The Interplay between Index Coding, Caching, and Beamforming for Fog Radio Access Networks

Salwa Mostafa*, Chi Wan Sung*, Terence H. Chan†, and Guangping Xu‡

*Department of Electrical Engineering, City University of Hong Kong, Hong Kong
†Institute for Telecommunication Research, University of South Australia, Adelaide, Australia
‡School of Computer and Communication Engineering, Tianjin University of Technology, Tianjin, China
smostafa3-c@my.cityu.edu.hk, albert.sung@cityu.edu.hk, Terence.Chan@unisa.edu.au, xugp@email.tjut.edu.cn

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9322474

***Abstract*—In fog radio access networks, the limited capacity of the fronthaul link is the bottleneck, which renders a high quality of service for video streaming difficult. To circumvent the problem, popular files can be cached in fog access points during off-peak hours. This work points out that beamforming in the access network can be exploited to reduce fronthaul traffic load by a joint design of cache placement scheme at the fog access points and index-coded transmission scheme over the fronthaul. Simulation results show that a percentage reduction of fronthaul traffic by more than 30% can be achieved.**

***Index Terms*—cooperative caching, index coding, beamforming, fronthaul traffic.**

I. Introduction

The fog radio access network (F-RAN) is a promising network architecture for 5G and beyond. In F-RANs, the fog access points (F-APs) are connected to the cloud server via fronthaul link. Due to the dense deployment of the F-APs and advanced physical layer techniques, the capacity of the access network has been significantly enlarged, which shifts the bottleneck of the network to the fronthaul link. To alleviate the burden over the fronthaul, popular contents are cached over the F-APs during off-peak time. How the contents are cached, however, plays an important role in minimizing the fronthaul traffic load.

In general, a file to be cached can either be coded or uncoded. In uncoded caching, the file is split into subfiles and stored in the F-APs directly. In coded caching, a file is coded before cached. For example, the use of MDS codes is considered in [1]. The optimal way of using coded caching is studied in [2], [3]. The system model considered, however, does not fully capture the increasingly complex structures of serving a user from multiple F-APs. For such a network scenario, different coding schemes are considered in [4]. In this paper, we extend the work by considering possible coded transmissions over the fronthaul and cooperative beamforming in the access network.

Beamforming is a well known physical-layer technique to increase cell coverage and boast the signal-to-noise ratio, and can be applied in cache enabled networks. For example, a joint design of cache placement and beamformer design is investigated in [5]–[7], and a joint F-AP clustering and beamformer design is studied in [7], [8]. Those works, however, focus only on the beamformer design. To the best of our knowledge, joint design of the whole system, including caching in the F-APs and delivery over both the fronthaul and access networks, has not been considered.

In this paper, we incorporate distributed beamforming in cache enabled networks in a unified framework. The interplay between index coding for fronthaul transmissions, caching at the F-APs, and beamforming for last-hop radio access is investigated, with the objective of minimizing the fronthaul traffic load. To exploit the beamforming of F-APs to users with index coding, we investigate the optimal transmissions for different caching schemes. For uncoded and repetition caching, the problem can be optimally solved in polynomial time. For MDS-coded caching, a heuristic algorithm is designed. Their performance in fully connected networks is mathematically analyzed, and in partially connected networks is evaluated by simulations. We show that in most network scenarios, MDS-coded caching outperforms the others.

The rest of this paper is outlined as follows. We present the system model in Section II and the caching schemes in Section III. We consider the joint design of caching and index coding for fully and partially connected networks in Sections IV and V, respectively. Simulation results are presented in Section VI and a conclusion is given in Section VII.

II. System Model

Consider a F-RAN, which consists of a cloud server, M cache-enabled F-APs, and N users. Denote the index sets of the F-APs and of the users by $\mathcal{M} \triangleq \{1, 2, \ldots, M\}$ and $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$, respectively. The cloud server has a library $\mathcal{W} = \{W^{(1)}, W^{(2)}, \ldots, W^{(F)}\}$ of F popular files, each of which has a size of B bits. Each user requests a file from $\mathcal{W}$ with a probability according to a file popularity distribution, $p_1, p_2, \ldots, p_F$, where $\sum_{f=1}^{F} p_f = 1$. The cloud server is connected to the F-APs via a wireless *broadcast* fronthaul. Typically, this link is bandwidth limited, and we assume that it supports noise-free transmissions of constant bit rate. To avoid overwhelming this link, a cache space of C bits is allocated to each F-AP, where $C < FB$.

