Joint Offloading Framework to Support Communication and Computation Cooperation

Kezhi Wang, Member, IEEE, Kun Yang, Senior Member, IEEE, Cunhua Pan, Member, IEEE, Jiangzhou Wang, Fellow, IEEE,

Abstract

In order to support communication and computation cooperation, we propose MC-RAN architecture, which consists of mobile cloud (MC) as the computation provision platform and radio access network (RAN) as the communication interface. The MC-RAN aims to undertake the following tasks: (1) to increase user equipments’ computing capacity by triggering offloading action, especially for those UEs which cannot complete the computations locally; (2) to reduce the energy consumption for all the UEs by considering limited computing and communication resources; (3) to decrease the whole MC-RAN’s energy consumption. To achieve the above tasks, uplink offloading framework is proposed. To reduce the signaling between UE and MC-RAN, decentralized local decision algorithm (DLDA) is firstly proposed for each UE to estimate the local execution and offloading energy and decide if offloading is in its interest. Then, centralized access control algorithm (CACA) is conducted by MC-RAN with the global information to decide the offloading set. Based on the offloading set, centralized resource allocation algorithm (CRAA) is employed to optimize the offloading power of each UE, corresponding receive beamforming vectors in MC-RAN and the active RRH and fronthaul pairs, with the objective of minimizing the energy consumption of the whole system including MC-RAN and UEs.

Index Terms - Communication and Computation Cooperation, Joint Offloading Framework, Resource Allocation, MC-RAN.

Kezhi Wang and Kun Yang are with the School of Computer Sciences and Electrical Engineering, University of Essex, CO4 3SQ, Colchester, U.K, (e-mails: {kezhi.wang, kunyang}@essex.ac.uk).

Cunhua Pan is with School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K, (e-mails: C.Pan@qmul.ac.uk)

Jiangzhou Wang is with School of Engineering and Digital Arts, University of Kent, Canterbury, Kent, CT2 7NZ, U.K, (e-mail: J.Z.Wang@kent.ac.uk).
I. INTRODUCTION

Nowadays, user equipments (UEs) like smartphones and hand-held terminals are enjoying increasing popularity. More and more resource-hungry applications such as high definition video gaming and virtual reality applications are developing and coming into play in our mobile devices. However, due to limited resources in terms of battery, CPU, storage, etc, UEs are struggling in keeping up with the development of the resource intensive applications.

Fortunately, mobile cloud computing (MCC) [1]–[3] was proposed to make UEs with computing intensive tasks be able to offload computations to the cloud to increase UEs’ experience and prolong their battery life. Several cloud offloading platforms have been proposed, such as ThinkAir [1], which can migrate the applications from the mobile devices to the cloud. In [4], a game theoretic approach has been proposed to make the decision for each UE about where to execute the computation. However, the above mentioned MCC systems used the normal cloud, such as Amazon elastic compute cloud (EC2) [5], to execute the offloaded computations. If UEs need cloud’s help, they have to send their instructions, along with the data all the way via the Internet to the cloud. This is not beneficial to the UEs with high communication reliability and low latency requirement.

Some papers proposed to set mobile cloud in mobile network [6]. This brings opportunities to the mobile operator to go beyond just a pipe provider to become the cloud service provider to the UEs. Moreover, the mobile operator has the potential to provide better cloud services to the UE than the normal cloud service provider. This is because mobile operator not only holds the information from the cloud, but also has the wireless channel status, so that they can better jointly optimize both networks and cloud. [2] has studied that different wireless bandwidths may have different impacts on UE’s offloading strategy. However, they only considered the situation where there is one UE conducting offloading. In wireless access channel, whether one UE decides to offload or not will induce interference to other UEs and affect other UEs’ decisions, as the interference may deteriorate other UEs’ signals. Some UEs may increase their transmission power to guarantee the high data rate and reliable transmissions. This action may in turn lead to the failure of the other UE’s packet transmissions. Moreover, it is anticipated that with the popularity of Internet of things (IoT), almost 50 billion objects will be connected to mobile networks by 2020. These IoT objects normally have small sizes and also the limited computing resources, and they may all need to access to the cloud via wireless networks for help. This will
bring both challenges and opportunities to the mobile operator.

Another challenge that mobile operator may face is the limited amount of wireless communication resources they possess, such as bandwidth [7]–[9]. This may become worse in case multiusers are offloading at the same time, which will cause severe interference and lead to inefficient use of wireless resources when there is no proper central coordinating control mechanism. Large amount of energy consumption will be incurred, thereby reducing mobile operators’ profits.

To overcome the above-mentioned problems, small cell networks have been proposed to improve the system capacity and combat interference [10]. However, with less computational capacity than normal macro cell, small cell networks may not be able to handle the signal processing in required time intervals and therefore fail to meet the quality of service (QoS) requirement of wireless transmissions. For instance, small cells may not be able to complete the processing of fast fourier transformation (FFT), forward error correction (FEC) in certain QoS, due to lack of sufficient computational resources and then may lead to the failure or error in the data transmissions.

Thus, the cloud-based network infrastructure, i.e., cloud radio access network (C-RAN) has emerged [11]–[13]. C-RAN moves most of the computing related tasks to central baseband unit (BBU) pool and distribute low complexity remote radio heads (RRHs) to the whole cell. BBU pool is composed of multiple computing servers, which not only provide computing resources to the networks as required but also put some of the computing servers into sleep mode for the purpose of energy saving [14]. In [15], [16], the recent advances and challenges of C-RAN were surveyed. Due to the centralized signal management, signals from other UEs can be coordinated and are no longer considered as detrimental interference but useful signals. Because of the centralized processing feature in C-RAN, it is of much interest to set mobile cloud right next to the C-RAN, managed by the mobile operator. In such a case, computing resources and communication resources may be monitored and processed together and bring not only good service to the UEs but also profit boosting to the mobile operator. Furthermore, computing resources are normally limited by the number of available physical machines in cloud and contribute to a large amount of the energy consumption. Therefore, how to manage and allocate the computing resources properly is a very challenging problem [14], [17], [18].

In this paper, by taking both advantages of C-RAN and mobile cloud, mobile cloud-radio access network (MC-RAN) architecture is proposed to support communication and computation cooperation. Similarly to C-RAN, MC-RAN is composed of mobile cloud (MC) as the comput-
ing resource provision platform and radio access network (RAN) as the wireless transmission interface. However, different from traditional C-RAN, mobile clones are considered in MC-RAN. Mobile clone is implemented by cloud-based virtual machine, which can be seen like the UE. Each UE will have a specific mobile clone, which holds the same operating system, software and configuration as its corresponding UE. UE that has the computing intensive tasks can offload the task to its own mobile clone to conduct the tasks. In MC-RAN, BBU is also implemented in the form of virtual machine in mobile cloud, which is in charge of signal processing related tasks, such as receiving the information from the UE in the uplink or returning the execution results back to UE in the downlink.

The MC-RAN aims to achieve the following targets:

- To increase UEs’ computing capacity and experience by enabling UEs to offload computation tasks to the mobile cloud. Special attention is paid to UEs which cannot complete the tasks locally. In other words, those UEs are given high priority when offloading process is triggered.
- To reduce energy consumption of all the UEs by accepting the offloading requests from as many UEs as possible, under current available resources, i.e., computing resource in mobile cloud, and communication resource in wireless networks.
- To reduce the whole MC-RAN’s energy consumption by properly managing the communication and computing resources. Further, switch off the redundant hardware equipment, i.e., RRH and fronthaul pairs, to save the system’s energy when necessary.

To achieve the above goals, uplink offloading framework is proposed. Based on the limited communication and computation resource, access control is first conducted, with the target of accepting as many offloading UEs as possible. More importantly, offloading priority is given to UEs which cannot complete the task locally. Then, resource allocation is conducted and coordinated by MC-RAN, by minimizing the energy consumption of the whole system, including MC-RAN and UE. The main contributions of this paper are summarized as follows:

- To reduce the signaling overhead and traffic between UE and MC-RAN, decentralized local decision algorithm (DLDA) is first proposed for each UE to estimate its transmission energy and local execution energy and then decide if offloading is needed. Estimation model of energy consumption without knowing other UE’s decision and corresponding interference is given. We show that minimizing the energy consumption of each UE with the single antenna
is equivalent to minimizing its power. Then, we show that UE can only trigger offloading action if the maximal available power is larger than the minimal required transmission power.

• To tackle the obstacle that each UE itself does not have the global information when conducting offloading, central access control algorithm (CACA) in MC-RAN is proposed to make the decision on which UE can be allowed to offload.

• Uplink-downlink duality is employed to establish a link between offloading action from UE side in the uplink and the available computing and communication resource from MC-RAN. The non-smooth indicator constraint in MC-RAN is approximated as a non-convex function and the successive convex approximation (SCA) is applied to deal with this non-convexity. Also, auxiliary variables are applied to make the problem feasible to be solved in MC-RAN.

• Given the feasible offloading set of UEs from CACA, central resource allocation algorithm (CRAA) is proposed to optimize the offloading power of each UE, the corresponding receive beamforming vectors and the active RRH and fronthaul pairs in MC-RAN, with the objective of minimizing the overall system energy consumption. SCA based iterative algorithm is developed to switch off redundant hardware infrastructures. Simulation results show that with the help of MC-RAN, most of the UEs which previously may not be able to execute the tasks locally now can not only complete the task in time, but also enjoy high computation resource in cloud. Also, large amount of energy consumption can be saved by using the proposed algorithm.

The remainder of this paper is organized as follows. Section II introduces the whole architecture design and system model of our newly proposed MC-RAN system. Section III presents energy consumption model and problem formulation. Section IV and V introduce how we do the access control to the offloading candidates, based on current available resource, including DLDA and CACA. Section VI presents CRAA to do resource allocation based on the offloading set obtained before. Simulation results are presented in Section VII, followed by conclusions in Section VIII.

