that the memory constraint \((M = 1)\) is satisfied. Since the 1st cache is connected to User 1 and User 4, we subscript the stored content in Cache 1 with \([4, 1]\), and repeat the same procedure for the other caches as well, i.e.,
- Cache 1 stores \(A_{(1,1)}, B_{(1,1)}, C_{(1,1)}\), and \(D_{(1,1)}\).
- Cache 2 stores \(A_{(1,2)}, B_{(1,2)}, C_{(1,2)}\), and \(D_{(1,2)}\).
- Cache 3 stores \(A_{(2,3)}, B_{(2,3)}, C_{(2,3)}\), and \(D_{(2,3)}\).
- Cache 4 stores \(A_{(3,4)}, B_{(3,4)}, C_{(3,4)}\), and \(D_{(3,4)}\).

**Delivery Phase:**
Let the user request profile be \(\{A, B, C, D\}\). In terms of subfiles,
- User 1 needs \(A_{(1,1)}, A_{(1,2)}, A_{(2,3)}\), and \(A_{(3,4)}\).
- User 2 needs \(B_{(1,1)}, B_{(1,2)}, B_{(2,3)}\), and \(B_{(3,4)}\).
- User 3 needs \(C_{(2,3)}, C_{(3,4)}, C_{(1,2)}\), and \(C_{(1,2)}\).
- User 4 needs \(D_{(3,4)}, D_{(4,1)}, D_{(1,2)}\), and \(D_{(2,3)}\).

Note that the red color (bold font) subfiles are already available at the corresponding users, only the black color (normal font) subfiles are needed for them. Each user requires 2 subfiles and thus a total 8 subfiles are involved in the server transmission. We can map the problem here to an instance of the index coding problem described in Section III, with \(n = 8\) virtual nodes or virtual users, each one requesting a distinct subfile. The side information at the virtual user representing (and requesting) some Subfile \(i\) are the subfiles available to the (real) user which is requesting Subfile \(i\). For example, the side information of the virtual node representing Subfile \(A_{(3,4)}\) are the subfiles available to the User 1, i.e., \(B_{(4,1)}, C_{(4,1)}, C_{(1,2)}\), and \(D_{(1,2)}\). We can solve this index coding problem to get the achievable transmission rate for our proposed scheme. In this case, it is possible to get an explicit characterization of the optimal server transmission scheme and given by:

\[
\begin{align*}
A_{(2,3)} & \oplus B_{(4,1)} \oplus C_{(1,2)} \oplus D_{(1,2)}, \\
B_{(3,4)} & \oplus C_{(1,2)} \oplus D_{(2,3)} \oplus A_{(2,3)}, \\
C_{(4,1)} & \oplus D_{(2,3)} \oplus A_{(3,4)} \oplus B_{(3,4)},
\end{align*}
\]

It can be verified (see [10] for details) that using the above server transmissions and the accessible cache contents, each user can recover its requested file. Also, while we have illustrated the delivery scheme for the request profile \(\{A, B, C, D\}\), a similar scheme works for any other request pattern say \(\{X_1, X_2, X_3, X_4\}\) by substituting \(A = X_1, B = X_2, C = X_3, D = X_4\) in equations (1),(2), and (3). Since each message is of size 1/4 units, the total server transmission size is given by \(R(1) = 3/4\) units.

- \(M = 2\)
  Store files \(A, B\) in Cache 1, Cache 3 and \(C, D\) in Cache 2, Cache 4. Since, each user has access to all files, the worst case transmission rate is \(R(2) = 0\) units.

The transmission rate \(R(M)\) at intermediate values is given by memory-sharing and thus the achievable rate-memory trade-off of our scheme is given by

\[
R^*(M) \leq R(M) = \begin{cases} 
4 - 13M/4, & \text{if } 0 \leq M \leq 1, \\
3/2 - 3M/4, & \text{if } 1 \leq M \leq 2, \\
0, & \text{if } M \geq 2.
\end{cases}
\]

On the other hand, the achievable rate \(\tilde{R}(M)\) for the coloring-based scheme proposed in [1] is given by

\[
\begin{cases} 
4 - 3M, & \text{if } 0 \leq M \leq 1, \\
2 - M, & \text{if } 1 \leq M \leq 2, \\
0, & \text{if } M \geq 2.
\end{cases}
\]

Figure 3 shows the comparison between the performance of the two schemes and demonstrates the improvement in rate using an index coding-based delivery scheme.

