IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 27, 2017, accepted August 19, 2017, date of publication September 14, 2017,
date of current version October 12, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2752279

Multi-Layer Cloud-RAN With Cooperative Resource Allocations for Low-Latency Computing and Communication Services

HAIBO MEI[1], KEZHI WANG[2], (Memeber, IEEE), AND KUN YANG[1,3], (Senior Member, IEEE)
[1]School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE2 1XE, U.K.
[3]School of Computer Sciences and Electrical Engineering, University of Essex, Colchester CO4 3SQ, U.K.

Corresponding author: Haibo Mei (haibo.mei@uestc.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61620106011 and Grant 61572389, in part by NIRVANA, in part by iCIRRUS, and in part by the EU FP7 Project CROWN under Grant GA-2013-610524.

ABSTRACT To improve low-latency computing and communication services, a new type of mobile edge computing architecture named multi-layer cloud radio access network (Multi-layer CRAN) is designed in this paper. In Multi-layer CRAN, a high-level edge cloud is deployed next to base band unit pool to handle the computing tasks of user equipment (UE) in centralized way. Meanwhile, a low-level edge cloud is deployed in each remote radio head (RRH) to locally handle UEs' computing tasks in a distributed way. Based upon Multi-layer CRAN, a cooperative communication and computation resource allocation (3C-RA) algorithm is further designed for lower service latency and energy cost, and higher network throughput in this paper. 3C-RA utilizes a distributed RRH cell coloring algorithm to enable each RRH to work out the resource allocation in an efficient and distributed way. 3C-RA employs a proportional fairness-based approach to allocate communication and computation resource in each RRH cell. A series of simulations on Multi-layer CRAN with 3C-RA were carried out. The simulation results validate that Multi-layer CRAN is more capable of providing low-latency computing and communication services, and 3C-RA enables Multi-layer CRAN to have lower service latency and energy cost and higher network throughput.

INDEX TERMS Multi-layer CRAN, communication and computation resource allocation, high-level edge cloud, low-level edge cloud.

I. INTRODUCTION

Currently, Cloud Radio Access Network (CRAN) as a type of network paradigm has been gradually deployed in countries such as China etc. [1]. CRAN promotes the merits of the cloud computing for use in mobile networks, and the ability to run a low Capital Expenditure and Operating Expenditure [2]. In reality, with the increase in popularity of high definition video, gaming, virtual reality, more and more resource-hungry tasks come into play in User Equipments (UEs). However, due to the limited UE resource, such as CPU, storage etc., it is very difficult for a UE to process those resource intense applications. To deal with this issue, Mobile Edge Computing (MEC) [3], [4] was proposed as an emerging technique in 5G networks to extend the computational capacity of UEs. MEC can enable a UE to offload the computing tasks to the cloud to lighten its load and to have a longer battery life. The authors of [5], [6] proposed applying MEC to extend the UEs computational capacities of cellular networks and Heterogeneous Networks (HetNets). In [7], a new type of CRAN was designed to have a MEC along with Base Band Unit (BBU) pool, which is the main concern of this paper.

CRAN working with MEC is still in its infancy. In practice, CRAN with MEC may not be able to provide the highly desired low-latency computing and communication services to act as an enabler for new potential IoT applications. In CRAN with MEC, if a UE offloads computing tasks to the remote cloud, the UE may experience a poorer Quality of Service (QoS). This is because when a UE transmitting intense data to the remote cloud through a constrained fronthaul, it may cause intolerable time delay. In addition, the capacity of a frounthaul is limited, so one fronthaul may not able to accommodate all the incoming UE requests.

To address above issues, in this paper, we propose a new type of Multi-layer MEC architecture, named Multi-layer CRAN. Multi-layer CRAN makes use of local clouds to handle proximity UEs' tasks to decrease the latency and to save

fronthaul capacity. In Multi-layer CRAN, a centralized High-level Edge Cloud (HEC) is developed next to the BBU pool to handle most of the offloaded computing tasks from Remote Radio Heads (RRHs). A Low-level Edge Cloud (LEC) is developed in each RRH to locally handle users' time sensitive tasks or the tasks not suitable to be offloaded to the HEC. Due to the transmission time saved in the fronthaul, Multi-layer CRAN therefore has the potential to better deliver low-latency computing and communication services.