This work was supported in part by a grant from City University of Hong Kong under Project 7005225, and National Natural Science Foundation of China (61971309) and Natural Science Foundation of Tianjin of China (17JCYBJC15600). (*Corresponding author: Chi Wan Sung.)*

F-AP m is connected to user n via a time-invariant Gaussian channel with amplitude gain $h_{nm} \in \mathbb{C}$ for $n \in \mathcal{N}$ and $m \in \mathcal{M}$. This amplitude gain is normalized such that the noise power at each receiver is equal to 1. The channel is assumed power limited, and each F-AP is subject to a peak power constraint of P. Given a fixed modulation and coding scheme, a target signal-to-noise ratio (SNR), γ, has to be met. If $|h_{nm}|^2 \geq \frac{\gamma}{P}$, the link is said to be *strong* and information can be successfully delivered. If $\frac{\gamma}{4P} \leq |h_{nm}|^2 < \frac{\gamma}{P}$, the link is said to be *weak*. Suppose user n requests a file which is cached on two F-APs (say m and m') but their links are weak. The two F-APs can use beamforming to transmit identical bits to user n, since the two transmit signals can be phase aligned and the received SNR becomes $(|h_{nm}|^2 + |h_{nm'}|^2 + 2|h_{nm}||h_{nm'}|)P \geq \gamma$. A user is said to be strongly associated with an F-AP if the corresponding link is strong, and weakly associated if the link is weak. A network is said to be *fully connected* if each user is associated, either weakly or strongly, to all F-APs. Otherwise, it is *partially connected*. The association matrix between the users and the F-APs is represented by an $N \times M$ ternary matrix $\boldsymbol{A}$, where a_{nm} equals 0, 1, or 2 if the link between user n and F-AP m is missing, weak, or strong, respectively.

III. Cache Placement and File Delivery

We consider three caching schemes. Under each scheme, a file, $W^{(f)}$, to be cached, is partitioned into $k \leq M$ subfiles, $W_1^{(f)}, W_2^{(f)}, \dots, W_k^{(f)}$, of equal size.

- *Uncoded Caching* ($k = M$): Each F-AP m stores the subfile $W_m^{(f)}$, for $m \in \mathcal{M}$. All the subfiles are also stored in the cloud.
- *Repetition Caching* ($k = \frac{M}{2}$): Each F-AP m stores the subfile $W_{(m \bmod k)+1}^{(f)}$, for $m \in \mathcal{M}$. We assume M is an even number throughout this paper. All the subfiles are also stored in the cloud.
- *MDS-Coded Caching* ($k \leq M$): The k subfiles are encoded using an $(M+k, k)$ MDS code to obtain $M+k$ coded packets. The first M of them are for caching, each placed in one F-AP. They are also stored in the cloud. In addition, the remaining k of them, denoted by $\mathcal{Z}$, are not stored in the F-APs but stored only in the cloud.

For each scheme, the information cached in an F-AP for a file is called a *packet*.(For uncoded and repetition caching, a packet stored in an F-AP refers to a subfile.) The packet sizes for uncoded caching, repetition caching, and MDS-coded caching are B/M, $2B/M$, and B/k bits, respectively. In general, the cache space may not be large enough to cache all files in the F-APs. In that case, files are cached according to *Most Popular First* (MPF), i.e., one after another in descending order of their popularity. (If there are two or more files of the same popularity, the tie is broken arbitrarily.) If the remaining space is not enough to cache a whole packet, part of a packet is cached, which completely fills up the available cache space.

If a user cannot obtain enough information from his associated F-APs, the cloud server has to broadcast a certain number of fixed-length packets, possibly coded, through the fronthaul link. After receiving a packet, an F-AP can either directly forward the packet, or compute over the received packet and its cached packets and send the results to one or more of its associated users. Afterwards, the F-AP discards the received packet before processing the next packet from the fronthaul link. This feature is reminiscent of *instant decodability* in the literature of index coding. In this paper, we consider intra-file index coding. Users requesting the same file are grouped and served together. We consider the design of index coding schemes for fronthaul transmissions. Packets stored in the cloud are transmitted over the fronthaul either *without coding* or *with bitwise XOR between two packets*. Our objective is to minimize the expected number of transmitted bits, $E[\Lambda]$, across the fronthaul under random realization of network connectivity and user requests.

IV. Fully Connected Networks

In this section, we consider the fronthaul traffic load minimization problem over a fully connected network. Since different files are treated independently, it suffices to describe the transmissions for one single file, $W^{(f)}$. For simplicity, we denote the set of users requested it by $\mathcal{N}$, where $|\mathcal{N}| = N$.