**Converse:** Recall that the server has \(N = 4\) files \(\{F_1, F_2, F_3, F_4\}\). Any uncoded placement policy divides each file \(F_i\) into \(2^L = 16\) disjoint parts (subfiles), denoted by \(\{F_{i,W} : W \in P(\{1, 2, 3, 4\})\}\), where \(F_{i,W}\) denotes the part of
requests an analysis of the achievable rate, we generate an instance of a virtual user corresponding to each subfile \( F \). Hence for which lists the combinations of non-zero for only some of the possible pairs \( W \) users in a virtual user representing (and requesting) some Subfile \( d \) have it in its cache. As before, the side information at the by a real user in the caching system, which does not already 

\[
\text{TABLE 1}
\begin{array}{|c|c|}
\hline
i & j \\
\hline 0 & 1,2,3,4 \\
1 & 0,1,2,3,4 \\
2 & 0,1,2,3,4 \\
3 & 0,1,3,4 \\
4 & 0,1 \\
\hline
\end{array}
\]

PAIRS \((i,j)\) FOR WHICH \(x_{i,j} = 0\)

file \( F_i \) which is available (via the caches) exclusively to the users in \( W \), and \( P(S) \) denotes the power set of \( S \).

Let \( x_{i,j} \) denote the total size of the file parts (in units) which are each stored on \( j \) caches and are available to \( i \) users. Hence, \( \sum_{i=0}^{4} \sum_{j=0}^{4} x_{i,j} = 4 \) (total size of all files). \( x_{2,1} + x_{3,2} + x_{4,2} + x_{4,3} + x_{4,4} = 4 \).

\[
\begin{align*}
x_{0,0} + x_{2,1} + x_{3,2} + x_{4,1} + x_{4,3} + x_{4,4} & = 4, \\
x_{2,1} + 2x_{3,2} + 2x_{4,3} + 3x_{4,4} & \leq 4M.
\end{align*}
\]

Our setup with \( K = 4 \) and \( L = 2 \) implies that \( x_{i,j} \) can be non-zero for only some of the possible pairs \((i,j)\). Table 1 lists the combinations of \( i \) and \( j \) which are not possible, and hence for which \( x_{i,j} = 0 \).

| \( d \) | \( F_{d_{01},W_1} \) for all valid \( d \) for valid subsets \( W_1 \subseteq \{ 1 : 4 \} \) \|
|---|---|
| \( d_{02} \) | \( F_{d_{02},W_2} \) for valid subsets \( W_2 \subseteq \{ 1 : 4 \} \setminus \{ u_1, u_2 \} \) \|
| \( d_{03} \) | \( F_{d_{03},W_3} \) for valid subsets \( W_3 \subseteq \{ 1 : 4 \} \setminus \{ u_1, u_2, u_3 \} \) \|
| \( d_{04} \) | \( F_{d_{04},W_4} \) for valid subsets \( W_4 \subseteq \{ 1 : 4 \} \setminus \{ u_1, u_2, u_3, u_4 \} \) \|

For example, when \( d = (1, 3, 4, 2) \) and \( u = (2, 3, 4, 1) \), the selected nodes include

\[
\begin{align*}
d_{u_1} = d_2 = 3 : & F_{3,W_1} \text{ for valid subsets } W_1 \subseteq \{ 3, 4, 1 \}, \\
d_{u_2} = d_3 = 4 : & F_{4,W_2} \text{ for valid subsets } W_2 \subseteq \{ 4, 1 \}, \\
d_{u_3} = d_4 = 2 : & F_{2,W_3} \text{ for valid subsets } W_3 \subseteq \{ 1 \}, \\
d_{u_4} = d_1 = 1 : & F_{1,W_4} \text{ for valid subsets } W_4 \subseteq \phi.
\end{align*}
\]

