Adaptive Radio Unit Selection and Load Balancing in the Downlink of Fog Radio Access Network

Di Chen, Volker Kuehn
Institute of Communications Engineering
University of Rostock
Email: {di.chen, volker.kuehn}@uni-rostock.de

Abstract—The Radio Access Network based on Fog computing (F-RAN) has been recently proposed to alleviate the heavy burden on backhauls of Cloud Radio Access Network (C-RAN). Equipping the edge device of C-RAN with local storage module (cache) is a specific realization of F-RAN. Some frequently requested files are cached at the Radio Units (RUs), both the traffic load and the overall delay can be reduced. In this paper, we consider the downlink of the Cache-Enabled F-RAN, where minimization of power consumption is investigated for the Green Communication. We focus on the design of green network at the system level, such that not only the transmission power, but also all additional operational power of active RUs (incl. power used for circuits, associated backhaul, etc.) is considered. Based on the channel states, requested and cached files, an efficient algorithm for the joint optimization of RU selection, clustering formulation and beamforming is proposed. Meanwhile, the desired Quality of Service (QoS) is guaranteed and the load on each active backhaul is balanced according to its capacity. Numerical results show that with the proposed algorithm, more RUs can be switched off to achieve greener network, as the cache memory is increased.

I. INTRODUCTION

Cloud Radio Access Network (C-RAN) [1] has attracted an explosive enthusiasm among researchers worldwide in recent years. Due to the centralized baseband processing in the Cloud, much more efficient interference management and traffic handling can be achieved, leading to much higher Spectral Efficiency (SE) and Energy Efficiency (EE) of the network. Hence, C-RAN has shown to be a promising architecture for the Fifth-Generation (5G) wireless system. However, one limitation of practical implementation of C-RAN is its prohibitively high demand on the capacity of backhaul, which connects the Radio Unit (RU) on the edge and the Central Processor (CP) in the Cloud. It becomes the main bottleneck of the performance of C-RAN. Several approaches have been considered in order to alleviate the heavy burden on backhauls. Flexible Assignment of RAN Functionality is a possible solution [2], which pushes a part of baseband processing functionalities from the Cloud back again to edge devices (e.g. RUs). Based on such ideas, Fog Radio Access Network (F-RAN), an evolution of C-RAN, has been recently proposed and studied in [3] and [4]. In F-RAN, edge devices are assumed to perform some baseband processing functions locally in a distributed manner, and have distributed storing capability (e.g. cache). Fog computing can greatly reduce the burden on backhaul as well as the delay, and keeps some benefits of centralized Cloud computing [3].

Introducing cache modules on edge devices but retaining all baseband processing functionalities still at the CP is a cheap and easy realization of F-RAN. Moreover, recent studies show that some popular multimedia streams, e.g., newly released movies, generate a significant portion of the traffic. The same content may be requested by many users simultaneously. This unequal popularity and multicast nature make caching popular contents more sensible [5]. In [6], information theoretical fundamental limits of caching are investigated and two caching schemes are proposed: uncoded caching and coded caching. Based on the theoretical results, design of Cache-Enabled F-RAN is addressed in many recent works [3], [4], [7]–[9]. In [7] and [8], F-RAN with uncoded caching is considered, where same files are cached at all RUs. Users requesting cached files, can be cooperatively served by RUs without consuming backhaul resources. Uncached files have to be fetched remotely from the CP via backhauls. If more RUs participate in a cluster of serving an uncached file (larger cluster), more backhaul resources are needed while the total transmission power can be decreased due to larger aggregated array gain. If less RUs are involved (smaller cluster), less backhaul resource cost leads to smaller aggregated array gain, thus more transmission power is to be consumed. The trade-off between total backhaul cost and total transmission power consumption is discussed in [7] and [8]. A similar trade-off for coded caching is studied in [9].

However, only transmission power is considered in all these works. In [10], it has shown that for a typical Micro Base Station (BS) of LTE system, the average transmission power is usually only 6.3 Watt, while all additional operational power (incl. power consumed by circuits, cooling system, etc.) can be as high as 56 Watt! Moreover, F-RAN is featured by its relatively large backhaul capacity with considerable power consumption. Hence, the scheme used in [3], [4], [7]–[9], such that activating all RUs for more potential array gain, in order to decrease the transmission power, might not necessarily be paid off finally, when such additional power is also considered, since deactivating some RUs might save more power. Thus, for the design of green networks, it is wiser to consider the power consumption at the system level. The RU selection problem is studied for C-RAN in [11], [12]. Seen from this perspective, for Cache-Enabled F-RAN, there are generally two trade-offs:

1. Trade-off between the total transmission power and total operational power: When more RUs are selected to be active, operational power is increased while transmission power can be reduced due to the increased array gain, and vice versa.