A. Optimal File Delivery

For repetition caching, each subfile of $W^{(f)}$ is stored twice. Therefore, all users can obtain the cached subfiles via a strong link or beamforming, thus incurring no fronthaul traffic.

For uncoded caching, each user needs the subfiles of $W^{(f)}$ cached on all F-APs. If an F-AP connects to all users via strong links, its cached subfile can be obtained by all users. Otherwise, its subfile needs to be sent over the fronthaul. Denote the set of those F-APs by $\mathcal{M}' \triangleq \{m \in \mathcal{M} \mid a_{nm} = 1 \text{ for some } n \in \mathcal{N}\}$. Since the network is fully connected, for any distinct $i, j \in \mathcal{M}'$, if $W_i^{(f)} \oplus W_j^{(f)}$ is transmitted, all users can obtain both $W_i^{(f)}$ and $W_j^{(f)}$ via two packet transmissions either over one strong link or beamforming on two weak links. Therefore, those packets can be paired up arbitrarily to form XOR packets. If the number of those packets is odd, the unpaired one is sent uncoded. This is clearly the best we can do. Hence, the minimum number of packets need to be sent to deliver $W^{(f)}$ to all users is $\lceil |\mathcal{M}'|/2 \rceil$.

For MDS-coded caching, each user requires k unique coded packets to reconstruct $W^{(f)}$. Let $\boldsymbol{s} \triangleq (s_1, s_2, \dots, s_N)$, where user n has s_n strong links. The number of remaining coded packet required for each user is $\boldsymbol{r} = \max(\boldsymbol{k} - \boldsymbol{s}, \boldsymbol{0})$, where $\boldsymbol{k}$ is an N-vector with each element equal to k, $\boldsymbol{0}$ is the zero vector, and the maximum is taken in a component-wise manner. To determine which packets to deliver, *binary* linear programming (LP) can be used:

$$\begin{aligned} \min \quad & \sum_{m=1}^{M} x_m \\ \text{subject to} \quad & \boldsymbol{P}\boldsymbol{x} \geq \boldsymbol{r}, \end{aligned} \tag{1}$$

where $\boldsymbol{P} \triangleq [P_{nm}]$ is an $N \times M$ binary matrix. If user n misses the coded packet m, then P_{nm} equals 1; otherwise it equals 0. The decision variable x_m is either zero or one, which indicates whether the coded packet cached in F-AP m should be sent, with or without index coding. The objective is to minimize the number of those packets. After an optimal vector $\boldsymbol{x}$ is obtained, the corresponding packets are paired up for XOR transmissions. If the weight of $\boldsymbol{x}$ is odd, the last packet is transmitted without index coding. This method is optimal if the extra parity packet in $\mathcal{Z}$ is not allowed to use.

B. Fronthaul Traffic Analysis

Now we analyze the expected fronthaul traffic for each caching scheme for the case where $F = 2$ and $MC = 2B$. (The same methodology can be applied for a large library of files.) The two files are requested by a user with probability p_1 and $p_2 \triangleq 1 - p_1$, where $p_1 \geq 0.5$. Each link is strong with probability q and weak with probability $1 - q$. For uncoded caching, both files can be cached. For repetition caching, only the first file is cached. For MDS-coded caching, let $k = \lceil Mq \rceil$ for both files. If $q \geq 0.5$, each of the M coded packets for the first file is entirely stored in each F-APs, while those for the second file is partially stored in each F-APs due to the limitation of cache space.

Theorem 1. *Consider a fully connected networks with $F = 2$ and $MC = 2B$.*

- *For repetition caching, $E[\Lambda]$ is given by*
$$(1 - p_1^N)B. \tag{2}$$
- *For uncoded caching, $E[\Lambda]$ is given by*
$$\sum_{n=0}^{N} b_{N,p_1}(n) \sum_{j=0}^{M} \left[b_{M,1-q^n}(j) + b_{M,1-q^{N-n}}(j)\right] \left\lceil \frac{j}{2} \right\rceil \frac{B}{M}, \tag{3}$$
where $b_{N,p}(i) \triangleq \binom{N}{i} p^i (1-p)^{N-i}$.
- *For MDS-coded caching with $k = \lceil Mq \rceil$, $E[\Lambda]$ is bounded below by*
$$(1 - p_1^N)\left(1 - \frac{\lceil Mq \rceil}{M}\right) 2B, \text{ for } q \geq 0.5. \tag{4}$$
Moreover, the lower bound is asymptotically tight when M goes to infinity.