Because of the separations of HEC and LEC in Multi-layer CRAN, the Cooperative Communication and Computation Resource Allocation becomes critical. In Multi-layer CRAN, the allocations of the communication and computation resource should be cooperatively done among RRHs. To fully unleash the potential advantages of Multi-layer CRAN, the resource allocation should minimize service latency and energy cost, and maximize the network throughput, and have time delay awareness and be scalable with respect to the network size. This makes traditional resource allocation methods for CRAN infeasible, due to their computational complexities as well as the signaling latency involved. The issue becomes even worse for the case of a Multi-layer CRAN having an increased network size.

To address above issue, we propose a Cooperative Communication and Computation Resource Allocation (3C-RA) algorithm in this paper to facilitate the resource allocation. 3C-RA utilizes a Distributed RRH Cell Coloring Algorithm (DRCC) to enable each RRH to carry out the resource allocation in a distributed way with efficiency. In each RRH cell, the communication resource allocation employs a proportional fairness based approach coupled with computation resource allocation. This paper carried out simulations to validate Multi-layer CRAN with 3C-RA. The simulation results show that Multi-layer CRAN can better provide low-latency computing and communication services with low energy cost. Also 3C-RA can enable Multi-layer CRAN to have a higher network throughput. In summary, the primary contributions of this paper are:

1) A Multi-layer MEC architecture is proposed to deliver low-latency computing and communication services. Based on the architecture, a Multi-layer CRAN is designed in this paper. Multi-layer CRAN has a HEC working as the central cloud to handle UE tasks with intensive computing. Meanwhile, Multi-layer CRAN has a number of LECs working as local clouds to handle proximity UE tasks that are sensitive to latency. This Multi-layer MEC architecture is to avoid UEs offloading data intense UE tasks to the remote cloud through constraint fronthauls. Multi-layer CRAN therefore can save UE task latency and fronthaul capacity.
2) A 3C-RA algorithm is designed to help Multi-layer CRAN make better use of the scarce computing and communication resource. 3C-RA employs a proportional fairness based approach for communication resource allocation coupled with computation resource allocation. Through 3C-RA, Multi-layer CRAN can have increased network throughput, improved low-latency computing and communication services, and decreased energy cost.
3) 3C-RA works in a distributed way and promises the efficiency through a DRCC algorithm. Each RRH in Multi-layer CRAN can efficiently carry out the resource allocation in distributed way. Therefore, 3C-RA has time delay awareness and scalability against the dynamic size and user requirements of Multi-layer CRAN.

The remainder of this paper is organized as follows. In section II, we give the related work. In section III, we describe Multi-layer CRAN system model and formulate related resource allocation problem. In Section IV, we present the 3C-RA algorithm. We discuss 3C-RA converging to optimal solution in section V. Simulation results and analysis are presented in Section VI. In Section VII, we give conclusions and future work.

II. RELATED WORK

There is a number of work proposed to help CRAN providing low-latency computing and communication services with MEC, such as [5]–[8]. However, that work cannot solve the task latency issues under severe situations, such as most of the UE tasks having to be offloaded to the remote cloud through congested fronthauls.

Some work has tried to release the fronthaul constraint to improve the performance of CRAN with MEC. For example in [9], a type of CRAN named a Heterogeneous CRAN (H-CRAN) was proposed, where user and control planes are decoupled. In H-CRAN, High Power Nodes (HPNs) are mainly used to provide seamless coverage and execute the functions of the control plane, while RRHs are deployed to provide a high-speed data rate for packet traffic transmission in the user plane. In [10], H-CRAN with multiple clouds was discussed, where a set of base stations share a local cloud. Generally, H-CRAN mainly utilizes macro base stations to work out the network controlling tasks to release the fronthaul constraint. However, H-CRAN involves complicated resource allocation [11], [12], and the way that macro base stations release the fronthaul constraint cannot fully solve the task latency issue [18].