This collection of nodes is depicted in Figure 4. As the figure illustrates, the corresponding subset \( J \) of nodes \( \{ F_{3,\phi}, F_{3,\{3,4\}}, F_{3,\{4,1\}}, F_{3,\{3,4,1\}}, F_{4,\phi}, F_{4,\{4,1\}}, F_{2,\phi}, F_{1,\phi} \} \) does not induce a cycle in the side information graph. Then from Lemma 1, we have

\[
R^* (M) \geq |F_{3,\phi}| + |F_{3,\{3,4\}}| + |F_{3,\{4,1\}}| + |F_{3,\{3,4,1\}}| + |F_{4,\phi}| + |F_{4,\{4,1\}}| + |F_{2,\phi}| + |F_{1,\phi}|.
\]

In general, we can find a similar inequality as above for each combination of request profiles \( d \) with distinct demands amongst the users (\( 4! \) permutations) and rotations \( u \) of the users (\( 4! \) rotations). We then sum all the \( (4 \times 4!) \) inequalities to obtain

\[
4 \ast (4!) \ast R^* (M) \geq \sum \sum \sum \sum \sum |F_{d_{i,j},W_j}|
\]

where recall that \( x_{i,j} \) denotes the total size of the file parts (in units) which are each stored on \( j \) caches and are available to \( i \) users. Hence, \( R^* (M) \geq x_{0,0} + 3x_{2,1} + 6 \ast 3 \ast x_{3,2} + 16 \ast 3 \ast x_{3,2}. \)

If we substitute \( x_{0,0} \) and \( x_{2,1} \) from (8) and (9) in (10), we get

\[
R \geq 4 - \frac{13}{4} M + \frac{11}{16} x_{3,2} + \frac{5}{8} x_{4,2} + \frac{23}{16} x_{4,3} + \frac{36}{16} x_{4,4}
\]

\[
R \geq 4 - 13M/4 \text{ units}
\]

\footnote{Our problem setup doesn’t support some subsets. One example is \( (2, 4) \), because no cache is common to User 2 and User 4 and if it is stored in 2 caches then it will be available to at least 3 users.}
If we substitute \( x_{2,1} \) and \( x_{4,2} \) from (8) and (9) in (10), we get
\[
R \geq 3/2 - 3M/4 \text{ units} \quad (12)
\]
As illustrated in Figure 3, these lower bounds match exactly with the achievable rate of our proposed scheme for this setup. Thus, for the \((N = 4, K = 4, L = 2)\)-CCDN setup, we have characterized the exact rate-memory trade-off under the restriction of uncoded placement.

V. GENERAL POLICY AND MORE RESULTS

In this section, we present a generalization of the placement and index coding-based delivery scheme described in the previous section to the general \((N, K, L)\)-CCDN setup. Let \( t = KM/N \), and \( i = \min\{L, K/2\} \). Let \( \mathcal{S} \) be the union of subsets \( s \) of \( \{1, 2, ..., K\} \) which satisfy (i) \( |s| = t \), and (ii) if \( t > 1 \), every two elements \((j, l)\) of \( s \) satisfy \( |j - l| \geq i \) and \( |K - |j - l|| \geq i \).

**Placement policy:** Let \( X = |\mathcal{S}| \) and divide each file into \( X \) parts, with at least one subfile corresponding to each subset \( s \) in \( \mathcal{S} \). Store the subfile corresponding to set \( s \) in all the caches whose index belongs to \( s \).

**Delivery policy:** After users have revealed their requests, form an instance of the index coding problem with the file parts which are unavailable locally, the server transmits messages according to the solution of the index coding problem.

The above policy creates overlaps in the cache contents to aid coded-multicasting opportunities, while ensuring that there are no redundant copies. For example, a user should not have two copies of the same subfile amongst its accessible caches. It can be shown that the above general policy has a lower server transmission rate than the coloring-based policy proposed in [1]. Furthermore, the scheme is exactly optimal under the restriction of uncoded placement for the following scenarios:

1) \((N, K, L)\)-CCDN for any \( K \leq 4 \).
2) \((N, K = 6, L = 3)\)-CCDN,
3) \((N, K, L = K - 1)\)-CCDN.

For each of these cases, the lower bounds follow a similar recipe to the one illustrated for the \((N = 4, K = 4, L = 2)\)-CCDN in Section IV. Due to space constraints, we skip the proofs here and provide the details in [10].