Proof. For repetition caching, each user can obtain all the subfiles of the first file even if all links are weak. If the second file is requested by any user, then B bits needs to be transmitted over the fronthaul, which occurs with probability $1 - p_1^N$. Hence, $E[\Lambda] = (1 - p_1^N)B$.

For uncoded caching, let N_1 be the number of users requesting file 1. Since there is no interfile coding,
$$E[\Lambda] = \sum_{n=0}^{N} b_{N,p_1}(n) \left(E[\Lambda_1 | N_1 = n] + E[\Lambda_2 | N_1 = n]\right), \tag{5}$$
where Λ_1 and Λ_2 are the traffic loads over the fronthaul due to files 1 and 2, respectively. Consider the traffic load due to file 1. Denote the set of F-APs that have weak links by $\mathcal{M}'$. As discussed in the previous subsection, the number of packets required is $\lceil |\mathcal{M}'|/2 \rceil$. Hence,
$$E[\Lambda_1 | N_1 = n] = \sum_{j=0}^{M} b_{M,1-q^n}(j) \left\lceil \frac{j}{2} \right\rceil \frac{B}{M}. \tag{6}$$
Similarly, we can obtain the traffic load due to file 2. Hence, we obtain (3).

Last, consider MDS-coded caching. We make an optimistic assumption that each user has k or more strong links. Consider the case where $q \geq 0.5$. Due to the MDS property, all users can obtain file 1 and the portion of file 2 that is stored in the network. If one or more users requests file 2, which occurs with probability $1 - p_1^N$, then the remaining portion of file 2 needs to be transmitted over the fronthaul. It is straightforward to show that the amount is equal to $(1 - \frac{k}{M})2B$, which gives (4). When M goes to infinity, the probability that each user has exactly k strong links approaches 1, so the bound becomes tight. □

V. Partially Connected Networks

In this section, we consider the delivery of a single file over partially connected networks. Our aim is to find algorithms to minimize the fronthaul traffic for the three caching schemes. We first show that the problem with repetition caching can be reduced to that with uncoded caching.

For repetition caching, every pair of F-APs that cache the same subfile can be combined into one single F-AP, so the network can be transformed into one that has $M/2$ F-APs with a new $N \times M/2$ association matrix $\boldsymbol{A}'$, whose entries are defined by $a'_{n,m} = \min(a_{n,m} + a_{n,m+M/2}, 2)$, for all $n \in \mathcal{N}$ and $m \in \mathcal{M}$. It means that the link in the transformed network between user n and F-AP m is strong if either of the corresponding links in the original network is strong or both are weak. The link is weak if exactly one of the corresponding links in the original network is weak. It is missing if both of the corresponding links are missing. It is clear that the transformed network is equivalent to the original network in the sense that the same number of fronthaul traffic is required for successful file delivery. In other words, it suffices to design algorithms for uncoded caching and MDS-coded caching only.

A. Optimal Index Coding for Uncoded Caching

Assume that each user is connected with at least one strong link or at least two weak links, for otherwise it cannot receive anything from the network, and the problem is clearly infeasible. Let $A[i, j]$ be the submatrix of $\boldsymbol{A}$ obtained by preserving only columns i and j of $\boldsymbol{A}$. A pair of distinct subfiles, i and j, denoted by (i, j), is said to be a *potential coded group*, if the sum of each row of $A[i, j]$ is greater than or equal to two. It has the property that if $W_i \oplus W_j$ is transmitted over the fronthaul, all users must have both W_i and W_j.

First, we identify those F-APs which have strong links to all users. The subfiles cached in those F-APs can be delivered directly without incurring fronthaul traffic. Next, we focus on the remaining subfiles and search for all potential coded

Algorithm 1: Index Coding for Uncoded Caching in Partially Connected Networks

```
Input  : A set of F-APs $\mathcal{M}$, a set of users $\mathcal{N}$, an
         association matrix $\boldsymbol{A}$.
Output: A set of packets $\mathcal{I}$.
1: Let $\mathcal{V} := \mathcal{M} \setminus \{m \in \mathcal{M} \mid a_{nm} = 2 \ \forall n \in \mathcal{N}\}$;
2: Construct a graph $G(\mathcal{V}, \mathcal{E})$, where $(i, j) \in \mathcal{E}$ if
   $(i, j) \in \mathcal{V}^2$ is a potential coded group;
3: Find a maximum matching $\mathcal{I}$ for $G$;
4: Add all unmatched vertices in $\mathcal{V}$ to $\mathcal{I}$;
5: return $\mathcal{I}$;
```

groups. Construct a graph with vertices corresponding to the remaining subfiles and edges corresponding to the potential coded groups. Then, we find a maximum matching in the graph, which indicates the coded packets to be transmitted over the fronthaul. Subfiles corresponding to unmatched vertices are transmitted as uncoded packets.