In contrast to the work in [5]–[12], offloading computational tasks to local clouds as a more effective solution was investigated and several platforms have been proposed in [13]–[16]. For example in [16], the authors tried to realize a type of distributed mobile cloud computing through a multi-user clustering solution. The solution focuses on managing resource for the set of radio access points forming the local cloud. All the work in [13]–[16] tried to make use of local cloud to handle UE tasks instead of offloading intense data to the remote cloud. However, for the case of a network having majority of the UEs that have high mobility, an un-balanced distribution and operate dynamic tasks, the work in [13]–[16] still cannot handle the task latency issue.

To implement local cloud with flexibility while maximally exploring network participants' computing capabilities,

a new cloud computing technology named fog computing has been proposed into mobile networks [17]. Fog computing is a term for an alternative to cloud computing that puts a substantial amount of storage, communication, control, configuration, measurement, and management at the edge of a network, rather than establishing channels for the centralized cloud storage and utilization. Therefore, fog computing extends the traditional cloud computing paradigm to the network edge. A new network paradigm named Fog computing based RAN (F-RAN) has been proposed in [18] and [19]. In F-RAN, RRHs and UEs can work as fog nodes to help release the pressure on fronthauls and the centralized cloud. In [18], the authors discussed how to allocate the amounts of assigned tasks for each fog node and the data sharing process to balance the computing and communication costs guaranteeing low-latency applications. In [19], the authors discussed the computing and communication tradeoff and unique characteristics for a F-RAN architecture with ultra low-latency applications. F-RAN is still in its infancy. There are still quite a number of outstanding problems that need further investigation, such as UEs transmission modes selection, interference suppression, UEs coordinated scheduling etc. Fog computing also suffers from a lacking of prosperous resource coordination [20]. In [20], the authors explored the possibility that CRAN and F-RAN complement each other in engineering practice, a harmonization between H-CRAN and the fog network was proposed.

In this paper, we still employ the local cloud idea to help CRAN provide low-computing and communication services with multi-layer MEC. In contrast to the work in [13]–[20], we propose a Multi-layer CRAN to better deploy local clouds close to UEs by rational network planning. Further supported by cooperative communication and computation allocation mechanism, such as 3C-RA, Multi-layer CRAN can improve low-latency computing and communication services even for the case of a network having majority of the UEs that have high mobility, an un-balanced distribution and operate dynamic tasks. Compared to F-RAN, Multi-layer CRAN does not suffer from a lack of prosperous resource coordination, and reduces the need for complex work, such as UEs transmission modes selection, interference suppression, and UEs coordinated scheduling etc.

III. MULTI-LAYER CRAN SYSTEM MODEL AND RELATED RESOURCE ALLOCATION PROBLEM

A. SYSTEM MODEL

1) USER SERVICE LATENCY AND ENERGY COST

We assume there are C RRHs in Multi-layer CRAN and each of which $j = 1, 2, ..., C$ forms a small cell. In a RRH cell j, there are N_j UEs supported by this cell. UE $i = 1, 2, ..., N_j$ in the coverage of j-th RRH is denoted as ij-th UE. The task of ij-th UE is formulated as

$$U_{ij} = (F_{ij}, D_{ij}), \quad \forall i \in N_j, \ \forall j \in C \tag{1}$$

where F_{ij} describes the total number of the CPU cycles needed to be completed for task U_{ij}, while D_{ij} denotes the whole size of the task's output data transmitting to the ij-th UE through CRAN after task execution, including the task's output parameter and calculation results etc. [8].

Based on (1), the latency of finishing the task from ij-th UE in RRH cell j is formulated as

$$T_{ij} = \frac{F_{ij}}{f_{ij}} + \frac{D_{ij}}{r_{ij}}, \quad \forall i \in N_j, \ \forall j \in C \tag{2}$$

where f_{ij} is the allocated computation capabilities serving UE task U_{ij}, and r_{ij} is the data rate of ij-th UE supported by RRH j. f_{ij} and r_{ij} will be further discussed and formulated in (6) and (7) respectively.