2. Trade-off between the transmission power of active RUs and the backhaul capacity cost: When more active RUs are involved in the cluster serving an uncached file, more backhaul capacity is needed, while less transmission power is consumed due to more cooperation, and vice versa.
Moreover, in works mentioned above, only the trade-off between the transmission power and total backhaul capacity cost is studied, while such optimal trade-off may result in severe imbalanced traffic load. We have preliminarily discussed the issue of load balancing in [13], however, RU selection is also not addressed there. Due to the interplay between RU selection and load balancing, in this paper, instead of such trade-offs, we investigate the practical design of green network, by taking the operational power, as well as individual backhaul capacities into account. In order to minimize the total power consumption of the downlink, based on the channel, requested files, cached files and individual backhaul capacities, we propose a joint optimization algorithm of RU selection and load-balancing oriented clustering among active RUs, which

1. adaptively deactivates RUs with comparatively less contributions, for the reduction of operational power;
2. selects the RUs with comparatively more contributions to be active, so as to decrease the transmission power;
3. dynamically excludes an active RU from a cluster, if it contributes comparatively less to the corresponding multicast group, so as to decrease the backhaul cost and satisfy individual backhaul capacity constraint;
4. involves an active RU in a cluster, when it contributes comparatively more to the corresponding multicast group, for the guarantee of QoS (regarded as the received SINR here);
5. optimizes beamforming vectors to minimize the total power consumption for higher EE.

Paper structure: In Sec. II we introduce the channel model and state the problem mathematically. The optimization algorithm is proposed and explained in Sec. III. Simulation results and conclusions are in Sec. IV and Sec. V, respectively.

II. SYSTEM MODEL AND PROBLEM STATEMENT

A. Channel and Power Model

We consider the downlink of a hexagonal multi-cell Cache-Enabled F-RAN, as illustrated in Fig. 1. Totally \( N \) Radio Units (RUs) are located in the network, each lies at the center of its cell and the distance between adjacent RUs is denoted by \( d_{RU} \). Each RU is equipped with \( L \) antennas and a cache. Let \( \mathcal{N} = \{1, 2, ..., N\} \) denote the set of RUs, the \( n \)-th RU connects to the CP via a backhaul of capacity \( C_{BH,n} \). The RU is assumed to be either in active or sleep mode, all active RUs cooperatively serve all users in all cells of the network. Similar to [10], [11], [14], the total power consumption \( P_n \) of RU \( n \) is modeled as

\[
P_n = \begin{cases} 
P_{active,n} = \frac{1}{\epsilon} P_{TX,n} + P_{RU} + P_{BH}, & P_{TX,n} > 0 \\
P_{sleep}, & P_{TX,n} = 0 \end{cases}
\]

where \( P_{TX,n} \) denotes its transmission power, \( \epsilon \in [0, 1] \) is the power amplifier efficiency. If RU \( n \) is selected to serve users, it is in active mode and \( P_{TX,n} > 0 \) holds. \( P_{RU} \) is the sum of all additional power (incl. power for circuits, ADC, cooling system, etc.) consumed by an active RU. \( P_{BH} \) denotes the power consumption of the backhaul associated with the active RU. They are assumed to be constants. We define the operational power of an active RU as \( P_o = P_{RU} + P_{BH} \), RU \( n \) will not serve users if it is in sleep mode and thus \( P_{TX,n} = 0 \). The power consumption \( P_{sleep} \) is usually much lower than \( P_o \).

Colored dots denote users. \( K \) single-antenna users are scheduled in each slot. They are uniformly and independently distributed within the network. Users with same color request the same file \( f \) and form the multicast group \( \mathcal{G}_m \). Let \( M \) denote the number of different multicast groups and assume that each scheduled user requests at most one file, so \( \mathcal{G}_i \cap \mathcal{G}_j = \emptyset \) for all \( i \neq j \) and \( \sum_{m=1}^{M} |\mathcal{G}_m| \leq K \) hold. The \( m \)-th multicast group \( \mathcal{G}_m \) is cooperatively served by a cluster of RUs, denoted by \( \mathcal{C}_m \) with \( \mathcal{C}_m \subseteq \mathcal{N} \). Unlike the multicast group \( \mathcal{G}_m \), which is fixed based on users’ requests, the clustering \( \mathcal{C}_m \) is subject to be dynamically optimized by the CP, and they can overlap with each other, i.e., \( \mathcal{C}_i \cap \mathcal{C}_j \) is not necessarily empty.