Notations: \( \mathbb{E}(x) \) denotes the expectation of \( x \), \( \mathcal{CN}(0, \sigma^2 I) \) denotes the complex Gaussian distribution with zero mean and covariance vector \( \sigma^2 I \), ‘s.t.’ is short for ‘subject to’, the log function is the logarithm function with base 2, \( | \cdot | \) denotes the size of the set, \( | \cdot |_1 \) is the indicator function defined in (17) and \( || \cdot || \) stands for either the Euclidean norm of a complex vector or the magnitude of a complex number, depending on the context.
II. Architecture Design and System Model

A. Architecture Design

In this section, we introduce the MC-RAN architecture by taking advantages of the popular C-RAN and MCC techniques, as shown in Fig. 1. MC-RAN is composed of the mobile cloud and RAN. Mobile cloud hosts both mobile clone (i.e., service computing unit) and BBU (i.e., communication computing unit). The allocation of the computational resources in BBU pool was discussed in [14], where the capacities of BBU can be dynamically adjusted to handle dynamic UE traffic and channel states. However, in this paper, it is assumed that the computational resources in mobile clones and BBUs can be jointly allocated according to the status of the wireless networks and UEs.

Fig. 1. MC-RAN architecture.

It is assumed that there is a MC-RAN network with $N$ UEs, each with one antenna, and $J$ RRHs, each has $K$ antennas connecting to the BBU pool through high-speed fiber fronthaul link, as shown in Fig. 1. Denote the set of the UEs as $\mathcal{N} = \{1, 2, \cdots, N\}$ and the set of the RRHs as $\mathcal{J} = \{1, 2, \cdots, J\}$. Note that the analytical work can be extended to UEs with multiple antennas, where each multiple-antenna UE can be regarded as the combination of several virtual single-antenna UEs. Hence, all the derivations and algorithms developed in this paper can be generalized.

Similar to [17], it is assumed that each UE $i$ has the task $U_i$ to be accomplished as follows

$$U_i = (F_i, D_i, T_i), \forall i \in \mathcal{N}$$ (1)
where $F_i$ (in cycles) describes the total number of the CPU cycles to be completed, $D_i$ (in bits) denotes the whole size of data required to be transmitted to MC-RAN if choosing to offload and $T_i$ (in seconds) is the delay constraint that this task has to be accomplished in order to satisfy the UE’s QoS requirement.

In MC-RAN, each UE belongs to one of the following sets, according to its own status and current available computing and communication resources:

- Offloading set $O$ is defined for UEs which have interests in offloading. We can further divide $O$ into $O^H$ and $O^L$, where $O^H$ is defined for UEs with high priority in offloading, whereas $O^L$ is defined for the rest of the offloading UEs. Then, one has $O=O^H \cup O^L$.
- Local execution set $L$ is defined for UEs which decide to execute tasks locally.
- Rescheduled set $R$ is defined for UEs which neither complete the tasks locally due to lack of computing resource, nor offload due to lack of computing or communication resource.

Thus, one has $\mathcal{N} = L \cup O \cup R$.

### B. Local Execution

For UEs which decide to conduct the task locally, i.e., $\forall i \in \mathcal{L}$, the execution time is

$$T_i^L = \frac{F_i}{f_i^L} \quad (2)$$

where $f_i^L$ is the computation capability (i.e, CPU cycles per second) for the $i$-th UE. Then, the computational power can be given as $[19] [20] [21]$

$$p_i^L = \kappa_i^L (f_i^L)^{\nu_i^L} \quad (3)$$

where $\kappa_i^L > 0$ and $\nu_i^L \geq 2$ are the positive constants. According to the realistic measurements, $\kappa_i^L$ can be set to $\kappa_i^L = 10^{-18}$ and $\nu_i^L$ can be set $\nu_i^L = 3$. By using (1) and (2), one has

$$C1 : T_i^L \leq T_i \quad (4)$$

Different UEs may have different computation capabilities and the constraints of $f_i^L$ is given by

$$C2 : f_i^L \leq f_{i,max}^L \quad (5)$$

where $f_{i,max}$ is the maximum computation capacity that the $i$-th UE can achieve and is finite. Defining the maximum power consumption for its UE as $P_{i,max}$, one can have $f_{i,max}^L = \sqrt[\nu_i^L]{\frac{P_{i,max}}{\kappa_i^L}}$. 


C. Task Offloading

For UEs who decide to offload the task, i.e., $\forall i \in \mathcal{O}$, the transmitted signal is written as

$$x_i = \sqrt{p_i^{tr}b_i}$$

(6)

where $p_i^{tr}$ denotes the transmission power of the UE $i$ and $b_i$ denotes the transmitting data symbol with unity average power $\mathbb{E}(|b_i|^2) = 1$. Then, the received signal at the RRHs is given by

$$y = \sum_{i \in \mathcal{O}} h_i \sqrt{p_i^{tr}b_i} + z,$$

(7)

where $h_i$ denotes the channel state information (CSI) from $i$-th UE to all the RRHs, $z$ denotes the additive white Gaussian noise (AWGN) vector and is assumed to be distributed as $\mathcal{C}\mathcal{N}(0, \sigma^2 I)$. Then, the signal-to-interference-plus-noise ratio (SINR) can be expressed by

$$\text{SINR}_{i}^{UP} = \frac{p_i^{tr}||m_i^T h_i||^2}{\sum_{k \in \mathcal{O}, k \neq i} p_k^{tr}||m_k^T h_k||^2 + \sigma^2 ||m_i||^2}$$

(8)

where $m_i$ denotes the receive beamforming vector in RRH for the $i$-th UE. Defining the maximum transmission power as $P_{i,max}$, we obtain

$$C3: p_i^{tr} \leq P_{i,max}$$

(9)

Thus, the achievable rate for UE $i$ is given by

$$r_{i}^{UP} = B \cdot \log(1 + \text{SINR}_{i}^{UP})$$

(10)

where $B$ is the wireless channel bandwidth.

If the $i$-th UE decides to offload the task to MC-RAN, the task data $D_i$ has to be transmitted to MC-RAN. From (10), the transmission time is given by

$$T_i^{TR} = \frac{D_i}{r_i^{UP}}$$

(11)

D. Mobile Cloud

Mobile cloud hosts mobile clones, BBU pool and middlehaul. The middlehaul connects mobile clone to BBU pool. They are introduced below.
1) Mobile Clone: If the task being offloaded to mobile clone, the execution time in $i$-th mobile clone can be expressed as

$$T^C_i = \frac{F_i}{f^C_i} \quad (12)$$

where $f^C_i$ is the computational capability of the $i$-th mobile clone. Then, the total time including data offloading and execution is given by

$$T^O_i = T^{Tr}_i + T^C_i. \quad (13)$$

As in [19], the time for sending data back to UE in the downlink is ignored. Then, the following QoS constraints must hold

$$C4 : T^O_i \leq T_i. \quad (14)$$

Assuming that different mobile clones may have different computational capabilities and the constraint of the computation capacity of the $i$-th mobile clone is given by

$$C5 : f^C_i \leq f^C_{i,max} \quad (15)$$

where $f^C_{i,max}$ is the maximum computation capacity that is allocated to the $i$-th mobile clone.

2) BBU pool: In [22], the architecture of general processing processor (GPP) based BBU pool was presented and it showed that there is relationship between computational resource of BBU and the number of served UEs. GPP based BBU can be scalable and adjusted in order to meet the requirement of the network. [23], [24] has given the relationship between CPU frequency and the processing time of the signal transmission. It has shown that in order to meet QoS of the transmission, the minimal CPU computational capability constraints have to be satisfied. Similarly, [14] has proposed that the BBU can be run in virtual machines in the cloud and its computational capacity can be dynamically allocated.

Defining $H$ as the computational resource required in BBU pool to serve one UE, the whole computational capacity of BBU pool is given by

$$f^B = \sum_{i \in O} \left| \| m_i \|_2 \right|_1 H \quad (16)$$

where

$$\left| \| m_i \|_2 \right|_1 = \begin{cases} 
0, & \text{if } \| m_i \|_2 = 0 \\
1, & \text{otherwise}
\end{cases} \quad (17)$$
\[ \sum_{i \in \mathcal{O}} \| m_i \|_2^2 \] stands for the number of UEs that the BBU can process. Without loss of generality, we assume \( H = 1 \) and one has

\begin{equation}
C6 : \sum_{i \in \mathcal{O}} \| m_i \|_2^2 \leq f_{B_{\text{max}}}^B
\end{equation}

where \( f_{B_{\text{max}}}^B \) is the maximum computational capacity allocated to BBU pool from the cloud.

Note that, in the MC-RAN, service computing capacity \( f_{C_{i,\text{max}}}^C \) given in (15) and communication computing capacity \( f_{B_{\text{max}}}^B \) given in (18) can be allocated and adjusted according to the requirements. If more network resource is needed, \( f_{B_{\text{max}}}^B \) can be allocated more than \( f_{C_{i,\text{max}}}^C \). On the other hand, if there are more resource hungry task requests, \( f_{C_{i,\text{max}}}^C \) can be allocated more than \( f_{B_{\text{max}}}^B \). \( f_{C_{i,\text{max}}}^C \) and \( f_{B_{\text{max}}}^B \) are important system parameters in the joint resource allocation algorithm.

3) Middlehaul: Middlehaul is defined as the link connecting BBU and mobile clone. The whole data rate for all the offloading UEs transmitting to their mobile clones is given by

\[ s^M = \sum_{i \in \mathcal{O}} \| m_i \|_2^2 T_{i,UP}^i \] (19)

The capacity of middlehaul is constrained as

\begin{equation}
C7 : \sum_{i \in \mathcal{O}} \| m_i \|_2^2 T_{i,UP}^i \leq S_{\text{max}}^M
\end{equation}

where \( S_{\text{max}}^M \) is the maximum capacity of the middlehaul. Note that in \( C7 \), it is assumed that the transmission data in middlehaul is the decoded data, similar to [25]. Also, the transmission data can be quantized to IQ samples, as analyzed in [26], [27], which is not considered in this paper.

Note also that the maximum BBU capacity in \( C6 \) may determine the maximum number of UEs allowed to offload at the same time slot, whereas the maximum middlehaul capacity in \( C7 \) may determine which UEs can be allowed to offload, based on their required offloading speed.

III. ENERGY CONSUMPTION AND PROBLEM FORMULATION

A. Energy Consumption at UE

If the \( i \)-th UE decides to execute computation locally, the energy consumption is given by

\[ E_i^L = p_i^L \cdot T_i^L. \] (21)

If the \( i \)-th UE decides to offload computation, the energy consumption is given by

\[ E_i^{Tr} = p_i^{Tr} \cdot T_i^{Tr}. \] (22)

For the UE that is able to perform the computation locally, the necessary condition for this UE to choose to offloading is that

\begin{equation}
C8 : E_i^{Tr} \leq E_i^L
\end{equation}
B. Energy Consumption at MC-RAN

The power consumption of MC-RAN includes three parts: power consumption at the RRHs, fronthaul links and mobile cloud.