We define E_{ij} as the energy cost of the UE task U_{ij} in cell j, which is formulated as

$$E_{ij} = \varphi(f_{ij})^{\vartheta-1} F_{ij} + \eta P_j(\frac{D_{ij}}{r_{ij}}), \quad \forall i \in N_j, \ \forall j \in C \tag{3}$$

where $\varphi \geq 0$ is the effective switched capacitance and $\vartheta \geq 1$ is the positive constant [21]. According to the realistic measurements, φ can be set to $\varphi = 10^{-11}$ [22]. $\eta \geq 0$ is a weight to the tradeoff between the energy consumptions in the mobile cloud and CRAN, and it can be also explained as the inefficiency coefficient of the power amplifier at RRH. P_j represents the power of RRH j.

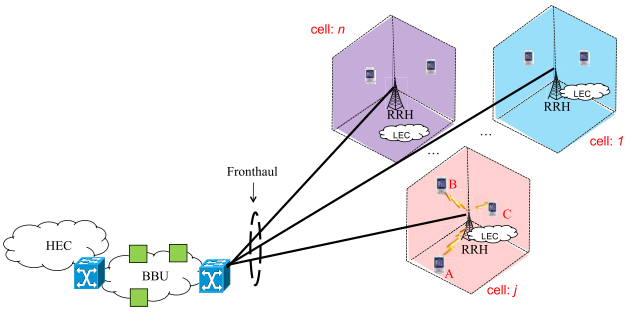

FIGURE 1. System Architecture on Multi-layer CRAN.

2) MULTI-LAYER CRAN ARCHITECTURE

In order to provide low-latency computing and communication services, the latency and energy cost of each UE task formulated in (2) (3) should be effectively decreased. Multi-layer CRAN is to fulfill the requirement, and its structure is shown in Fig.1. In Multi-layer CRAN, the LEC in a RRH is mainly used to serve time sensitive tasks of proximity UEs, while the HEC next to the BBU pool is mainly deployed to serve the computing intensive tasks offloaded by RRHs. From users' perspectives, a UE is served by a RRH with a LEC, only if the task of the UE is time delay sensitive that is not suitable to be offloaded to the HEC. We define such a UE as a LUE. In contrast, a UE served by a RRH and the HEC, if the task of the UE is not very sensitive to latency but involves intense computing. We define such a UE as a HUE.

3) DATA RATE IN MULTI-LAYER CRAN

In Multi-layer CRAN, the data rate of a UE is closely related to the output of the employed communication resource allocation mechanism, which decides the quality of the channels of the UE. The communication resource in Multi-layer CRAN has K OFDM based Radio Blocks (RBs) with a total bandwidth B. According to frequency reuse [23], each RRH cell in Multi-layer CRAN can have all the K RBs leading the frequency reuse factor to be 1.

We employ the Signal to Interference plus Noise Ratio (SINR) to evaluate the channel quality of a RB in Multi-layer CRAN. In RRH cell j, if RB k is allocated to ij-th UE, the SINR of RB k is formulated as

$$S_{ikj} = \frac{\frac{P_j}{K}(d_{ij})^{-1}h_{ikj}}{\sum_{\forall t \in Q_j}\frac{P_t}{K}(d_{it})^{-1}h_{ikt} + N_0} \tag{4}$$

where d_{ij} is the distance from ij-th UE to its associated RRH j. h_{ikj} represents the channel gain of RB k from ij-th UE to the associated RRH j. P_j is the power of RRH j. N_0 denotes the estimated power of noise in cell j (in dBm). Q_j is the group including all the external and proximity interfering RRHs to cell j. A UE will receive inter cell interference from the RRHs in group Q_j, if its allocated RBs are used by those interfering RRHs in Q_j at the same time.