Let \( s_m \) be a transmitted symbol of file \( f \) requested by users in multicast group \( \mathcal{G}_m \), with normalized power \( \mathbb{E}\{|s_m|^2\} = 1 \) \( \forall m \in \{1, ..., M\} \). All RUs in cluster \( \mathcal{C}_m \) transmit it cooperatively. For user \( k \in \mathcal{G}_m \), let \( \mathbf{h}_{k,n} \in \mathbb{C}^{L \times 1} \) denote the channel vector between RU \( n \) and itself. Thus, \( \mathbf{h}_k = [\mathbf{h}_{k,1}, \mathbf{h}_{k,2}, ..., \mathbf{h}_{k,n}, ..., \mathbf{h}_{k,N}]^H \in \mathbb{C}^{NL \times 1} \) is the aggregated channel vector from all RUs to it. Global Channel State Information (CSI) is assumed to be available at the CP and the block-fading channel remains constant within each scheduling slot but changes from one to another. Due to the multicast scenario, content-centric [8] beamforming optimization and construction are performed at the CP, such that it is based on the transmitted content, or equivalently, its served multicast group. Assume that at RU \( n \), the beamforming vector for multicast group \( \mathcal{G}_m \) is \( \mathbf{v}_{m,n} \in \mathbb{C}^{L \times 1} \). Thus, \( \mathbf{v}_{m,n} = \mathbf{0}_{L \times 1} \), or equivalently \( P_{TX,n,f(m)} = \mathbb{E}\{|\mathbf{v}_{m,n}|^2\} = 0 \) means that RU \( n \) is not involved in cluster \( \mathcal{C}_m \) to serve \( \mathcal{G}_m \), i.e., \( n \notin \mathcal{C}_m \). Moreover, \( \mathbf{v}_{m,n} = \mathbf{0}_{L \times 1} \) \( \forall m \), or equivalently \( P_{TX,n} = \sum_{m=1}^{M} P_{TX,n,f(m)} = \sum_{m=1}^{M} \mathbb{E}\{|\mathbf{v}_{m,n}|^2\} = 0 \) means that RU \( n \) is in sleep mode without serving any multicast group. The aggregated beamforming vector from all RUs for multicast group \( \mathcal{G}_m \) is \( \mathbf{v}_m = [\mathbf{v}_{m,1}^H, \mathbf{v}_{m,2}^H, ..., \mathbf{v}_{m,n}^H, ..., \mathbf{v}_{m,N}^H]^H \in \mathbb{C}^{NL \times 1} \). Hence, the SINR at user \( k \) can be expressed as

\[
\text{SINR}_k = \frac{\|\mathbf{h}_k^H \mathbf{v}_m\|^2}{\sigma^2_k + \sum_{i \neq m} \|\mathbf{h}_k^H \mathbf{v}_i\|^2}, \quad k \in \mathcal{G}_m,
\]

where \( \sigma^2_k \) denotes variance of the i.i.d additive complex Gaussian noise with zero mean at user \( k \).
B. File and Cache Model

Totally $F$ files are assumed to be potentially requested. All of them have normalized length 1 and are available at the CP. However, they have different probabilities (i.e., popularity) to be requested. Without loss of generality, we index the files in the order from the most to the least popular ones, such that the most popular file has index $f = 1$ and the least popular file has index $f = F$. The popularity is modeled by Zipf distribution [5], i.e., the probability of file $f$ being requested is

$$P(f) = \frac{f^{-\alpha}}{\sum_{j=1}^{F} j^{-\alpha}}, \quad f = \{1, 2, ..., F\}. \quad (3)$$

Parameter $\alpha$ is related to the skewness of the distribution, larger $\alpha$ makes more biased popularity distribution.

Similar to [4], [7], [8], an uncoded caching scheme is adopted. All RUs have a cache of memory $S$, and caches the same most popular files until it is full. Hence, files with index smaller than or equal to $S$ will be cached. We assume that the cached contents are fixed, the caching placement problem will not be addressed, which is beyond of the scope of this paper. Compared to the coded caching scheme, where different RUs cache different fractions of a file, uncoded caching has lower content diversity but can achieve higher spatial diversity, which potentially leads to less power consumption but more load on the backhaul. Hence, the load handling is a significant issue. The traffic load on each active backhaul is balanced according to its capacity. The problem (6) guarantees the QoS of each scheduled user should be met and the traffic load on each active backhaul does not exceed its capacity: When RU $n$ is active and involved in cluster $C_m$ (i.e., $\|v_{m,n}\|^2 > 0$, or equally $\|v_{m,n}\|^2 \leq 1$), and the requested file $f_m$ is not cached (i.e., $c_{f_m,n} = 0$), at least $\log_2(1 + \Gamma_n)$ capacity is required (Gaussian codebook is assumed) at RU $n$ for $G_m$.