The pseudo-code is stated in Algorithm 1. Each element in $\mathcal{I}$ represents an uncoded packet while each pair in $\mathcal{I}$ represents a coded packet by XOR between the pair. The number of packet transmissions is given by the cardinality of $\mathcal{I}$. Identifying the subfiles in Step 1 requires $O(NM)$. Finding all potential coded group in Step 2 requires $O(NM^2)$. The maximum matching problem in Step 3 can be solved in $O(M^{2.5})$ by Micali-Vazirani Algorithm [9]. The overall time complexity of Algorithm 1 is therefore $O(NM^{2.5})$.

Lemma 2. *There always exists an optimal solution which does not contain a cycle of three or more coded packets, i.e.,* $W_{k_1} \oplus W_{k_2}, W_{k_2} \oplus W_{k_3}, \ldots, W_{k_n} \oplus W_{k_1}$*, where* $k_1, k_2, \ldots, k_n$ *are distinct and* $n \geq 3$.

Proof. If there is such an optimal solution, the n coded packets can be replaced by the corresponding n uncoded subfiles, i.e, $W_{k_1}, W_{k_2}, \ldots, W_{k_n}$. The resultant solution is feasible and has exactly the same number of packets transmitted as the given optimal solution. □

Lemma 3. *Given that* $W_i \oplus W_j$ *has been transmitted over the fronthaul link, all users can obtain* W_i *if and only if they can obtain* W_j*, for any* $i \neq j$.

Proof. Assume that all users can obtain W_i after transmitting $W_i \oplus W_j$. For each user, there are only two possibilities. Each user either has a strong link with W_i or W_j (or both), or weak links with both W_i and W_j. In the former case, the user can obtain both subfiles via the strong link. In the latter case, the user can obtain both subfiles via beamforming using both links. In either case, the user can obtain W_j as well. The converse can be proved with the same argument. □

Theorem 4. *Algorithm 1 is optimal.*

Proof. In Step 2, only potential coded groups are considered. We need to show that there is always an optimal solution which uses only uncoded packets and potential coded groups.

Algorithm 2: Index Coding for MDS-Coded Caching in Partially Connected Networks

```
Input  : A set of F-APs $\mathcal{M}$, a set of users $\mathcal{N}$, an
         association matrix $\boldsymbol{A}$, a set of MDS coded
         packets $\mathcal{Z}$.
Output: A set of packets $\mathcal{I}$.
1: Let $r_n$ be the extra number of packets required by
   user $n$ for $n \in \mathcal{N}$;
2: Let $\mathcal{V} := \mathcal{M} \setminus \{m \in \mathcal{M} \mid a_{nm} = 2 \ \forall n \in \mathcal{N}\}$;
3: Construct a graph $G(\mathcal{V}, \mathcal{E})$, where $(i, j) \in \mathcal{E}$ if
   $(i, j) \in \mathcal{V}^2$ is a potential coded group;
4: Find a maximum matching $\mathcal{P}$ for $G$;
5: while $r_n > 0$ for some $n$ do
6:    if $\mathcal{P}$ is non-empty then
7:       Move an arbitrary element $p$ from $\mathcal{P}$ to $\mathcal{I}$;
8:       Update $r_n$ for all $n$, assuming $p$ is broadcast;
9:    else
10:      Move $\max_n r_n$ elements from $\mathcal{Z}$ to $\mathcal{I}$;
11:      Let $r_n := 0$ for all $n$;
12:   end if
13: end while
14: return $\mathcal{I}$;
```

By Lemma 2, there is an optimal solution that has no cycle of coded packets. Assume that this optimal solution contains a coded packet $W_i \oplus W_j$, where $i \neq j$ and (i, j) is not a potential coded group. By definition, not all users can obtain one of the subfiles, say W_i. By Lemma 3, not all users can obtain W_j as well. Therefore, both W_i and W_j need to be involved in other packets in the solution. We need to show that it is unnecessary to consider $W_i \oplus W_j$ in Step 2.