1) RRH: In the uplink, there are two states for each RRH, either in active model or sleep model. If RRH is in active model, signals from all the UEs can be received, and more power consumption is required for maintaining the active mode of the RRH. However, if low data flow is predicted/monitored in RRH, this RRH may be put into sleep mode for the sake of power saving. Therefore, we model the power consumption of RRH $i$ as

$$p_R^j = \sum_{i \in O} \| \mathbf{m}_{ij} \|^2 \left( p_{j}^{RRH,active} + p_{j}^{RRH,sleep} \right), \forall j \in \mathcal{J},$$

where $p_{j}^{RRH,sleep}$ represents the circuit power consumption when the $j$-th RRH is in sleep mode, $p_{j}^{RRH,active}$ denotes more power (i.e., $p_{j}^{RRH,active}$) to be consumed if the $j$-th RRH is in active mode, and $|x|_1$ is given by (17), which can be either 1 for active status of RRH or 0 for sleep mode. The status of each RRH can be decided via the value of $|x|_1$.

2) Fronthaul: The fronthaul is the link that carries the data streams from RRH to BBU pool. Similarly, if the $j$-th RRH is active, the corresponding $j$-th fronthaul has also to be active in order to transmit the signal from this RRH to mobile cloud. Thus, the power consumption of the $j$-th fronthaul can be given by

$$p_F^j = \sum_{i \in O} \| \mathbf{m}_{ij} \|^2 \left( p_{j}^{FH,active} + p_{j}^{FH,sleep} \right), \forall j \in \mathcal{J},$$

where $p_{j}^{FH,sleep}$ represents the circuit power consumption when $j$-th fronthaul is in sleep mode, whereas $p_{j}^{FH,active}$ denotes more power (i.e., $p_{j}^{FH,active}$) to be consumed if fronthaul $j$ is in active mode.

3) Mobile Cloud: Mobile cloud has always to be active in order to support UE’s communication and computation. Therefore, we assume that the power consumption of mobile cloud is a constant $p_M$ for simplicity.

4) Whole Energy Consumption: The whole energy consumption in MC-RAN is given by

$$p_{MC-RAN} = \sum_{j \in \mathcal{J}} (p_R^j + p_F^j) + p_M = \sum_{j \in \mathcal{J}} \left( \sum_{i \in O} \| \mathbf{m}_{ij} \|^2 \right) \left( p_j^\Phi + p_{j}^{sleep} \right) + p_M$$

where $p_j^\Phi = p_j^{RRH,active} + p_j^{FH,active}$, $p_{j}^{sleep} = \sum_{j \in \mathcal{J}} \left( p_j^{RRH,sleep} + p_j^{FH,sleep} \right)$. Thus, the energy consumption can be expressed as

$$E_{MC-RAN} = p_{MC-RAN} T$$
where \( T \) is the time MC-RAN operates. In practice, we may set \( T = \max\{T_i^\text{Tr}, i \in \mathcal{N}\} \), as it runs until all the UEs complete offloading.

\[ \text{C. Problem Formulation and Analysis} \]

Before formulating the problem, one may consider the following practical system limitations:

1. Due to \( C1 - C2 \), not all the UEs have the capacity to complete the tasks locally;
2. Due to \( C3 - C5 \), not all the UEs are able to offload their computations, as the required transmission power may be larger than the maximal power capability of the UEs;
3. Due to the limitation of the available resources in MC-RAN, i.e., \( C6 - C7 \), not all the offloading requests from the UEs can be accepted by MC-RAN;
4. Due to \( C8 \), if the transmission energy consumption required is larger than the local execution energy consumption, UE may not have the interest in offloading. However, it is difficult for UE itself to know how much energy it needs to offload, as it is affected by the decisions of the other UEs and the interference caused by them.

Therefore, one may propose a two-stage optimization problem as follows, i.e., \( \mathcal{P}1 \) for access control and \( \mathcal{P}2 \) for energy minimization resource allocation as follows

\[
\mathcal{P}1 : \quad \max_{\mathcal{R}, \mathcal{O}, \mathcal{L}} |\mathcal{O}| \\
\text{s.t.} \quad C1 - C8
\]

\[
\mathcal{P}2 : \quad \min_{\mathbf{m}, \mathbf{p}^\text{Tr}, \mathcal{J}^*} E^{\text{MC-RAN}} + \lambda \cdot E^{\text{UE}} \\
\text{s.t.} \quad C1 - C8, \forall i \in \mathcal{O}^*.
\]

In \( \mathcal{P}1 \), we aim to find the largest offloading subset, i.e., \( \mathcal{O}^* \) that can be supported by MC-RAN. Then, based on solution \( \mathcal{O}^* \) from \( \mathcal{P}1 \), one can solve \( \mathcal{P}2 \), which optimizes the transmission power from each UE, the corresponding receive beamforming vectors and the active RRH and fronthaul pairs in MC-RAN, in order to minimize the whole energy consumption.

In \( \mathcal{P}2 \), \( \lambda \geq 0 \) is a weighting factor to control the energy consumption priority between MC-RAN and UEs, \( \mathbf{m} \) is a collection of all the receive beamforming vectors in MC-RAN, \( \mathbf{p}^\text{Tr} \) is a collection of the transmission powers for all the UEs. Note that different from other papers, such as [4], which only considered minimizing the energy consumption for all the UEs, we consider minimizing the whole energy overhead including the MC-RAN and UEs, as energy overhead or the electricity cost are among the most important factors in the overall operational expenditure.
and is of huge importance and interest in the operators’ perspectives [28]. Note also that one RRH corresponds to one fronthaul and \( J^* \) can be seen as the set of all the active RRH and fronthaul pairs. By optimizing the number of the active RRH and fronthaul pairs, significant energy saving can be made in MC-RAN.

Remark 1: From the above two NP-hard optimizations, one can see that central decision by MC-RAN is essential to get the optimal solutions, as each UE itself does not have the global information such as other UE’ decision or the current available resource in MC-RAN. On the other hand, to reduce the complexity of the central decision as well as decrease the overhead and traffic between UE and MC-RAN in wireless channel and fronthaul, distributed local decision in each UE without global information is necessary as well.

To make local decision, each UE itself has to estimate its local execution and transmission energy first. Only those UEs that satisfy at least one of the following conditions

- have enough transmission power to transmit data;
- cannot complete their tasks locally;
- see the potential to reduce their energy consumption

have the possibility to send the offloading requests to MC-RAN and participate in resource competition.

Moreover, one can see that in \( P_1 \), before maximizing the size of offloading set \( |O| \), one should first consider to minimize the cardinality of set \( \mathcal{R} \), in order for more UEs to satisfy the latency requirements (i.e., \( C1 \) or \( C4 \)), or increase the offloading gain.

Based on the above Remark 1, three processes are conducted to deal with \( P_1 \) and \( P_2 \) as follows

- Process 1: Local decision (i.e., Algorithm 1: DLDA, introduced in Section IV) is first made in each UE. This process aims to roughly decide the set \( O^L, O^H, \mathcal{R} \) and \( \mathcal{L} \). Only those UEs that meet the offloading criterion can send the requests to the MC-RAN for resource competition. In this step, each UE can roughly decide whether to offload or not based only on its local information, such as its own channel state information, its own processing capacity, etc. Note that as \( O^L \) and \( O^H \) in this process is not the final offloading set, they still need to be checked by MC-RAN with the global information next.
- Process 2: After receiving the offloading requests from the UEs, central access control decision (i.e., Algorithm 2: CACA, introduced in Section V) is conducted in MC-RAN, to
decide which UEs can be allowed to offload, based on the current available communication and computation resource. Then $O^L$, $O^H$, $R$ and $L$ can be finally decided in this process.

- **Process 3**: Based on the offloading set (i.e., $O^L$ and $O^H$) determined by **Process 2**, central resource allocation is decided by MC-RAN (i.e., Algorithm 3: CRAA, introduced in Section VI), to allocate the corresponding communication and computation resource to each offloading UE.

IV. **Decentralized Local Decision Algorithm (DLDA)**

In this section, DLDA is proposed so that each UE can make the local decision first. After DLDA, $R$, $O^H$, $O^L$ and $L$ can be roughly determined, without the help of the global information from other UEs and MC-RAN. DLDA can be seen as the pre-screening of the offloading candidates before they are finally determined by CACA in MC-RAN (which will be introduced in the next section).

A. **Deciding $O^H$ Roughly**

In this subsection, we aim to decide $O^H$, in which the UEs cannot complete the tasks locally and then are assigned with high priority in offloading. For each UE, we formulate the local execution energy minimization as

$$ P1.1 : \min_{f^L_i} E^L_i, \forall i \in N $$

s.t. $C1$, $C2$.  

For above problem, as the delay constraint for the task is $T_i$, one can easily obtain the optimal clock frequency $f^L_i$ as $f^L_i = \frac{F_i}{T_i}$, the optimal power consumption as $p^L_i = \frac{\kappa}{f^L_i} F^\nu_i L_i$ and the optimal energy consumption as

$$ E^L_i = \frac{\kappa}{f^L_i} F^\nu_i L_i \frac{L_i}{T_i} $$

However, the above solution is only feasible if $f^L_i \leq f^L_{i,max}$ and there is no solution if

$$ f^L_i > f^L_{i,max}, $$

which means the minimum clock frequency required for executing the task locally is larger than the maximum clock frequency available at this UE. For those UEs, they cannot complete the task locally and the only way to complete the task is to offload it to the cloud. Thus, we can add those UEs into $O^H$. Then, the rest of the UEs are added into set $N \setminus O^H$ for further processing.
B. Deciding $O^L$, $R$ and $L$ Roughly

In this subsection, we aim to decide $O^L$, $R$ and $L$ roughly, without the assistance of the global information. To this end, we have to determine the minimal required transmission power/energy for each UE if offloaded. This operation can find out whether the UE has the ability to offload the task to MC-RAN.

**Proposition 1**: For each UE choosing to offload, minimization of its energy consumption is equivalent to minimizing its power consumption.