The objective function (5) can be equivalently written as

$$\min_{\{v_{m}\}} \sum_{m=1}^{M} \|v_{m}\|^2 + \sum_{n=1}^{N} \Delta P \left( \sum_{m=1}^{M} \|v_{m,n}\|^2 \right) + \epsilon N P_{\text{sleep}}, \quad (8)$$

where $\Delta P = \epsilon (P_s - P_{\text{sleep}})$. Since the last term is a constant, it is sufficient to set the sum of first and second term as the equivalent objective function.

**Remark:** Compared to problems considered in [7]–[9] for F-RAN, the objective of proposed problem (5) makes deactivation of some RUs possible: If the operational power saved by deactivating a RU can compensate the increased transmission power among the others (since the array gain is decreased), and the remaining RUs can still satisfy the QoS and traffic constraints, this RU will be forced to be switched off. Namely, the decrease in the second term of (8) must lead to an increase of the first term, and vice versa. Note that the constraint (7) makes the proposed problem more stringent than the case considered in [7]–[9], where only sum capacity is considered. Hence, without this more stringent constraint, the total power consumption of the solution for the proposed algorithm must be at least not higher than that of previous works, since we take the possibility of deactivation into account. While this additional constraint might result in higher transmission power, the results of this work might consume more total power in some cases, as shown later in Sec. IV. However, the solution of proposed problem guarantees that the traffic load on each active backhaul is balanced according to its capacity.

According to (5)-(7) and the descriptions above, the RU selection, clustering among active RUs and beamforming depend on the requested and cached files, target QoS, individual backhaul capacities and the channel. For different scheduling slots, the above parameters (except for backhaul capacities) change dynamically and independently. It is necessary to find an efficient algorithm to solve the problem. Then, the RU selection and clustering formulation are implicitly optimized and determined by the solution, i.e., $P_{TX,n}(m) = \sum_{m=1}^{M} \|v_{m,n}\|^2$ and $P_{TX,n}(m) = \|v_{m,n}\|^2$ respectively.

III. OPTIMIZATION ALGORITHM

Both the objective function (8) and constraints (6)-(7) are generally non-convex. Moreover, the $l_0$-norms in (7) and (8) make the corresponding function be step-like and similar to a Mixed Integer Non-Linear Programming (MINLP) problem, which is NP-hard. Hence, in the following 2 subsections, we adopt two techniques to convexify and simplify them.
A. SDR: Convexification of SINR constraints (6)

From the CP’s point of view, the downlink can be regarded as a virtual multi-antenna multicast system, which is similar to the model considered in [15], where a so-called Semi-Definite Relaxation (SDR) technique is proposed for the convexification of a similar SINR constraint. We also adopt this idea here.

Let $V_m = v_m v_m^H$ and $H_k = h_k h_k^H$, $\forall m, k$, where both $V_m, H_k \in \mathbb{C}^{N_L \times N_L}$ are positive semidefinite matrices. Then define a matrix $J_n$ at RU $n$ as $J_n = \text{Diag}(0_{(n-1)N_L \times 1}^H, 1_{N_L \times 1}, 0_{(N-n)N_L \times 1})$. Thus, we have $\|v_m\|^2 = \text{tr}(v_m v_m^H)$, $\|v_m\|^2 = \text{tr}(v_m v_m^H)$, and $\|h_m\|^2 = \text{tr}(H_m H_m^H)$. So the problem can be equivalently written as:

$$\min_{(v_m)_{m=1}^M} \sum_{m=1}^M \text{tr}(v_m v_m) + \sum_{n=1}^N \Delta P \left| \sum_{m=1}^M \text{tr}(v_m J_n) \right|_0$$  \hspace{1cm} (9)

s.t. $\Gamma_m \left( \sigma_k^2 + \sum_{i \neq m} \text{tr}(H_i H_k) \right) - \text{tr}(H_m H_k) \leq 0, \forall k \in G_m \lor G_m';$

$$V_m \succeq 0, \forall m = \{1, 2, ..., M\};$$  \hspace{1cm} (10)

rank $(V_m) = 1, \forall m = \{1, 2, ..., M\}$,

$$\sum_{m=1}^M (1 - c_{f,n,m}) |\text{tr}(v_m J_n)|_0 \log_2 (1 + \Gamma_m) \leq C_{BH,n}, \forall n \in N.'$$  \hspace{1cm} (13)

The first term of (9), together with (10) and (11) form a standard Semi-Definite Programming (SDP) problem. The SDR technique is to drop the non-convex rank-1 constraint (12) and solve the remaining SDP problem. If the obtained $V_m$ has rank 1, the EigenValue Decomposition (EVD) can be used to obtain the corresponding optimal beamforming vector $v_m$. Otherwise randomization and scaling method is used to generate a suboptimal solution. Details can be found in [15]. However, these procedures cannot be applied directly here, since the non-convex and discrete $\ell_0$-norm still exists in the second term of (9) and the backhaul constraint (13).