First, consider the case where the uncoded packet W_i is in the solution. Then $W_i \oplus W_j$ can be replaced by W_j, and we are done. Next, consider the case where the solution contains the following maximal chain of $m + n + 1$ coded packets: $W_{l_1} \oplus W_{l_2}, \ldots, W_{l_m} \oplus W_i, W_i \oplus W_j, W_j \oplus W_{k_1}, \ldots, W_{k_{n-1}} \oplus W_{k_n}$. Since there is no cycle, those subfile indices are all distinct. If all users can obtain W_{k_n}, then by Lemma 3, they can obtain $W_{k_{n-1}}$ as well. By replacing $W_{k_{n-2}} \oplus W_{k_{n-1}}$ by $W_{k_{n-2}}$, we can then shorten the chain. By repeating the argument, we can assume that not all users can obtain W_{l_1} or W_{k_n}. Since the chain is maximal, the solution must contain the uncoded packets W_{l_1} and W_{k_n}. Totally, there are $m + n + 3$ packets, which carry $m + n + 2$ subfiles. A better solution can be obtained by replacing the $m + n + 3$ packets by the $m + n + 2$ uncoded subfiles, contradicting the assumption that the given solution is optimal. Hence, there is no need to consider $W_i \oplus W_j$ in Step 2.

Note that the more disjoint potential coded groups are selected, the less uncoded subfiles need to be transmitted. Hence, Step 3 minimizes the number of packet transmissions. □

B. Heuristic Algorithm for MDS-Coded Caching

For MDS-coded caching, each user requires k unique coded

Table I
PARAMETERS FOR PARTIALLY CONNECTED NETWORKS

Parameters	Value
Cell radius (R)	500 m
Number of F-APs (M)	10 F-APs
Number of Users (N)	$5-55$ users
F-APs Peak Power (P)	2 W
Target SNR (γ)	$6-14$ dB
Path loss at distance d Km	$140.7+36.7\log_{10}d$, dB
Noise Power (σ^2) (10 MHz bandwidth)	-102 dBm
Number of Files (F)	10 files
Distribution Skewness (α)	1.5
File Size (B)	100 Mbits
Cache Size (C)	100 Mbits

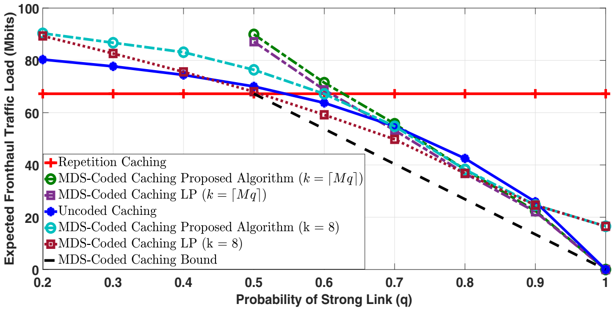

Figure 1: Expected fronthaul traffic load with beamforming for fully connected networks where $N=5$, $M=10$, $F=2$ and $p_1=0.8$.

packets to reconstruct his requested file. Let $\mathcal{N}$ contain the indices of the users who cannot reconstruct the requested file. First, the number of distinct coded packet required for each user $n\in\mathcal{N}$ is computed as follows: $r_n=\max(k-s_n,0)$, where s_n is the number of strong links associated with user n. Next, we identify the F-APs that have strong links to all users and exclude them from $\mathcal{M}$ to obtain $\mathcal{V}$. Afterwords, we construct a graph G with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$ defined by the potential coded groups in $\mathcal{V}$. Then, we find a maximum matching $\mathcal{P}$ in the graph, which corresponds to a set of XOR coded packets. To satisfy the users, we broadcast these XOR coded packets one by one. If all of them have been sent while some users are still not satisfied, we broadcast parity packets from the set $\mathcal{Z}$. Once all users obtain k unique coded packets, the algorithm returns the transmitted coded packets so far.

The pseudo-code is stated in Algorithm 2. Each of Steps 1 and 2 requires $O(NM)$. Finding all potential coded group in Step 3 requires $O(NM^2)$. The maximum matching problem in Step 4 can be solved in $O(M^{2.5})$. The while loop in Step 5 executes $O(M)$ times, and Step 8 requires $O(N)$. The overall time complexity of Algorithm 2 is therefore $O(NM^{2.5})$.