**Proof**: The transmission energy minimization of each UE is formulated as $\mathcal{P}1.2$, if we do not consider interference from other UEs.

\[
\mathcal{P}1.2 : \min_{p_{i,Tr}^T} E_{i,Tr}^*, \forall i \in \mathcal{N}
\]

s.t. $C3 - C5$.

For $\mathcal{P}1.2$, one can see the objective function can be written as

\[
E_{i,Tr} = \frac{p_{i,Tr}^T D_i}{B \cdot \log(1 + \frac{p_{i,Tr}^T \|m_i^T h_i\|^2}{\sum_{k \in O, k \neq i} p_{k,Tr}^T \|m_i^T h_k\|^2 + \sigma^2 \|m_i\|^2})}.
\]

By taking the derivative of $E_{i,Tr}$ with respect of $p_{i,Tr}^T$, one can get

\[
\frac{\partial E_{i,Tr}}{\partial p_{i,Tr}^T} = \frac{D_i \left((\|m_i^T h_i\|^2 p_{i,Tr}^T + \iota) \log \left(\frac{\|m_i^T h_i\|^2 p_{i,Tr}^T + 1}{\|m_i^T h_i\|^2 + \sigma^2 \|m_i\|^2}\right)\right)}{B \left(\|m_i^T h_i\|^2 p_{i,Tr}^T + \iota\right) \log^2 \left(\frac{\|m_i^T h_i\|^2 p_{i,Tr}^T + 1}{\|m_i^T h_i\|^2 + \sigma^2 \|m_i\|^2}\right)},
\]

where $\iota = \sum_{k \in O, k \neq i} p_{k,Tr}^T \|m_i^T h_k\|^2 + \sigma^2 \|m_i\|^2$. It is easy to find that

\[
\frac{\partial E_{i,Tr}}{\partial p_{i,Tr}^T} > 0.
\]

With the decrease of $p_{i,Tr}^T$, $E_{i,Tr}$ decreases. Therefore, the minimal transmission energy can be obtained if the minimal transmission power is applied.

**Proposition 2**: If UE decides to offload, the minimal transmission power is

\[
p_{i,Tr,\min}^T = \left(2 - \frac{p_{i,\min}}{p_{i,\max}} - 1\right) \sigma^2 \frac{\|h_i\|^2}{\|h_i\|^2}
\]

and the minimal transmission energy is

\[
E_{i,\min}^* = p_{i,\min}^T \left(T_i - \frac{F_i}{f_{C,mac}}\right)
\]

**Proof**: The minimal transmission power is determined by the minimum achievable rate. By using $C4$ and $C5$, one can get the minimum achievable rate as

\[
C9 : r_{i,up} \geq R_{i,\min}
\]
where

\[ R_{i,\text{min}} = \frac{D_i}{T_i - F_i} \]  \hspace{1cm} (40)

Then, from (8) and (10), one can get the transmission power as

\[ p_{T,\text{min}} = \left( 2 \frac{R_{i,\text{min}}}{\| \mathbf{m}_i^T \mathbf{h}_i \|^2} - 1 \right) \lambda \]  \hspace{1cm} (41)

The minimal transmission power can be obtained by assuming there is only one UE conducting offloading, i.e., no interference from other UEs. By applying minimum mean square error (MMSE) receiver, one can get the minimal transmitting power and energy as (37) and (38), respectively.

Note that from (37) and (38), the minimum transmit power and the corresponding minimum energy can be calculated based only on local information, since the required information is available at each UE. Then, based on above analysis, we propose the local decision DLDA conducted in each UE to initially decide \( O^L, O^H, R \) and \( L \), without any global information.

**Algorithm 1** Decentralized Local Decision Algorithm (DLDA).

1. Each UE \( \forall i \in \mathcal{N} \) obtains \( f_{i}^{L^*} \) and \( E_{i}^{L^*} \) by solving problem \( \mathcal{P}1.1 \);
2. If \( f_{i}^{L^*} > f_{i,\text{max}}^{L} \)
3. Add this UE into \( O^H \);
4. Else
5. Add this UE into \( \mathcal{N} \setminus O^H \);
6. End If
7. Each UE \( \forall i \in O^H \) obtains \( p_{T,\text{min}}^{i} \) from (32);
8. If \( p_{T,\text{min}}^{i} \geq P_{i,\text{max}} \)
9. Move this UE from \( O^H \) to \( R \) and update \( O^H \);
10. End If
11. Each UE \( \forall i \in \mathcal{N} \setminus O^H \) obtains \( p_{T,\text{min}}^{i} \) from (37);
12. If \( p_{T,\text{min}}^{i} \geq \max (P_{i,\text{max}}, P_{i,\text{max}}^{\Delta}) \)
13. Add this UE into \( L \);
14. Else
15. Add this UE into \( O^L \);
16. End If
17. Output \( O^L, O^H, L, R \) to Stage 2.

DLDA is summarized in Algorithm 1, where each UE first checks if it can complete task locally, by solving \( \mathcal{P}1.1 \). The UEs that cannot complete the tasks will be assigned high priority...
(adding them to $O^H$). Otherwise, they can be added to $\mathcal{N} \setminus O^H$. For the UEs in the set $O^H$, if the UEs are not able to offload the tasks either, i.e., $P^T_{i,\min} \geq P_{i,\max}$, then they will be moved from $O^H$ to $\mathcal{R}$. For the UEs in the set $\mathcal{N} \setminus O^H$, if their minimal transmission power obtained from (37) is larger than the maximal power, i.e., $P^T_{i,\min} \geq P_{i,\max}$, the UEs will be moved to $\mathcal{L}$. Also, if their minimal transmission energy is larger than local execution energy, i.e., $E^T_{i,\min} \geq E^L_i$, then the UEs will be added into set $\mathcal{L}$ as well. For the rest of UEs in set $\mathcal{N} \setminus O^H$, they can be added into offloading set $O^L$ as the offloading candidates, which will be updated in central decision next based on available resource in MC-RAN.

For the UE finally accepted by MC-RAN to offload task, its transmission power needs to meet the following constraint (42), otherwise this UE may not have interest in offloading.

$$p^T_i \leq \min \left( P_{i,\max}, P^\Delta_{i,\max} \right),$$

where $P^\Delta_{i,\max} = \frac{E^L_i - E^F_i}{1 - T_i}$.

V. CENTRAL ACCESS CONTROL ALGORITHM (CACAL)

Due to the limited available resources in MC-RAN and unknown interference among the UEs, the offloading set $O = O^L \cup O^H$ obtained in the last section may not be all accepted by MC-RAN. Thus, central decision is introduced to conduct in MC-RAN to make the final decision on which UEs can be accepted.

In order to maximize the number of offloading UEs and meanwhile minimize the size of $\mathcal{R}$, $P1$ can be further transformed into the following two problems

$$\mathcal{P}1.3 : \max_{O^H, \mathcal{R}} |O^H|$$

s.t. $C3 - C7$, $\forall i \in O^H$ from DLDA

and

$$\mathcal{P}1.4 : \max_{O^L, \mathcal{L}} |O^L|$$

s.t. $C3 - C7$, $\forall i \in O^L$ from DLDA,

Resource Guarantee for UEs in $O^H$ from $\mathcal{P}1.3$

In $\mathcal{P}1.3$, MC-RAN first accepts UEs with high offloading priority, i.e., by considering set $O^H$ from DLDA. UEs which cannot be accepted by MC-RAN will be moved to set $\mathcal{R}$. Then, in $\mathcal{P}1.4$, based on the remaining resource and also by guaranteeing resource for UEs in set $O^H$, MC-RAN accepts requests from the rest of UEs. UEs which cannot be accepted in $\mathcal{P}1.4$ will be
moved to $\mathcal{L}$. Next, we will propose central accesses control decision, i.e., CACA, which includes Part I (Section V. A) and Part II (Section V. B) to deal with $\mathcal{P}1.3$ and $\mathcal{P}1.4$, respectively.

A. Deal with $\mathcal{P}1.3$

In $\mathcal{P}1.3$, $C4$ and $C5$ can be transformed to $C9$. Also, if the individual power constraint (i.e., $C3$) is not considered, $\mathcal{P}1.3$ can be approximated as

$$\max_{m,p^{Tr},O^H,R} \quad |O^H|$$

s.t. $C6, C7, C9, \forall i \in O^H$

Note that $m$ and $p^{Tr}$ obtained here are not the optimal values in terms of the whole energy minimization. They still have to be updated in the next section. However, they can help to decide the maximal number of the offloading UEs here. In the above problem, the variables $p^{Tr}$ are affected by $C9$ in the uplink, whereas $C6$ and $C7$ are constraints related to MC-RAN. Therefore (45) is very hard to solve. Next, we establish the link between UE and MC-RAN, by applying uplink and downlink duality. Inspired by [29], we provide the uplink sum-power minimization as (46) and virtual downlink sum-power minimization as (47) below

$$\min_{m,p^{Tr}} \sum_{i \in O^H} p^{Tr}_i$$

s.t. $C6, C7, C9$

$$\min_{v} \sum_{i \in O^H} v^H_i v_i$$

s.t. $C10 : \sum_{i \in O^H} \|v_i\|^2_1 \leq f^B_{\max}$, $C11 : \sum_{i \in O^H} \|v_i\|^2_1 R_i,_{\min} \leq S_{\max}$

$$C12 : r^{VD}_i \geq R_i,_{\min}; \forall i \in O^H$$

where $v_i$ is the virtual downlink transmission beamforming vector from all the RRHs to $i$-th UE, $v$ is a collection of all the $v_i$ for $\forall i \in O^H$, $r^{VD}_i$ is the virtual downlink transmission data rate defined as

$$r^{VD}_i = B \cdot \log(1 + \text{SINR}^{VD}_i)$$

and

$$\text{SINR}^{VD}_i = \frac{||h^H_i v_i||^2}{\sum_{k \in O, k \neq i} ||h^H_i v_k||^2 + \sigma^2}$$

DRAFT
where $m^*$, $p^T r^*$ and $v^*$ as the optimal solutions to problems (46) and (47), respectively. Then similar to [30], $v^*$ and $m^*$ can be set to be identical and moreover, one can have $\sum_{i \in \mathcal{O}^H} p_i^T r^* = \sum_{i \in \mathcal{O}^H} v_i^H v_i$ in above problems. Also, similar to [29], for any given feasible solution to problem (46), one can always find a corresponding feasible solution to problem (47), and vice versa. Therefore, problems (46) and (47) can take the same optimal value with the same set of beamforming vectors, i.e., $v^*$ and $m^*$ can be set to be identical.