B. Re-weighted $\ell_1$-norm: Convexification of $\ell_0$-norm

We approximate the non-convex and discrete $\ell_0$-norm by a re-weighted $\ell_1$-norm, which is linear, continuous and convex. This idea is proposed in a compressive sensing literature [16].

Consider minimizing a linear combination of $\ell_0$-norm of each element in vector $x_N \times 1$, i.e., $\min \sum_{n=1}^N a_n |x_n|_0$, and $x$ should satisfy some constraints. The $\ell_0$-norm of $x_i$ is iteratively approximated as

$$|x_i^{(t+1)}|_0 \approx w_i^{(t+1)} x_i^{(t+1)}, \text{ with } w_i^{(t+1)} = \frac{1}{x_i^{(t)} + \tau},$$  \hspace{1cm} (14)

where $t$ denotes the iteration number, $w_i$ is the re-weighted coefficient of $x_i$, and $\tau$ is the stability parameter. Firstly we drop the superscript $(t)$ and $(t+1)$ to explain such approximation. Now we have $|x_i|_0 \approx w_i x_i$. When $\tau > x_i$, the approximation of $\ell_0$-norm is close to 1. Contrarily, the approximation quickly approaches 0 for $\tau \ll x_i$. Therefore, $\tau$ can be regarded as a threshold parameter that determines whether $x_i$ is on (1) or off (0). By carefully choosing the value of $\tau$, this continuous and linear approximation can capture the behavior of discrete non-convex $\ell_0$-norm.

The superscripts in (14) denote the iterative re-weighted procedure. In the $t$-th iteration, the approximated minimization problem is solved, then $x_i^{(t)}$ can be obtained and $w_i^{(t+1)}$ used for the next iteration is updated. If the solved $x_i^{(t)}$ decreases in iteration $t$, compared to the previous one, it would have higher weight $w_i^{(t+1)}$ in the next iteration, then its value will be forced to further reduce and encouraged to drop below the threshold $\tau$. By continuing such re-weighted procedure iteratively, some elements of $x$ will be finally forced to be close to 0, and the remaining elements can still satisfy all constraints. More details of such approximation can be found in [16]. By adopting this idea, some RUs will be forced to be deactivated and some active RUs will be excluded from participating in serving some multicast groups, as the iteration goes on. Therefore, the NP-hard MINLP problem can be avoided.

C. Reformulation and solution of the original problem

By combining the two techniques introduced above, the original problem can be reformulated as follows:

$$\mathcal{P}_{\text{Ref}}^{(t)} : \min_{(v_m)_{m=1}^M} \sum_{m=1}^M \text{tr}(v_m v_m) + \sum_{n=1}^N \Delta P \sum_{m=1}^M \text{tr}(v_m J_n)$$  \hspace{1cm} (15)

s.t. $\Gamma_m \left( \sigma_k^2 + \sum_{i \neq m} \text{tr}(H_i H_k) \right) - \text{tr}(H_m H_k) \leq 0, \forall k$;

$$\sum_{m=1}^M R_{m,n} w_{m,n} \text{tr}(v_m v_m) - C_{BH,n} \leq 0, \forall n \in N;'$$  \hspace{1cm} (17)

$$V_m \succeq 0, \forall m = \{1, 2, ..., M\},$$  \hspace{1cm} (18)

where

$$u_{m,n} = \frac{1}{\sum_{m=1}^M \text{tr}(v_m v_m) + \tau_1}, \quad w_{m,n} = \frac{1}{\text{tr}(v_m v_m) + \tau_2},$$  \hspace{1cm} (19)

are re-weighted coefficients in $t$-th iteration for the approximation of $\sum_{m=1}^M \text{tr}(v_m v_m)$ and $\text{tr}(v_m v_m)$, which determine whether RU $n$ is selected to be active and involved in cluster $C_m$, respectively. $\tau_1$ and $\tau_2$ denote the respective threshold. $R_{m,n} = (1 - c_{f,n,m}) \log_2 (1 + \Gamma_m)$ is constant in each slot and known to the CP. We drop the rank-1 condition (12) since SDR technique is used, which will be compensated at the last step of the algorithm below. Problem $\mathcal{P}_{\text{Ref}}^{(t)}$ consists of only a linear objective function, $K + N$ linear inequality constraints and $M$ positive-semidefinite constraints, which is a standard SDP problem and can be efficiently solved by many solvers, such as SDPT3 and Sedumi.