VI. SIMULATION MODEL AND RESULTS

First, we consider the fully connected network that we have investigated in Section IV. We consider the scenario where $M=10$, $N=5$ and $p_1=0.8$, and plot the expected fronthaul traffic of different caching schemes against the probability of strong link, q in Fig. 1. For MDS-coded caching, we consider two settings, namely, $k=\lceil Mq\rceil$ and $k=8$. For each setting, we consider two transmissions scheme, namely, the LP method in (1) and the heuristic algorithm designed for the partially connected network. Note that uncoded caching stores both files in the network. When $q=1$, all links are strong, and clearly uncoded caching performs the best, as no fronthaul traffic is needed. On the other hand, when q becomes smaller, it is outperformed by all four versions of MDS codes. When $q>0.8$, setting k larger than 8 gives better performance. Otherwise, when $q<0.8$, setting $k=\lceil Mq\rceil$ is too optimistic. Overall, setting $k=8$ performs well over a large range of q. Besides, in both settings, LP outperforms the heuristics, as expected, since LP provides optimal index coding solution. It still outperforms uncoded caching when q is further reduced down to 0.5. When $q<0.5$, repetition caching dominates, since it can make use of the weak links to perform beamforming. The advantage is more obvious when N is small and p_1 is large so that the chance that file 2 is requested is small.

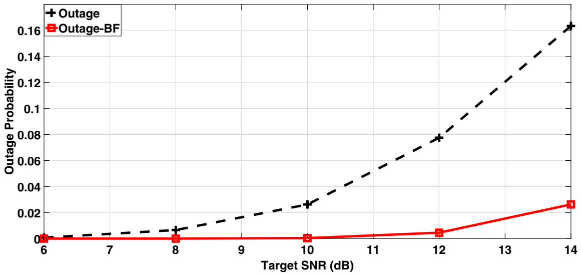

Figure 2: Outage probability for partially connected networks.

Next, for the partially connected network, we consider a single cell of radius R with a cloud server located at its center. The F-APs and the users are randomly distributed according to a homogeneous Poisson point process. The F-APs are restricted to an inner concentric circle with radius $R/2$ while the users are distributed over the whole cell. The signal attenuation from an F-AP to a user follows the path-loss model in the 3GPP standard [10]. Each user requests a file from the library according to the Zipf distribution $p_f=f^{-\alpha}/\sum_{i=1}^{F}i^{-\alpha}$ where $\alpha\geq 0$ is the distribution skewness. Large value of α means more requests are concentrated on fewer popular files. The simulation parameters are listed in Table I, and results are averaged over 1,000 random realizations.

To investigate the effect of the link condition between the F-APs and the users, we vary the target SNR, γ. If γ is larger, there are more weak and missing links. A user is said to be in outage if he is unable to obtain his requested file. Fig. 2 shows that beamforming reduces outage probability significantly for high target SNR.

For MDS-coded caching, it is unclear how the parameter k should be determined. Figs. 3 and 4 show the optimal choice of k for different γ and N, respectively. If γ is large, a small value of k should be chosen to tolerate more missing links. If N is large, most files in the library are requested by some users, so it would be better to choose a large value of k, which reduces the redundancy due to coding.

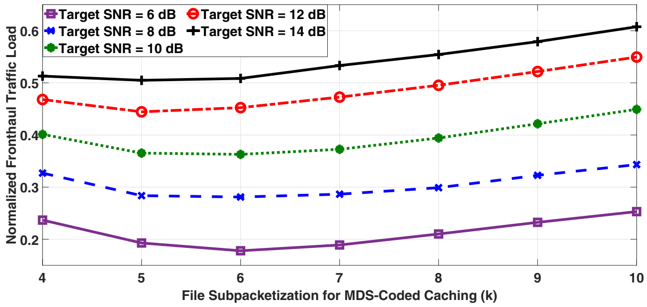

Figure 3: Normalized fronthaul traffic load with beamforming for partially connected network where $N = 15$ and $\alpha = 1.5$.

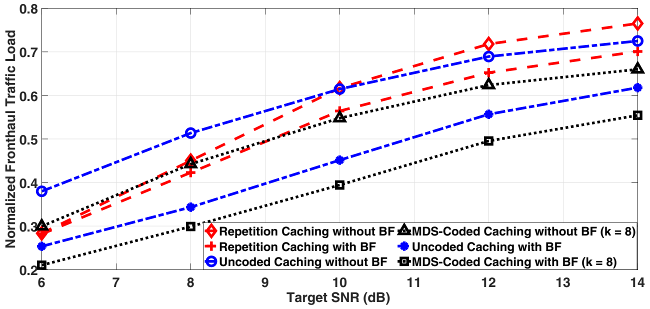

Figure 5: Normalized fronthaul traffic load for partially connected network where $N = 15$ and $\alpha = 1.5$.