In problem (47), $C_{12}$ can be transformed to the second-order cone (SOC) constraint in the virtual downlink as [31]

$$
C_{13} : \sqrt{1 - \frac{1}{2 R_{i, min}}} \sqrt{\sum_{k \in \mathcal{O}^H} ||h_i^H v_k||^2 + \sigma^2} \leq \Re (||h_i^H v_i||^2)
$$

(50)

Next, by using nonnegative auxiliary variables, similar to [32] and above uplink-downlink duality, (45) can be transformed to

$$
\min_{v, y, \Omega^H, R} \sum_{i \in \mathcal{O}^H} v_i^H v_i + Q \sum_{i \in \mathcal{O}^H} y_i,
$$

$$\text{s.t.: } C_{10}, C_{11}, C_{14} : \sqrt{1 - \frac{1}{2 R_{i, min}}} \sqrt{\sum_{k \in \mathcal{O}^H} ||h_i^H v_k||^2 + \sigma^2} \leq \Re (||h_i^H v_i||^2) + y_i, \forall i \in \mathcal{O}^H
$$

(51)

where $Q$ is a large positive constant, $\{y_i, i \in \mathcal{O}^H\}$ are the nonnegative auxiliary variables and $y$ is a collection of $\{y_i, i \in \mathcal{O}^H\}$. One can see that there always exist large enough variables $\{y_i, i \in \mathcal{O}^H\}$ to satisfy all the constraints in above problem. By solving (51), we can obtain the value of $\{y_i, i \in \mathcal{O}^H\}$. The number of zero entries in $\{y_i, i \in \mathcal{O}^H\}$ in (51) corresponds to the number of accepted UEs, i.e., $|\mathcal{O}^H|$ in (45). Similarly, one can also obtain the set of the accepted UEs by checking $\{i | r_i^{VD} \geq R_{i, min}, i \in \mathcal{O}^H\}$. $\sum_{i \in \mathcal{O}^H} v_i^H v_i$ in objective function of (51) is to minimize the power consumption of all the offloading UEs.

Moreover, $C_{10}$ and $C_{11}$ in (51) include the non-smooth indicator function, which make (51) intractable. They can be approximated by applying the following fractional function, i.e.,

$$
f_\theta(x) = \frac{x}{x + \theta}
$$

(52)

where $\theta$ is a very small positive value. Then $C_{10}$ and $C_{11}$ can be approximated as $C_{15}$ and $C_{16}$, respectively.

$$
C_{15} : \sum_{i \in \mathcal{O}} f_\theta (||v_i||^2) \leq f_{max}^B
$$

(53)

$$
C_{16} : \sum_{i \in \mathcal{O}} f_\theta (||v_i||^2) R_{i, min} \leq s_{max}^M
$$

(54)
Then, by using (C15) and (C16), problem (51) can be transformed into the following problem

$$
\min_{v, y, \text{Update } O^H, R} \sum_{i \in O^H} v_i^H v_i + Q \sum_{i \in O^H} y_i
$$

s.t. : $C14, C15, C16, \forall i \in O^H$  

Problem (55) is more tractable than (45), as both the objective function and constraints in (55) are continuous and differentiable. Although Problem (55) is still nonconvex due to the concavity of $f_\theta(\cdot)$ in (C15) and (C16), it is a well-known difference of convex (d.c.) program, which can be solved effectively by the SCA method [33]. This approach was proposed to approximate the concave function as Taylor expansion with first order. Therefore, by using the concavity of $f_\theta(x)$, one can have

$$
\begin{align*}
  f_\theta(||v_i||^2) &\leq f_\theta(||v_i(t)||^2) + \alpha_i(t)(||v_i||^2 - ||v_i(t)||^2) \\
\end{align*}
$$

where $v_i(t)$ is the solution of $i$-th UE in the $t$-th iteration, $\alpha_i(t) = f_\theta'(||v_i(t)||^2)$ and $f_\theta'(x)$ is the first-order derivative of $x$. By replacing $f_\theta(\cdot)$ in (55) with the right hand side of (56), we can solve the following optimization in the $(t+1)^{th}$ iteration as

$$
\begin{align*}
  \min_{v, y, \text{Update } O^H, R} \sum_{i \in O^H} v_i^H v_i + Q \sum_{i \in O^H} y_i \\
  \text{s.t. : } C14, C17 : \alpha_i(t)||v_i||^2 \leq F_{max}^B - (f_\theta(||v_i(t)||^2) - \alpha_i(t)||v_i(t)||^2) \\
  C18 : R_{i, \min} \alpha_i(t)||v_i||^2 \leq S_{max}^M - (f_\theta(||v_i(t)||^2) - \alpha_i(t)||v_i(t)||^2), \forall i \in O^H
\end{align*}
$$

Then, (57) is a convex problem, which can be solved by interior point method efficiently. The UE with the largest gap to its target SINR, i.e., $y_i$ in (C14) are most likely to be forced to further reduce its virtual downlink transmission power to zero and encouraged to drop out of $O^H$ eventually. However, UE with smallest gap to its target SINR, such as $y_i = 0$ in (C14) will keep its virtual downlink transmission power non-zero and thus one can have $|||v_i||^2|_1 = 1$ to indicate UE is accepted by MC-RAN.

Note that (45) did not consider the individual power consumption and its constraint. Thus, the next step is to obtain the individual power $p_i^{Tr}$. Similar to [29], by setting $m_i = v_i$ for all $i \in O^H$ and using fixed-point method in (46), $\{p_i^{Tr}, i \in O^H\}$ is obtained. Then, we are able to check if allocated $\{p_i^{Tr}, i \in O^H\}$ can be met by each UE’s maximum transmission power, i.e., C3.

We define a new set $B$ that includes UEs whose allocated transmission power is larger than the maximum transmission power. Then one can have $B = \{i | p_i^{Tr} \geq P_{i, \max}, i \in O^H\}$. Define a
set of the normalized power violation factor for each user in $B$ as \( \eta_i = \frac{p_{i,T} - P_{i,max}}{P_{i,max}}, i \in \mathcal{O}^H \). Then, our idea is to first remove the \( i^* \)-th UE with the biggest normalized power violation factor, i.e., \( i^* = \text{argmax}(\eta_i, i \in B) \) from set \( \mathcal{O}^H \) in problem (45) and then redo problem (45) again until $B = \emptyset$. Finally, we update \( \mathcal{O}^H \) by removing UEs which cannot archive its required transmission data rate $R_{i,min}$. The whole process of first part of CACA to deal with $P_{1.3}$ is summarized in Algorithm 2: Part I, where $v(t)$ and $\alpha(t)$ are the collection of $v_i(t)$ and $\alpha_i(t)$, respectively, in the $t$-th iteration.

**Algorithm 2: Part I Centralized Access Control Algorithm (CACA): Part I.**

1. Initialize $t = 1$, $v(0)$ and $\alpha(0)$;
2. Repeat
3. Solve (57) to get $v(t)$ with $v(t-1)$, $\alpha(t-1)$;
4. Update $\alpha(t)$ with $v(t)$;
5. Terminate If convergence or maximum number of iterations are reached;
6. Obtain $B = \{ i | p_{i,T} \geq P_{i,max}, i \in \mathcal{O}^H \}$, by solving (46);
7. If $B = \emptyset$, then, go to 11;
8. Else, go to 10;
9. End if
10. Order $\eta_i = \frac{p_{i,T} - P_{i,max}}{P_{i,max}}, i \in B$ and find the biggest $i^* = \text{argmax}(\eta_i, i \in B)$;
   Remove $i^*$-th UE from $\mathcal{O}^H$ and add it into $\mathcal{R}$, go to 2;
11. Update $\mathcal{O}^H = \{ i | r_{i,U} \geq R_{i,min}, i \in \mathcal{O}^H \}$, by applying equation (10);
12. Update $\mathcal{R}$ by adding $\{ i | r_{i,U} < R_{i,min}, i \in \mathcal{O}^H \}$ into $\mathcal{R}$.

**B. Deal with $P_{1.4}$**

MC-RAN first guarantees the resource provided to $\mathcal{O}^H$ obtained from the last subsection and then consider to accept other UEs in $\mathcal{O}^L$, if there are still remaining resources. Thus, after guaranteeing the resource provided to $\mathcal{O}^H$ from $P_{1.3}$, the remaining resource in MC-RAN BBU can be given by

$$C19 : \sum_{i \in \mathcal{O}^L} f_\theta (||v_i||^2) \leq f_{\max}^B - ||\mathcal{O}^H||$$

The remaining resource in MC-RAN middlehaul can be given by

$$C20 : \sum_{i \in \mathcal{O}^L} f_\theta (||v_i||^2) R_{i,min} \leq S_{\max}^M - \left( \sum_{i \in \mathcal{O}^H} R_{i,min} \right)$$

(59)
Then, by using the similar method to \(P_{1.3}\), \(P_{1.4}\) can be approximated as the following relaxed access control problem

\[
\begin{align*}
\min_{\mathbf{v}, \mathbf{z}, \text{Update } O^C, \mathcal{L}} & \quad \sum_{k \in \mathcal{H} \cup \mathcal{L}} \mathbf{v}_k^H \mathbf{v}_k + Q \sum_{j \in \mathcal{C}} z_j \\
\text{s.t.:} & \quad C'19, C'20, \\
C'21 : & \quad \sqrt{\frac{1}{2} R_{m,\min}} \sum_{k \in \mathcal{H} \cup \mathcal{L}} ||\mathbf{h}_k^H \mathbf{v}_k||^2 + \sigma^2 \leq \Re \left( ||\mathbf{h}_m^H \mathbf{v}_m||^2 \right), \forall m \in \mathcal{H} \\
C'22 : & \quad \sqrt{\frac{1}{2} R_{j,\min}} \sum_{k \in \mathcal{H} \cup \mathcal{L}} ||\mathbf{h}_k^H \mathbf{v}_k||^2 + \sigma^2 \leq \Re \left( ||\mathbf{h}_j^H \mathbf{v}_j||^2 \right) + z_j, j \in \mathcal{C},
\end{align*}
\]  

\[60\]

where \(\{z_i, i \in \mathcal{C}\}\) is a set of nonnegative auxiliary variables to ensure the feasibility of above problem, \(\mathbf{z}\) is a collection of \(\{z_i, i \in \mathcal{C}\}\), \(C'21\) is applied to guarantee the QoS from \(\mathcal{H}\) and \(C'22\) is the relaxed constraint to guarantee the QoS from \(\mathcal{C}\). \[60\] and auxiliary variable \(\alpha_i(t)\) is applied to help solving above problem as in last subsection.