**Initialization:** Since the re-weighted coefficients depend on the solution of problem in the previous iteration (19), in order to obtain $V_m^{(t)}$, we solve an initial problem $\mathcal{P}_{\text{Init}} = \mathcal{P}_{\text{Ref}}$ as

$$\mathcal{P}_{\text{Init}} : \min_{(v_m)_{m=1}^M} \sum_{m=1}^M \text{tr}(v_m v_m)$$  \hspace{1cm} (20)

s.t. $\Gamma_m \left( \sigma_k^2 + \sum_{i \neq m} \text{tr}(H_i H_k) \right) - \text{tr}(v_m v_m) \leq 0, \forall k$;

$$V_m \succeq 0, \forall m = \{1, 2, ..., M\},$$  \hspace{1cm} (22)
such that only QoS is guaranteed without considering the operational power (second term of (15)) and individual backhaul capacity (17). $w_{m,n}^{(1)}$ can be computed based on $V_m^{(0)}$.

Then from the first iteration, the second term of (15) and backhaul constraint (17) are added, two re-weighted coefficients are amended gradually in each iteration. RU $i$ might be regarded as being switched off (deactivation) gradually, when its transmission power $P_{TX,i}$ falls below $\tau_1$. An active RU $j$ might be gradually excluded from cluster $C_m$, when its corresponding power for this multicast group $P_{TX,i,j}(m)$ falls below $\tau_2$. The objective of minimization problem (5) and constraints (6)-(7) ensure that such deactivation and exclusion happen, only when the resultant total power consumption can be decreased, the clustering can meet the QoS of each user, and the load on each backhaul does not exceed its capacity. The overall algorithm is listed below. At last, EVD or searching and scaling method is performed to obtain the beamforming vector $v_m$, more details can be found in [15].

Algorithm 1: Iterative Optimization Steps

1. Initialization: Solve the standard SDP problem $P_{Init}$ to obtain $V_m^{(0)}$. Compute $w_{m,n}^{(1)}$ and $w_{m,m}^{(1)}$ based on (19), $\forall m,n$. Construct problem $P_{Ref}^{(1)}$, set $t ← 1$.
2. repeat
   3. Solve standard SDP problem $P_{Ref}^{(t)}$ to obtain $V_m^{(t)}$
   4. Compute $w_{m,n}^{(t+1)}$ and $w_{m,m}^{(t+1)}$ based on (19), $\forall m,n$.
      Then construct problem $P_{Ref}^{(t+1)}$, set $t ← t + 1$.
   5. until Convergence or reaching max iteration number
6. if $\text{rank}(V_m^{(t)}) = 1$ then
   7. Perform EVD to obtain optimal $v_m$.
8. else
   9. Use Gaussian randomization and scaling [15] to obtain the approximate solution $v_m^*$. 

IV. SIMULATION RESULTS

In this section, we provide some simulation results based on the proposed algorithm. The model in Fig. 1 is considered, with the simulation parameters listed in Table I. Most results are based on them unless otherwise stated.

At first we compare the total power consumption of the proposed algorithm with the case discussed in [7] and [8] (Benchmark), where only transmission power and total backhaul capacity cost are considered. For different realizations (scheduled intervals) of the simulation, different groups of users with different request submissions will be scheduled. The CP runs the optimization algorithm accordingly and feedbacks the optimized results to RUs. Among all realizations simulated, we select two representative scenarios and show the results.

Representative Scenario 1 (Abundant local resources): In this interval, after 12 scheduled users submit their requests according to Zipf distribution (3), the CP finds that only 4 different files are requested, and 3 of them have already been cached at RUs. In such scenarios, most requested files are available at local caches without needing to be delivered from the CP via backhauls. Hence, there are comparatively sufficient caching and backhaul capacity resources. For the two algorithms, the transmission power of each RU $P_{TX,n}$ is recorded and illustrated in Fig. 2(a) and (b). The total power consumed by transmission $\frac{1}{2}P_{TX,tot} = \frac{1}{2}\sum_{n=1}^{N}P_{TX,n}$ and the total operational power $P_{tot} = \sum_{n=1}^{N}P_{TX,n} + \sum_{n=1}^{N}P_{RU,n} + \sum_{n=1}^{N}P_{Sleep,n}$ are shown in Fig. 2(c). The total power consumption $P_{tot}$ is $\frac{1}{2}P_{TX,tot} + P_{RU,tot}$ is shown in Fig. 2(d).