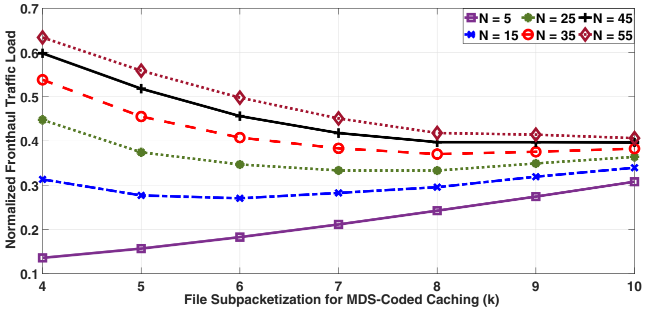

Figure 4: Normalized fronthaul traffic load with beamforming for partially connected network where $\gamma = 8$ dB and $\alpha = 1.5$.

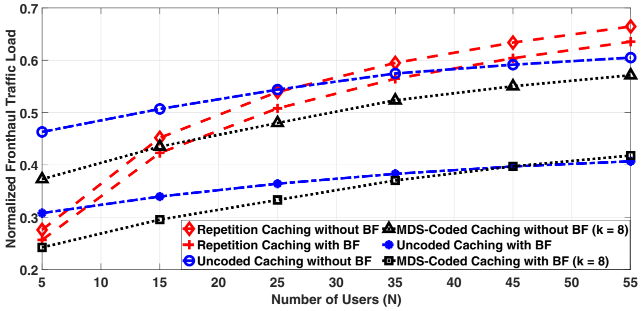

Figure 6: Normalized fronthaul traffic load for partially connected network where $\gamma = 8$ dB and $\alpha = 1.5$.

Figs. 5 and 6 show the fronthaul traffic load in the partially connected network with various values of γ and N, respectively. For MDS-coded caching, k is chosen to be 8, which gives reasonably good performance over a wide range of parameters. It can be seen that beamforming (labeled as "BF") can reduce the fronthaul traffic load for all schemes, but the effect is more significant for uncoded caching and MDS-coded caching. For example, when $\gamma = 8$ and $N = 15$, the reduction in fronthaul traffic is 6.3%, 32.8%, and 32.4% for repetition caching, uncoded caching, and MDS-coded caching, respectively. In general, MDS-coded caching outperforms the two other schemes, except only in more extreme cases. For example, when $N = 5$, only a few popular files are requested, so the disadvantage of low storage efficiency of repetition caching diminishes. When $N = 55$, almost all files in the library are requested by some users, so uncoded caching has an advantage due to its high storage efficiency.

VII. Conclusion

Distributed beamforming is a promising physical-layer technique to increase cell coverage and boast received SNR. In this work, we show that not only can it lower the outage probability of a F-RAN with cache-enabled F-APs, but also reduce fronthaul traffic load. Three caching schemes, namely, uncoded caching, repetition caching, and MDS-coded caching are investigated. For these schemes, algorithms are designed to facilitate index coding in the fronthaul transmissions, which reap the potential gain offered by beamforming and yield a significant reduction in fronthaul traffic.

References

[1] J. Liao, K.-K. Wong, Y. Zhang, Z. Zheng, and K. Yang, "Coding, multicast, and cooperation for cache-enabled heterogeneous small cell networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6838–6853, 2017.

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

[3] K. Zhang and C. Tian, "Fundamental limits of coded caching: From uncoded prefetching to coded prefetching," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1153–1164, 2018.

[4] S. Mostafa, C. W. Sung, and G. Xu, "Code rate maximization of cooperative caching in ultra-dense networks," in *IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6, 2019.

[5] X. Wu, Q. Li, V. C. Leung, and P. Ching, "Joint fronthaul multicast and cooperative beamforming for cache-enabled cloud-based small cell networks: An MDS codes-aided approach," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4970–4982, 2019.

[6] R. Sun, Y. Wang, N. Cheng, L. Lyu, S. Zhang, H. Zhou, and X. Shen, "QoE-driven transmission-aware cache placement and cooperative beamforming design in cloud-RANs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 636–650, 2019.

[7] M.-M. Zhao, Y. Cai, M.-J. Zhao, B. Champagne, and T. A. Tsiftsis, "Improving caching efficiency in content-aware C-RAN-based cooperative beamforming: A joint design approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4125–4140, 2020.

[8] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6118–6131, 2016.

[9] S. Micali and V. V. Vazirani, "An $o(\sqrt{|V|}|e|)$ algoithm for finding maximum matching in general graphs," in *21st Annual Symposium on Foundations of Computer Science (SFCS)*, pp. 17–27, 1980.

[10] "Further advancements for E-UTRA physical layer aspects (Release 9), 3GPP standard TS 36.814," Mar. 2010.