Note that we first have to check if there are still remaining resource after we support the UEs from \(\mathcal{H}\). From \(C'19\) and \(C'20\), one can see if \(f_{\max}^B - |\mathcal{H}| > 0\) and \(S_{\max}^M - (\sum_{i \in \mathcal{H}} R_{i,\min}) > 0\), we can consider the offloading requests from \(\mathcal{C}\). Otherwise, we stop and conclude that no UEs from \(\mathcal{C}\) can be supported. Then, following the above transformation and solution in above subsection, \[60\] can be solved effectively. Similar to Part I of CACA, after we obtain the optimal solution from \[60\], we have to obtain \(p^T_r\) from \[46\] by using uplink-downlink duality. Then, for UEs in set \(\mathcal{H}\), we check whether the allocated transmission power violates UE’s maximum power constraint, i.e., \(C'3\). For UEs in set \(\mathcal{C}\), we have to not only check whether the allocated transmission power is larger than UE’s local execution power, but also check whether the allocated transmission power is larger than UE’s local execution power, i.e., \[42\]. Therefore, we define a new set \(\mathcal{D}\) as \(\mathcal{D} = \{i | p_i^T_r \geq P_{i,\max}, i \in \mathcal{H}\} \cup \{i | p_i^T_r \geq \max(\bar{P}_{i,\max}, \bar{P} \Delta_{i,\max}), i \in \mathcal{C}\}\). If \(\mathcal{D} \neq \emptyset\), we should move some UEs from \(\mathcal{C}\) to \(\mathcal{L}\). Next, define a set of the normalized power violation factor for each user in \(\mathcal{C}\) as \(\{\varphi_i = \frac{p_i^T_r - \max(\bar{P}_{i,\max}, \bar{P} \Delta_{i,\max})}{\max(\bar{P}_{i,\max}, \bar{P} \Delta_{i,\max})}, i \in \mathcal{C}\}\). Then, our idea is to first remove the \(i^*\)-th UE with the largest normalized power violation factor, i.e., \(i^* = \arg\max(\varphi_i, i \in \mathcal{C})\) from set \(\mathcal{C}\) and then redo problem \[60\] again until \(\mathcal{D} = \emptyset\). Finally, \(\mathcal{C}\) is updated by removing UEs which cannot achieve its required transmission data rate \(R_{i,\min}\).

We summarize the whole process of CACA: Part II in Algorithm 2: Part II, where \(v(t)\) and
\( \alpha(t) \) are the collection of \( \{v_m(t), v_j(t), \forall m \in \mathcal{O}^H \text{ from } \mathcal{P}1.3 \text{ and } \forall j \in \mathcal{O}^L \text{ from } \mathcal{P}1.2 \} \) and \( \{\alpha_m(t), \alpha_j(t), \forall m \in \mathcal{O}^H \text{ from } \mathcal{P}1.3 \text{ and } \forall j \in \mathcal{O}^L \text{ from } \mathcal{P}1.2 \} \), respectively, in the \( t \)-th iteration.

Algorithm 2: Part II  Centralized Access Control Algorithm (CACA): Part II.

1: \textbf{Check} if there are remaining resource by using \( f^B_{\text{max}} - |\mathcal{O}^H| \) and \( S^M_{\text{max}} - (\sum_{i \in \mathcal{O}^H} R_{i,\text{min}}) \) if above two equations is larger than zero, go to 2, otherwise \textbf{Stop};
2: Initialize \( t = 1, v(0) \) and \( \alpha(0) \);
3: Repeat
4: Solve (60) to get \( v(t) \) with \( v(t-1), \alpha(t-1) \), by using \( \alpha_i(t) \) in (56) and similar method in last subsection;
5: Update \( \alpha(t) \) with \( v(t) \);
6: \textbf{Terminate} If convergence or maximum number of iterations are reached;
7: Obtain Set \( D = \{i | p_{Tr}^i \geq P_{i,\text{max}}, i \in \mathcal{O}^L \text{ or } p_{Tr}^i \geq \max \{P_{i,\text{max}}, P^\Delta_{i,\text{max}}\}, i \in \mathcal{O}^C \} \), by solving (66);
8: If \( D = \emptyset \); then, go to 12;
9: Else, go to 11;
10: End if
11: Order \( \varphi_i = \frac{p_{Tr}^i - \max\{P_{i,\text{max}}, P^\Delta_{i,\text{max}}\}}{\max\{P_{i,\text{max}}, P^\Delta_{i,\text{max}}\}} \) \( i \in \mathcal{O}^C \) and find the biggest \( i^* = \arg\max\{\varphi_i, i \in \mathcal{O}^C\} \)
Remove the \( i^* \)-th UE from \( \mathcal{O}^C \) and add it into \( \mathcal{L} \), go to 3;
12: Update \( \mathcal{O}^L = \{i | r_{UP}^i \geq R_{i,\text{min}}, i \in \mathcal{O}^L \} \), by applying equation (10);
13: Update \( \mathcal{L} \) by adding \( \{i | r_{UP}^i < R_{i,\text{min}}, i \in \mathcal{O}^C \} \) into \( \mathcal{L} \).

VI. CENTRAL RESOURCE ALLOCATION ALGORITHM (CRAA)

After deciding the offloading set, i.e., \( \mathcal{O}^* = \mathcal{O}^C \cup \mathcal{O}^H \) from CACA above, we aim to reduce the whole system’s energy, including MC-RAN and UEs. According to \textbf{Proposition 1}, \( \mathcal{P}2 \) can be transformed to power minimization as

\[
\min_{m, p^{Tr}, J^*} \quad p^{MC-RAN} + \lambda \sum_{i \in \mathcal{O}} p_{Tr}^i \\
\text{s.t. } C3 - C7, \ \forall i \in \mathcal{O}^*
\]

(61)

where \( m \) and \( p^{Tr} \) are the collection of \( m_i \) and \( p_{Tr}^i, i \in \mathcal{O}^* \), respectively. Note that from CACA, we have already checked that the available resource in MC-RAN can support UEs in sets of \( \mathcal{O}^* \). Thus \( C6 \) and \( C7 \) can be dropped from (61). Also, \( C4 \) and \( C5 \) can be transformed to \( C9 \). If we
do not consider the individual power constraint (i.e., $C3$ or (42)) and omit the constant terms, i.e., $p_{\text{sleep}}$ and $p_{M}$, (42) can be written as

$$\min_{m, p_{T r}, J^*} \sum_{j \in J} \left( \sum_{i \in O} ||m_{ij}||^2 \right) + \lambda \sum_{i \in O} p_{T r}^{p_j}$$

s.t. $C9, \forall i \in O^*$

Similarly, by using the uplink-downlink duality [29], (62) can be transformed to

$$\min_{m, p_{T r}, J^*} \sum_{j \in J} \left( \sum_{i \in O} ||v_{ij}||^2 \right) + \lambda \sum_{i \in O} v_i^H v_i$$

s.t. $C23 : \sqrt{1 - \frac{1}{2}} \sqrt{\sum_{k \in O} ||h_i^H v_k||^2 + \sigma^2} \leq \text{Re} (||h_i^H v_i||^2), i \in O^*$

By applying the fractional function in (52) in the objective function of (63), one can have

$$\min_{m, p_{T r}, J^*} \sum_{j \in J} \left( \gamma_j(t) \left( \sum_{i \in O} ||v_{ij}||^2 \right) p_j^\phi \right) + \lambda \sum_{i \in O} v_i^H v_i$$

s.t. $C23$

Similarly, by using the following Taylor expansion approximation

$$f\theta(\sum_{i \in O} ||v_{ij}(t)||^2) \leq f\theta(\sum_{i \in O} ||v_{ij}(t)||^2) + \gamma_j(t)\left( \sum_{i \in O} ||v_{ij}||^2 - \sum_{i \in O} ||v_{ij}(t)||^2 \right), \forall j \in J$$

where $\gamma_j(t) = f\theta(\sum_{i \in O} ||v_{ij}(t)||^2)$, one can solve the following optimization in $(t+1)^{th}$ iteration as

$$\min_{m, p_{T r}, J^*} \sum_{j \in J} \left( \gamma_j(t) \sum_{i \in O} ||v_{ij}||^2 p_j^\phi \right) + \lambda \sum_{i \in O} v_i^H v_i$$

s.t. $C23$

From (66), it can be seen that the lower power the RRH and fronthaul pair needs to transmit to all the UE in the virtual downlink, the more likely this RRH and fronthaul pair will be removed from the serving set $J$. After solving (66), we can obtain the smallest active set of RRH and fronthaul pairs $J^*$. However, as we did not consider the individual power constraints (i.e., $C3$ or (42)), some UEs may violate its transmission capacity. Thus, the next step is to switch on some of the RRH and fronthaul pairs to lower the UEs’ transmission power, if there are some violations. Then, we obtain $p^{T r}$ by solving (46). Define two sets as $\mathcal{E}^H = \{ i | p_{T r}^{i} \geq P_{i,\text{max}}, i \in O^H \}$ and $\mathcal{E}^L = \{ i | p_{T r}^{i} \geq \max \left( P_{i,\text{max}}, P_{i,\text{max}}^{\Delta} \right), i \in O^L \}$. If $\mathcal{E}^H \cup \mathcal{E}^L \neq \emptyset$, there are some UEs’ allocated transmission power exceeding their capacity. Thus, we have to switch on some RRH
and fronthaul pairs to reduce UEs’ transmission power in the uplink. Next, we show which RRH and fronthaul pairs should be selected to switch on to support UEs which violate their transmission capacity. Similar to [29], we define the price for RRHs which are not in the serving set \( j \notin J^* \) as

\[
\phi_j = \sum_{i \in \mathcal{O}_H} \frac{p^{Tr}_i - P_{i,max}}{P_{i,max}} \|v_{ij}\|^2 + \sum_{i \in \mathcal{O}_L} \frac{p^{Tr}_i - \max(P_{i,max}, P_{i,max}^\Delta)}{\max(P_{i,max}, P_{i,max}^\Delta)} \|v_{ij}\|^2, j \notin J^* \tag{67}
\]

Then, we can switch on the RRH and fronthaul pair with the largest \( \{\phi_j, j \notin J^*\} \), i.e., adding \( j \)-th RRH and fronthaul pair into set \( J^* \), \( J^* = J^* \cup \arg\max \{\phi_j, j \notin J^*\} \). Next, we still have to check \( p^{Tr} \) by solving (46) again to see if there are some UEs which violate their transmission power. If so, we calculate the price in (67) again and add RRH and fronthaul pairs accordingly until \( \mathcal{E}_H \cup \mathcal{E}_L = \emptyset \).