Analysis of Fig. 2: We explain the results produced by the proposed algorithm at first. (a), when most requested files have been cached, abundant caching resources result in low traffic load on backhauls. Thus, there are sufficient backhaul capacities for the delivery of uncached files, and forming larger clusters for more cooperation to decrease the transmission power is not difficult. Hence, switching off some RUs and backhauls for saving power is possible, and the QoS can still be guaranteed. The transmission power of 3 RUs fall below $-170$ dBm in less than 10 iterations, these RUs are determined to be switched off by the CP. Therefore in (c), the total operation power $P_{RU,tot}$ drops. The total power consumed by transmission $\frac{1}{2}P_{TX,tot}$ increases in the first 10 iterations. The reason is as follows: We start with solving the initial problem $P_{Init} (20)-(22)$, where individual capacity constraint and the operational power do not exist. In the next iterations, they are added and the re-weighted coefficients are amended in each iteration. Hence, the transmission power is increased due to: 1. individual backhaul capacity constraint; 2. less potential aggregated array gain resulting from deactivation.

Now we check the results generated by Benchmark scheme. Since only the transmission power and total backhaul capacity cost are considered, all RUs are selected to be active for increasing the potential aggregated array gain to reduce the transmission power, so the total operational power keeps the same, as shown in (c). Moreover, we see that the transmission

<table>
<thead>
<tr>
<th>TABLE I. SIMULATION PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of RU (Hexagonal Cell): $N$</td>
</tr>
<tr>
<td>Equipped antennas of each RU: $L$</td>
</tr>
<tr>
<td>Distance between adjacent RUs: $d_{RU}$</td>
</tr>
<tr>
<td>Transmit Antenna Gain</td>
</tr>
<tr>
<td>Additional Power consumption: $P_{RU}$</td>
</tr>
<tr>
<td>Power consumption in sleep mode: $P_{Sleep}$</td>
</tr>
<tr>
<td>Power Amplifier Efficiency: $\epsilon$</td>
</tr>
<tr>
<td>Active Backhaul Power consumption: $P_{BH}$</td>
</tr>
<tr>
<td>Total number of users: $K_{tot}$</td>
</tr>
<tr>
<td>Number of scheduled users per interval: $K$</td>
</tr>
<tr>
<td>Background noise</td>
</tr>
<tr>
<td>SGPP LTE-A path loss model</td>
</tr>
<tr>
<td>Log-normal shadowing</td>
</tr>
<tr>
<td>Rayleigh small scale fading</td>
</tr>
<tr>
<td>Bandwidth: $B$</td>
</tr>
<tr>
<td>Target SINR at each user: $\gamma_{Ref} = 1 \forall n$</td>
</tr>
<tr>
<td>Total number of files: $F$</td>
</tr>
<tr>
<td>Skew parameter of zipf distribution: $\alpha$</td>
</tr>
<tr>
<td>Cache Memory: $S$</td>
</tr>
<tr>
<td>Individual backhaul capacity: $C_{BH,n} = C_{BH} \forall n$</td>
</tr>
<tr>
<td>Threshold parameters in (19): $\tau_1, \tau_2$</td>
</tr>
<tr>
<td>Maximum iteration number: $N_{max}$</td>
</tr>
</tbody>
</table>
power of each RU $P_{TX,n} = \sum_{m=1}^{M} ||v_{m,n}||^2_2$ and the total power consumed by transmission $1/2 P_{TX, tot} = 1/2 \sum_{n=1}^{N} P_{TX,n}$ also keep almost the same in each iteration, see (b) and (c). The reason is as follows: The Benchmark scheme in [7] and [8] also starts with a initial problem, without considering total capacity cost. This constraint is added from the second iteration. In Representative Scenario 1, the total backhaul capacity is not the bottleneck since most files are available in caches. Therefore, adding total capacity cost will not significantly increase the transmission power in the remaining iterations. The total power consumption is shown in (d), which can be significantly reduced due to deactivation with the proposed algorithm. However, as we said in Remark on page 3, the total power consumption with the proposed algorithm is not always less than that of the Benchmark, since we solve a more stringent problem to balance the traffic load on each active backhaul, which might increase the transmission power. Hence, in scenarios where less or even no RUs and backhauls can be deactivated, it might be higher, as we show next.

Analysis of Fig. 3: It shows the results of Representative Scenario 2, where most requested files have to be delivered via backhauls due to the lack of cached files, resulting in heavy traffic load on them. The backhaul capacity becomes a bottleneck, and forming larger clusters for more cooperations also becomes difficult. In such scenarios, with the proposed algorithm, only RU 6 and its backhaul are switched off after 16 iterations, as shown in (a). In (b), all RUs are still active with the same reason described above. (c) shows that the transmission power of the both algorithms increase after adding capacity constraints from the second iteration. However, the increasing amount of the proposed algorithm is much higher, due to balancing the traffic load on very limited backhaul resources. The increased transmission power is even larger than the decreased operational power from the deactivation of RU 6. Hence, the total power consumption is higher, as shown in (d). Although more total power is consumed, we are going to show that the traffic load on each backhaul is balanced.