According to the RB SINR formulated in (4), the data rate of RB k serving ij-th UE in RRH cell j can be expressed as

$$r_{ijk} = B \cdot log_2(1 + S_{ikj}) \tag{5}$$

According to (5), the data rate of ij-th UE served by RRH cell j is formulated as

$$r_{ij} = \sum_{k=1}^{K} \alpha_{ikj} r_{ikj} \tag{6}$$

where α_{ikj} represents the RB allocation policy for UEs in RRH cell j. $\alpha_{ikj} = 1$ means RB k is allocated to ij-th UE, while $\alpha_{ikj} = 0$ means not.

4) COMPUTATION CAPACITY IN MULTI-LAYER CRAN

In Multi-layer CRAN, the available communication resource is represented as certain amount of RBs, each of which is an atom resource unit. Similarly, we define a Computing Block (CB) as the atom computation resource unit, which has computation capacity Δf. Accordingly, the computation resource allocated to ij-th UE in cell j is represented as

$$f_{ij} = \sum_{f=1}^{F_j^{max}} \beta_{ijf} \Delta f \tag{7}$$

where β_{ijf} denotes whether CB f allocated to ij-th UE ($\beta_{ijf} = 1$) or not ($\beta_{ijf} = 0$) in cell j. F_j^{max} represents the number of CBs available to ij-th UE in cell j, which is composed by two parts i.e. $F_{j,LEC}^{max}$ from LEC and $F_{j,HEC}^{max}$ from HEC and $F_j^{max} = F_{j,HEC}^{max} + F_{j,LEC}^{max}$.

B. PROBLEM FORMULATION

In Multi-layer CRAN, UE tasks should be served by Multi-layer CRAN with acceptable latency. For ij-th UE in cell j, its task latency is close related to the communication resource allocation i.e. $\alpha_j = [\alpha_{ikj}]_{N_j \cdot K}$, and the computation resource allocation i.e. $\beta_j = [\beta_{ijf}]_{N_j \cdot F_j^{max}}$. To describe this problem, we define a utility function on UE task latency as

$$G_{ij}(\alpha_j, \beta_j) = \begin{cases} 1, & \text{if } T_{ij} \leqslant T_{ij}^{max} \\ 0, & \text{if } T_{ij} > T_{ij}^{max} \end{cases} \tag{8}$$

where if ij-th UE in cell j having its task handled in acceptable latency i.e. T_{ij}^{max}, G_{ij} is set to 1, otherwise G_{ij} is set to 0.

The main objective of this paper therefore is to maximize the sum of the task latency utilities of all the UEs in a RRH cell j. This is formulated as

$$G_j(\alpha_j, \beta_j) = \sum_{\forall i \in N_j} G_{ij}(\alpha_j, \beta_j) \tag{9}$$

According to (9), the optimization problem of RRH cell j is formulated in (10). Problem $\mathcal{P}$ in (10) aims at maximizing the number of UEs that successfully have their tasks handled in acceptable latency in each cell, with prerequisites formulated in (11)-(13). (11) denotes the overall energy cost caused by the UE tasks in cell j should not exceed the maximal allowed constraint: E_j^{max}. (12) denotes the total number of RBs allocated to UEs in cell j should not exceed the number of available RBs. (13) denotes the overall size of the computation capacities allocated from HEC and LEC to UEs should be less than the maximum allowed capacities in cell j i.e. $F_j^{max} = F_{j,LEC}^{max} + F_{j,HEC}^{max}$.

$$\mathcal{P}: \max_{\alpha_j, \beta_j} G_j(\alpha_j, \beta_j), \quad \forall j \in C \tag{10}$$

$$s.t. \sum_{\forall i \in N_j} E_{ij} \leqslant E_j^{max} \tag{11}$$

$$\sum_{\forall i \in N_j} \sum_{k=1}^{K} \alpha_{ikj} \leqslant K \tag{12}$$

$$\sum_{\forall i \in N_j} \sum_{f=1}^{F_j^{max}} \beta_{ijf} \leqslant F_j^{max} \tag{13}$$

IV. COOPERATIVE COMMUNICATION AND COMPUTATION RESOURCE ALLOCATION

To solve problem $\mathcal{P}$, the 3C-RA algorithm in this paper cooperatively works out the communication and computation resource allocation of each RRH cell of Multi-layer CRAN to decrease UE task latency and energy cost. 3C-RA utilizes a cell coloring based distributed resource allocation approach to enable each RRH cell to carry out the resource allocation in an efficient and distributed way. When 3C-RA completes, the resource allocation results: (α_j, β_j, $\forall j \in C$) are obtained solving problem $\mathcal{P}$.