While our ultimate goal is to characterize the exact rate-memory trade-off for the general \((N, K, L)\)-CCDN problem, there are several challenges involved. The general placement and delivery policy described above does provide an achievable rate-memory curve. However, proving a tight lower bound is more challenging for the general setup. For example, the achievable rate-memory trade-off of our policy for the \((N = 6, K = 6, L = 2)\)-CCDN is given by
\[
R^* (M) \leq R(M) = \begin{cases} 
6 - 25M/6, & \text{if } 0 \leq M \leq 1, \\
29/9 - 25M/8, & \text{if } 1 \leq M \leq 2, \\
4/3 - 4M/9, & \text{if } 2 \leq M \leq 3, \\
0 & \text{if } M \geq 3.
\end{cases}
\]

Unfortunately, for this case we do not yet have matching lower bounds. A critical step in proving the index coding-based lower bound (illustrated in Section IV) was to generate several inequalities by considering different request patterns \( d \) and different user orders \( u \), and then combining them together. For the \( L = 1 \) case, [2], [3] considered all possible permutations \( u \) to get a tight lower bound. However, for the multi-cache-access case with \( L > 1 \), all user permutations are not equivalent since some of the cache-access subsets are not feasible.

For example, recall the lower bound argument for the \((N = 4, K = 4, L = 2)\)-CCDN discussed in Section IV. For any request profile \( d \), if we use the user permutation \( u = (2, 4, 3, 1) \), we will get the inequality \( R^* (M) \geq |F_{d_2,|d|} + |F_{d_3,|d|} + |F_{d_4,|d|} + |F_{d_3,|d|} + |F_{d_4,|d|} | \), which contains one fewer term \((|F_{d_5,|d|})\) than the inequality we get for the user permutation \( u = (2, 3, 4, 1) \), given by \( R^* (M) \geq |F_{d_2,|d|} + |F_{d_3,|d|} + |F_{d_4,|d|} + |F_{d_3,|d|} + |F_{d_4,|d|} | \). For the special cases mentioned above where we get matching lower bounds with \( L > 1 \) (including the \((N = 4, K = 4, L = 2)\)-CCDN case), rotations (instead of permutations) for the user order \( u \) are all equivalent and suffice to get a tight lower bound. Unfortunately, they do not suffice in the general case and hence, we need more sophisticated analysis to devise tight lower bounds for the general \((N, K, L)\)-CCDN setup. This is part of our future work.

REFERENCES

[1] J. Hachem, N. Karamchandani, and S. N. Diggavi, “Coded caching for multi-level popularity and access,” IEEE Transactions on Information Theory, vol. 63, no. 5, pp. 3108–3141, 2017.
[2] K. Wan, D. Tuninetti, and P. Piantanida, “On the optimality of uncoded cache placement,” in Information Theory Workshop (ITW), 2016 IEEE. IEEE, 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 Transactions on Information Theory, 2017.
[4] Cisco Whitepaper: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/np-ngn-ip-next-generation-network/white_paper_c11-481360.html.
[5] M. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” IEEE Transactions on Information Theory, vol. 60, no. 5, pp. 2856–2867, 2014.
[6] K. Shamimugam, N. Golrezaei, A. Dimakis, A. Molisch, and G. Caire, “Femtocaching: Wireless content delivery through distributed caching helpers,” IEEE Transactions on Information Theory, vol. 59, no. 12, pp. 8402–8413, 2013.
[7] M. A. Maddah-Ali and U. Niesen, “Decentralized coded caching attains order-optimal memory-rate tradeoff,” Networking, IEEE/ACM Transactions on, vol. 23, no. 4, pp. 1029–1040, Aug 2015.
[8] Z. Bar-Yossef, Y. Birn, T. Jayram, and T. Kol, “Index coding with side information,” IEEE Transactions on Information Theory, vol. 57, no. 3, pp. 1479–1494, 2011.
[9] F. Arbabjolfaei, B. Bandemer, Y.-H. Kim, E. Şas¸lo˘glo, and L. Wang, “On the capacity region for index coding,” in Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on. IEEE, 2013, pp. 962–966.
[10] “On the exact rate-memory trade-off for multi-access coded caching with uncoded placement,” 2018, https://www.dropbox.com/s/mrnic0bjbgs5pv2/multiacc.pdf?dl=0.