We summarize the above process in Algorithm 3 below to deal with \( P2 \), where \( \gamma(t) \) is the collection of \( \{\gamma_j(t), j \in J\} \) in the \( t \)-th iteration.

---

**Algorithm 3** Central Resource Allocation Algorithm (CRAA).

1: Initialize \( t=1, \gamma(0), v(0), J^* = J \);
2: Repeat
3: \hspace{1em} Solve (66) to get \( v(t) \) with \( v(t-1), \gamma(t-1) \);
4: \hspace{1em} Update \( \gamma(t) \) with \( v(t) \);
5: \hspace{1em} Terminate If convergence or maximum number of iterations are reached;
6: \hspace{2em} Obtain \( \mathcal{E}_H = \{i|p^{Tr}_i \geq P_{i,max}, i \in \mathcal{O}_H\} \) and \( \mathcal{E}_L = \{i|p^{Tr}_i \geq \max(P_{i,max}, P_{i,max}^\Delta), i \in \mathcal{O}_L\} \), by solving (46);
7: \hspace{1em} While \( \mathcal{E}_H \cup \mathcal{E}_L \neq \emptyset \)
8: \hspace{2em} Obtain \( \phi_j, j \notin J^* \) by using (67);
9: \hspace{2em} Update RRH set \( J^* = J^* \cup \arg\max \{\phi_j, j \notin J^*\} \);
10: \hspace{2em} Obtain sets \( \mathcal{E}_H = \{i|p^{Tr}_i \geq P_{i,max}, i \in \mathcal{O}_H\} \) and \( \mathcal{E}_L = \{i|p^{Tr}_i \geq \max(P_{i,max}, P_{i,max}^\Delta), i \in \mathcal{O}_L\} \), by solving (46);
11: \hspace{1em} End While
12: Output \( p^{Tr}, m \) and \( J^* \).

---

To conclude, by using above proposed DLDA, CACA and CRAA, we are able to obtain the accepted offloading UEs’ set as well as the corresponding resource allocation solution, by solving \( P1 \) and \( P2 \), respectively. The overall process is illustrated in Fig. 2. Note that the parallel and distributed methods for nonconvex optimization such as [34], [35] are not suitable to deal with the above proposed problems, as our proposed algorithms are sequentially operated through three
processes. Specifically, we first execute the local decision to roughly decide the sets of $\mathcal{O}_L$, $\mathcal{O}_H$, $\mathcal{R}$ and $\mathcal{L}$, based only on each UE’s local state information. Then, the centralized access control is executed at the MC-RAN based on the roughly results from local decision process. Finally, based on results from centralized admission control, the power allocation solution and receiving beamforming vectors that minimizes the energy consumption can be decided. In our proposed three-process algorithm, the former process does not depend on the latter one, and only the latter one depends on the solutions from former one. However, to successfully apply the alternative algorithm, one condition should be that the former process depends on the solutions from the latter one, and vice versa so that they can alternatively updated. Hence, the alternative algorithm cannot be applied in our considered problem.

![Diagram](image)

Fig. 2. The overall process to solve the access control $\mathcal{P}_1$ and resource allocation $\mathcal{P}_2$.

VII. SIMULATION RESULTS

In this section, simulation results are presented to show the effectiveness of the proposed algorithm. Matlab with CVX tool [36] has been applied. The simulation scenario is shown in Fig. 3 where there are $N = 20$ UEs, each with one antenna and $L = 20$ RRHs, each equipped with $K = 2$ antennas. All the RRHs and UEs are assumed to be randomly distributed in a square area of coordinates $[0, 2000] \times [0, 2000]$ meters. The path and penetration loss are assumed as $p(d) = 148.1 + 37.6\log_{10}(d)$, where $d$ (km) is the propagation distance. It is assumed that the small scale fading is independent circularly symmetric Gaussian process distributed as $CN(0, 1)$. The noise power spectral density is assumed to be $-75$ dBm/Hz. The system bandwidth $B$ is set to 10 MHz, the capacity of middlehaul is set to 8 Mbps, the maximum transmission power for
Fig. 3. The simulation environment with $N = 20$ UEs, each equipped with $K = 2$ antennas and with $L = 20$ RRHs, each equipped one antenna, assumed to be randomly distributed in a square area of coordinates $[0, 2000] \times [0, 2000]$ meters.

### TABLE I

| $D_i - D_{10}$ | 0.04 | 0.32 | 3.23 | 0.06 | 0.26 | 0.04 | 0.6 | 0.07 | 0.09 | 25 |
|----------------|------|------|------|------|------|------|-----|------|------|----|
| $D_{10} - D_{20}$ | 0.61 | 0.69 | 0.082 | 0.56 | 0.11 | 1.1 | 2.4 | 0.59 | 0.24 | 0.035 |
| $F_i - F_{10}$ | 9.85 | 9.6 | 0.95 | 1.9 | 0.912 | 83.7 | 0.89 | 31 | 4.5 | 6.5 |
| $F_{10} - F_{20}$ | 0.88 | 2.5 | 0.9 | 21 | 14 | 25 | 0.95 | 15 | 1.2 | 85 |

each UE is set to 1 W, the power $p^j_\Phi$ for all the RRHs is set to 1 W and $\lambda$ is set to 1. Moreover, the computing capacity in each UEs is $1 \times 10^6$ cycles/s, while the maximum computation capacity for each mobile clone is $1 \times 10^8$ cycles/s. Time slot is set to 1 s for all the UEs. $p^{\text{sleep}}$ and $p^M$ are omitted, as they are constants. Unless noted otherwise, each UE has the computing task to be completed, with the transmission data $D_i, i \in \mathcal{N}$ and has CPU cycles required $F_i, i \in \mathcal{N}$, as shown in Table I.

### A. Performance for Local Decision Algorithm

In this subsection, we assess the performance of local decision algorithm, by using the 3-th UE as an example.

Fig. 4 shows the power consumption of the 3-th UE versus the required CPU cycles of the computation, i.e. $F_3$. In this figure, how the 3-th UE makes local decision based on the change of
$F_3$ is shown. We draw the maximal power which the UE can achieve, i.e., $P_{3,\text{max}}$ as the threshold in the figure. It can be seen that with the increase of $F_3$, more local power consumption $P_{3}^{L_*}$ is consumed. This is because with the increase of $F_3$, the required computing resource $f_{3}^{L_*}$ increases as well, resulting in more local power consumption. The minimal transmission power, i.e., $p_{3,\text{min}}^{Tr}$ keeps nearly the same across all the examined $F_3$, as the task offloading data, i.e., $D_3$ does not change here. Moreover, the UE cannot complete the computation locally when $F_3$ exceeds $10^6$ cycles, as the maximal CPU capacity $f_{i,\text{max}}^{L_*}$ is $1 \times 10^6$ cycles/s in 3-th UE. In this situation, the UE can offload the computation to the cloud with high priority, i.e., being added into set $\mathcal{O}^H$, if the minimal transmission power is lower than or equal to UE’s capacity, i.e., $p_{3,\text{min}}^{Tr} \leq P_{3,\text{max}}$. Otherwise this UE has to be added into set $\mathcal{R}$. If the minimal transmission power is larger than local execution power, i.e., $p_{3,\text{min}}^{Tr} > P_{3}^{L_*}$, the UE has to be added into set $\mathcal{L}$, otherwise, it can be added into set $\mathcal{O}^C$, if it can complete the task locally.

![Fig. 4. Power consumption of the 3-th UE versus the required CPU cycles of the computation, i.e. $F_3$.](image)

Fig. 5 shows the power consumption of the 3-th UE versus the required transmission data i.e. $D_3$ when offloading. In this figure, how the 3-th UE makes local decision based on the change of $D_3$ is also shown. It can be seen that the local computation power, i.e., $P_{3}^{L_*}$ keeps nearly the same across all the examined $D_3$, as required computation, i.e., $F_3$ does not change here. Moreover, with the increase of $D_3$, more transmission power is consumed, as expected. If the minimal transmission power $p_{3,\text{min}}^{Tr}$ is less than local execution power, i.e., $p_{3,\text{min}}^{Tr} \leq P_{3}^{L_*}$, the UE will be added into offloading set $\mathcal{O}$ (which can be either $\mathcal{O}^C$ or $\mathcal{O}^H$ depending on local execution capacity), otherwise it can be executed locally, i.e., to be added in set $\mathcal{L}$. From this
figure, it can be seen that it is not beneficial to offload the computation with a large amount of transmission data.

![Power consumption graph](image)

Fig. 5. Power consumption of the 3-th UE versus the required transmission data, i.e., $D_3$ when offloading.

**B. Performance for Central Decision Algorithm**

In this subsection, we examine the performance of central decision algorithm. We compare our proposed solutions with the following algorithms:

1) **All Local Execution (ALE)**, All the UEs execute the tasks locally. If it cannot complete the tasks locally, we assume the power consumption is 1 W (maximal power consumption which UEs can achieve).

2) **Joint Access Control (JAC)**, This algorithm was proposed in [32], which only consider accepting as many UEs as possible. However, this algorithm does not give UEs which cannot complete the tasks locally with higher priority.

3) **All Open (AO)**, All the RRH and fronthaul pairs are switched on in support of the UEs offloading.

Fig. 6 shows the number of UEs which can neither complete the tasks locally, nor offload to the cloud, i.e., UEs in set $R$ versus the capacity of BBU pool. It can be seen that by conducting ALE for all UEs, 14 out of 20 UEs are not able to complete the tasks. However, with the help of MC-RAN and by triggering offloading action, more UEs can complete their tasks. With increasing BBU pool’s capacity, more UEs can be supported, i.e., less UEs in set $R$. Our proposed CACA solution outperforms JAC, as CACA first accepts the UEs which cannot complete the
tasks locally, and then accepts the rest of UEs. However, JAC gives all the offloading UEs the same priority and therefore may drop some UEs which cannot complete the tasks themselves, resulting in them being moved to set $\mathcal{R}$.