A. 3C-RA FOR A SINGLE RRH CELL

We first discuss 3C-RA for a single RRH cell. According to problem $\mathcal{P}$ and related constraints, 3C-RA intends to maximize the number of UEs having their task handled in acceptable latency in a RRH cell j. To solve this problem, 3C-RA assumes all the UEs in cell j to have their tasks finished within T_{ij}^T. T_{ij}^T is the configurable latency threshold, and ($T_{ij}^T \leqslant T_{ij}^{max}, \forall i \in N_j$). In this way, 3C-RA can work out the communication and computation resource allocation with a pre-define prerequisite, which makes sure 3C-RA fully considering the UE latency requirements. Based on this assumption, we therefore change (3) into (14), with $\eta = 1$ and $\vartheta = 2$.

$$E_{ij} = \frac{\varphi(F_{ij})^2 r_{ij} + P_j D_{ij} T_{ij}^T r_{ij} - P_j (D_{ij})^2}{T_{ij}^T (r_{ij})^2 - D_{ij} r_{ij}} \tag{14}$$

According to (14), if setting the energy cost E_{ij} of ij-th UE equal to a configurable threshold: E_{ij}^T, we can calculate out r_{ij} as the data rate that can guarantee UE task U_{ij} finished with allowed latency T_{ij}^T and energy cost E_{ij}^T. r_{ij} is calculated out by solving (14) as a quadratic equation, in which r_{ij} is the only variable. The energy cost threshold E_{ij}^T is configurable and should be set to have $\sum_{i=1}^{N_j} E_{ij}^T \leqslant E_j^{max}$ to fulfill the energy cost constraint defined in (11). With such specified data rate of each UE, 3C-RA can easily solve the communication and computation resource allocation problem of a single cell by Algorithm 1.

1) ALGORITHM 1

In Algorithm 1, 3C-RA first works out the communication resource i.e. RBs allocation of RRH cell j from step 2 to 3 for all the UEs. The RB allocation works based upon a proportional fairness approach [24]. At step 2, Algorithm 1 calculates out the requested data rata of each UE task by solving (14). For example of UE task U_{ij}, r_{ij} is calculated, guaranteeing UE task U_{ij} finished with latency T_{ij}^T and energy cost E_{ij}^T. Afterwards, Algorithm 1 works out the proportional fairness based RBs allocation at step 3.

After communication resource allocated, Algorithm 1 then works out the computation resource allocation for cell j. In order to do so, Algorithm 1 has to understand the computation capacity request and the mobile edge cloud i.e. LEC or HEC choice of each UE task. At Step 4 and 5 of Algorithm 1, the requested computation capacities of each UE task are calculated. Take ij-th UE as an example, f_{ij} is calculated by solving (2), subject to r_{ij} as the available data rate of the UE and T_{ij}^T as the maximally allowed latency of UE task U_{ij}. Step 6 and 7 of Algorithm 1 utilize a Inverse Cumulative Ranking method to enable $\lceil \frac{F_{j,LEC}^{max}}{F_{j,LEC}^{max}+F_{j,HEC}^{max}} N_j \rceil$ UEs with low latency task i.e. low T_{ij}^{max} to choose LEC as their computation service provider and join group L_j. However, if a ij-th UE with intense computing i.e. $f_{ij} > \phi\Delta f$, this UE will choose HEC instead. This is because LEC is not prone to handle computing intense tasks, as its computation capacity is limited. The way putting UEs with lower latency requirement to choose LEC is reasonable, as LEC can save latency without transmitting data through fronthaul to better fulfill those UEs' latency requirements. Apart form the UEs choosing LEC, the rest UEs in cell j will choose HEC and join group H_j.