Analysis of Fig. 4: When more files have to be delivered via backhauls due to the lack of cached files, the issue of traffic handling can not be ignored. Hence, we consider Scenario 2 described above. Without loss of generality, we number 3 cached files as $f(1)$, $f(2)$ and $f(3)$, respectively, which are requested by multicast groups 1, 2 and 3. Delivery of the other 4 uncached files consumes backhaul resources. Since $C_{BH} = 100$ Mbps $\approx 2 \times B \log_2(1+\Gamma)$, at most 2 data streams can be supported by each backhaul. We record the transmission power $P_{TX,n,f(m)} = ||v_{m,n}||^2_2$ for multicast group $m$ at RU $n$, for all $m,n$, in each iteration, which formulates the clustering at each RU. This figure shows the clustering of RU 2 and 5, which are both selected to be active, according to Fig. 3(a). From (a) and (c), after about 13 iterations, the transmission power of two multicast groups fall below $-150$ dBm, meaning that the RU will not be involved in the corresponding clusters. Thus, the proposed algorithm formulates the clustering such that exactly 2 data streams of uncached files are transmitted at RU 2 and 5. However, (b) and (d) show that with the Benchmark scheme, RU 2 has to support 3 data streams of uncached files and RU 5 supports only 1. Although this is optimal if all RUs share a common backhaul resource, which is not always the case in practice, where individual backhaul (e.g. optical fiber) capacities are usually predetermined. Hence, the clustering formulated by the Benchmark scheme may cause traffic congestion or resource waste. Moreover, for cached files without consuming backhaul resources, both algorithms retain all active RUs in their clusters, so as to decrease the transmission power by increasing the aggregated array gain. This result is in consistence with the theory proposed in [7].

Analysis of Fig. 5: Then we fix all the other parameters in Table I, except $C_{BH}$ and $S$, in order to compare the average total power consumption for different individual backhaul capacities 1 and cache memories. We set a large number of realizations and run both algorithms for solving each resultant problem. Some of them are infeasible. This is due either to many uncached files being requested, or to small individual backhaul capacities and cache memories leading to little cooperation possibility to counteract the bad channel condition.

\[1\] The Benchmark scheme only considers total capacity, thus we consider the average individual capacity for a fair comparison.
the QoS of all users cannot be fulfilled simultaneously. Firstly we show this outage probability of the proposed algorithm based on 500 realizations in (a). With more cache memory and larger backhaul capacity, outage probability can be reduced due to more cooperation being possible. Then we average the results of 300 feasible realizations and obtain (b). By increasing either individual backhaul capacity or cache memory, the power consumption for both algorithms decrease due to more cooperation becoming possible. Moreover, larger $C_{BH}$ or $S$ makes local resources more abundant, thus realizations similar to Representative Scenario 1 become more probable. more RUs are possible to be switched off with the proposed algorithm, leading to far less total power consumption than the Benchmark scheme. When $C_{BH}$ or $S$ is smaller, it is likely that more realizations are similar to Representative Scenario 2. Less or even no RUs can be switched off, and traffic handling on active backhauls becomes an important issue. Such load balancing makes the solution of proposed algorithm consume more total power than that of the relaxed problem in Benchmark. When $C_{BH}$ and $S$ get large enough, the solution of both algorithms enter the saturation region, since the current resources has been sufficient to allow full cooperation for most requests, increasing local resources further cannot further increase the possibility of cooperation in order to decrease the power significantly. With the proposed algorithm, it is also not possible to deactivate more RUs. However, it shows that the power consumption in this region can be greatly reduced with the proposed algorithm, due to the large operational power saved by deactivation. Moreover, it should be noted that increasing the cache memory is usually much easier and cheaper compared to increasing the backhaul capacity.

V. Conclusion

In this paper, by reformulating a non-convex NP-hard MINLP problem as a standard SDP problem, an efficient algorithm is proposed to jointly optimize the RU selection, clustering formulation and beamforming, in the downlink of Cache-Enabled F-RAN at the system level. It can adaptively deactivate some RUs for saving the operational power and still guarantee the QoS. Meanwhile, the load on each active backhaul is balanced according to its capacity, and it minimizes the total power consumption with the optimized beamforming vectors. Simulation results show that if most request files has been cached, RU deactivation should be considered for lower operational power consumption. Otherwise the load balancing becomes an important issue, for avoiding the traffic congestion and resource waste. Increasing the cache memory for higher content hitting rate is an economical way to achieve greener network, through optimization with the proposed algorithm.

REFERENCE