![Image](image.png)

Fig. 6. The number of UEs which can neither complete the tasks locally, nor offload to the cloud, i.e., UEs in set $\mathcal{R}$ versus the capacity of BBU pool.

Fig. 7 shows the number of active RRH and fronthaul pairs, i.e., the size of $\mathcal{J}^*$ versus the capacity of BBU pool. AO which keeps all the RRH and fronthaul pairs on at all the time serves as a benchmark. CACA-CRAA indicates we conduct CACA first to do the access control and then apply CRAA to reduce system’s energy, while JCA-CRAA indicates we apply JAC first and then apply CRAA. One can see that AO needs the most resource, while two CRAA based algorithms, i.e., JCA-CRAA and CACA-CRAA do not need to have all the RRH and fronthaul pairs on at all the time to save energy. Also, with the increase of BBU pool’ capacity, the number of active RRH and fronthaul pairs by applying CACA-CRAA and JAC-CRAA are also increased. This is because with the increase of BBU pool’s capacity, the number of supported UEs increases, so that more RRH and fronthaul pairs are needed. Moreover, one can see that CACA-CRAA needs more RRH and fronthaul pairs than JAC-CRAA in most of the cases in order to support the UEs with high offloading priority. However, JAC-CRAA requires less RRH and fronthaul pairs, as it gives every UEs the same weight and only accept UEs which contribute to the whole energy minimization. Nevertheless, JAC-CRAA may result in more UEs which cannot complete their tasks, as explained before.

Fig. 8 shows the sum power consumption of MC-RAN and UEs versus the capacity of BBU
pool. JAC-AO indicates that we conduct JAC first whereas CACA-AO indicates that we first conduct CACA and then for both the algorithms, we do AO, i.e., switch on all the RRH and fronthaul pairs next. It can be seen that with the capacity increase of the BBU pool, the system power consumption increases, as expected. Also, CACA-CRAA and JAC-CRAA require less power consumption than the CACA-AO and JAC-AO, respectively. This is because CACA-CRAA and JAC-CRAA are able to switch some unnecessary RRH and fronthaul pairs into sleep mode to save power. Moreover, CACA-CRAA and CACA-AO requires a little bit more power consumption than JAC-CRAA and JAC-AO, respectively, as CACA-CRAA and CACA-AO needs to support the UEs with high priority offloading requests and those UEs may be scattered around more RRHs, resulting in more RRH and fronthaul pairs being switched on.

VIII. CONCLUSION

In this paper, we proposed a novel MC-RAN architecture, which can support UEs’ offloading and computation. Uplink offloading framework with local decision in each UE and central decision in MC-RAN are proposed. For local decision, DLDA was proposed for UE to decide if it can complete the tasks locally and if offloading is in its interest. Then, central decision including CACA and CRAA were proposed, first to accept offloading request from UEs which cannot complete the tasks locally, then accept the rest of UEs for the local energy saving purpose and finally to minimize the whole energy consumption, by proper allocating the resource and
switching off unnecessary hardware of MC-RAN. Simulation results have been provided to show the effectiveness of the proposed architecture and algorithms.

ACKNOWLEDGEMENT

This work was supported by UK EPSRC NIRVANA project (EP/L026031/1) and EU Horizon 2020 iCIRRUS project (GA-644526). Also, we would like to thank the Editor and all the anonymous Reviewers for their valuable time and constructive suggestions.

REFERENCES

[1] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 945–953.

[2] K. Kumar and Y.-H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” *Computer*, vol. 43, no. 4, pp. 51–56, April 2010.

[3] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, “Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks,” *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 45–55, Nov 2014.

[4] X. Chen, “Decentralized computation offloading game for mobile cloud computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, April 2015.

[5] Amazon, “Amazon elastic compute cloud,” (EC2), [https://aws.amazon.com/ec2/](https://aws.amazon.com/ec2/)

[6] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, and A. Neal, “Mobile edge computing white paper,” *ETSI*, 2014.

[7] H. Zhu and J. Wang, “Chunk-based resource allocation in ofdma systems, part i: chunk allocation,” *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2734–2744, September 2009.

[8] ——, “Chunk-based resource allocation in ofdma systems, part ii: Joint chunk, power and bit allocation,” *IEEE Transactions on Communications*, vol. 60, no. 2, pp. 499–509, February 2012.
[9] H. Zhu, “Radio resource allocation for ofdma systems in high speed environments,” IEEE Journal on Selected Areas in Communications, vol. 30, no. 4, pp. 748–759, May 2012.

[10] T. Nakamura, S. Nagata, A. Benjebbour, Y. Kishiyama, T. Hai, S. Xiaodong, Y. Ning, and L. Nan, “Trends in small cell enhancements in LTE advanced,” IEEE Communications Magazine, vol. 51, no. 2, pp. 98–105, February 2013.

[11] X. Rao and V. Lau, “Distributed fronthaul compression and joint signal recovery in Cloud-RAN,” IEEE Transactions on Signal Processing, vol. 63, no. 4, pp. 1056–1065, February 2015.

[12] China Mobile Research Institute, “C-RAN white paper: The road towards green RAN. [online],” (June 2014), Available: http://labs.chinamobile.com/cran.

[13] C. Pan, H. Zhu, N. J. Gomes, and J. Wang, “Joint precoding and RRH selection for user-centric green MIMO C-RAN,” to appear in IEEE Transactions on Wireless Communications, 2017.

[14] J. Tang, W. P. Tay, and T. Quek, “Cross-layer resource allocation with elastic service scaling in cloud radio access network,” IEEE Transactions on Wireless Communications, vol. 14, no. 9, pp. 5068–5081, Sept 2015.

[15] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, “Cloud RAN for mobile networks - a technology overview,” IEEE Communications Surveys Tutorials, vol. 17, no. 1, pp. 405–426, Firstquarter 2015.

[16] C. L. I, J. Huang, R. Duan, C. Cui, J. . Jiang, and L. Li, “Recent progress on C-RAN centralization and cloudification,” IEEE Access, vol. 2, pp. 1030–1039, 2014.

[17] K. Wang, K. Yang, and C. Magurawalage, “Joint energy minimization and resource allocation in C-RAN with mobile cloud,” IEEE Transactions on Cloud Computing, vol. PP, no. 99, pp. 1–11, 2016.

[18] K. Wang, K. Yang, X. Wang, and C. Magurawalage, “Cost-effective resource allocation in C-RAN with mobile cloud,” in IEEE International Conference on Communications (ICC), 2016, pp. 1–6.

[19] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. Wu, “Energy-optimal mobile cloud computing under stochastic wireless channel,” IEEE Transactions on Wireless Communications, vol. 12, no. 9, pp. 4569–4581, September 2013.

[20] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, 2010, p. 4.

[21] J. Tang, W. P. Tay, and Y. Wen, “Dynamic request redirection and elastic service scaling in cloud-centric media networks,” IEEE Transactions on Multimedia, vol. 16, no. 5, pp. 1434–1445, August 2014.

[22] L. Guangjie, Z. Senjie, Y. Xuebin, L. Fanglan, N. Tin-tokk, S. Zhang, and K. Chen, “Architecture of GPP based, scalable, large-scale c-ran bbu pool,” in 2012 IEEE Globecom Workshops (GC Wkshps), Dec 2012, pp. 267–272.

[23] N. Nikaein, “Processing radio access network functions in the cloud: Critical issues and modeling,” in Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services, 2015, pp. 36–43.

[24] K. Wang, K. Yang, H. H. Chen, and L. Zhang, “Computation diversity in emerging networking paradigms,” IEEE Wireless Communications, vol. 24, no. 1, pp. 88–94, February 2017.

[25] B. Dai and W. Yu, “Sparse beamforming and user-centric clustering for downlink cloud radio access network,” IEEE Access, vol. 2, pp. 1326–1339, 2014.

[26] Y. Zhou and Y. Xu and W. Yu and J. Chen, “On the optimal fronthaul compression and decoding strategies for uplink cloud radio access networks,” IEEE Transactions on Information Theory, vol. 62, no. 12, pp. 7402–7418, Dec 2016.

[27] S. H. Park and O. Simeone and O. Sahin and S. Shamai Shitz, “Fronthaul compression for cloud radio access networks: Signal processing advances inspired by network information theory,” IEEE Signal Processing Magazine, vol. 31, no. 6, pp. 69–79, Nov 2014.

[28] A. Fehske and G. Fettweis and J. Malmodin and G. Biczok, “The global footprint of mobile communications: The ecological and economic perspective,” IEEE Communications Magazine, vol. 49, no. 8, pp. 55–62, August 2011.
[29] S. Luo, R. Zhang, and T. J. Lim, “Downlink and uplink energy minimization through user association and beamforming in C-RAN,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 494–508, Jan 2015.

[30] M. Schubert and H. Boche, “Iterative multiuser uplink and downlink beamforming under sinr constraints,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2324–2334, July 2005.

[31] A. Wiesel, Y. Eldar, and S. Shamai, “Linear precoding via conic optimization for fixed MIMO receivers,” *IEEE Transactions on Signal Processing*, vol. 54, no. 1, pp. 161–176, January 2006.

[32] E. Matskani, N. D. Sidiropoulos, Z. q. Luo, and L. Tassiulas, “Convex approximation techniques for joint multiuser downlink beamforming and admission control,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2682–2693, July 2008.

[33] Q. T. Dinh and M. Diehl, “Local convergence of sequential convex programming for nonconvex optimization,” *Recent Advances in Optimization and its Applications in Engineering*, pp. 93–102, 2010.

[34] G. Scutari and F. Facchinei and L. Lampariello, “Parallel and distributed methods for constrained nonconvex optimization part i: Theory,” *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 1929–1944, April 2017.

[35] G. Scutari and F. Facchinei and L. Lampariello and S. Sardellitti and P. Song, “Parallel and distributed methods for constrained nonconvex optimization-part ii: Applications in communications and machine learning,” *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 1945–1960, April 2017.

[36] M. Grant and S. Boyd., “CVX: Matlab software for disciplined convex programming, version 3.0,” (June 2015), Available: http://cvxr.com/cvx.