Algorithm 1 3C-RA for RRH Cell j

1 Task inputs: $T_{ij}^{max}, T_{ij}^T, E_{ij}^T, \forall i \in N_j$;
2 Calculate $(r_{ij}, \forall i \in N_j)$ by solving (14) considering $E_{ij} = E_{ij}^T,\ T_{ij} = T_{ij}^T$;
3 Carry out RB allocation of cell j based upon Proportional Fairness (See Algorithm 2), taking N_j and r_{ij} as the inputs;
4 $r_{ij} = \sum_{k=1}^{K} \alpha_{ikj} r_{ikj}, \forall i \in N_j$;
5 Calculate f_{ij} of each UE task by solving (2) considering T_{ij}^T and r_{ij};
6 Rank each UEs in cell j through Inverse Cumulative Ranking according to T_{ij}^{max};
7 Group $\lceil \frac{F_{j,LEC}^{max}}{F_{j,LEC}^{max}+F_{j,HEC}^{max}} N_j \rceil$ UEs which are in lower rank into L_j as LUEs subject to ($f_{ij} < \phi\Delta f$), and the rest UEs grouped into H_j as HUEs;
8 Carry out CB allocation from HEC to HUEs (grouped as H_j) of cell j based upon Proportional Fairness (See Algorithm 3), taking H_j, f_{ij} and $F_{j,HEC}^{max}$ as the inputs;
9 Carry out CB allocation from LEC to LUEs (grouped as L_j) of cell j based upon Proportional Fairness (See Algorithm 3), taking L_j, f_{ij} and $F_{j,LEC}^{max}$ as the inputs;
10 return (α_j, β_j);

After the computation capacity request and mobile edge cloud choice of each UE task being decided, Algorithm 1 carries out step 8 to allocate computation resource i.e CBs from HEC to each HUEs (UEs in group H_j) based on the proportional fairness approach specified in Algorithm 3. Similarly, the CBs allocation from LEC to each LUEs (UEs in group L_j) is carried out at step 9 of Algorithm 1. After CBs allocation done, 3C-RA finishes and returns the resource allocation results of cell j: (α_j, β_j) at step 10.

2) ALGORITHM 2

3C-RA utilizes Algorithm 2 to work out the communication resource allocation of cell j in a proportional fairness way. Algorithm 2 works out the RBs allocation mainly in outer loops from step 3 to 18. In one outer loop, the proportional fairness based RBs allocation is carried out through inter loops from step 6 to 16. In one inter loop, there will be a RB k^* allocated to UE i^*, which has the highest proportional fairness value (step 8 of Algorithm 2). However, if a candidate UE i^* already has enough RBs allocated to have data rate $R_{i^*} = r_{i^*,j}$ (step 10 of Algorithm 2), this UE will not be allocated RB any more to avoid greedy. The proportional fairness values of each RB to UE pair e.g.P_{ik} are calculated by step 7 of Algorithm 2. A proportional fairness value takes the product of the requested data rate of the candiate UE,

Algorithm 2 RB Allocation for Cell j Based Upon Proportional Fairness

1 Task inputs: I_j, $(r_{ij}, \forall i \in I_j)$;
2 $G = 0$, $K_d = \{\}$,$s = 0$;
3 **while** $(G < \lceil \xi \times I_j \rceil) \& (s < S_{max})$ **do**
4 $\quad R_i(t+1) = \frac{(t_c-1)R_i(t)+\sum_{k=1}^{K} \alpha_{ikj} r_{ikj}}{t_c}, \forall i \in I_j$;
5 $\quad \alpha_{ikj} = 0, (\forall i \notin K_d, \forall k \in K)$;
6 $\quad$ **while** $(\exists k \in K \rightarrow \sum_{\forall i \in I_j} \alpha_{ikj} = 0)$ **do**
7 $\quad\quad P_{ik} = \frac{r_{ij} r_{ikj}}{(t_c-1)R_i(t+1)+\sum_{k=1}^{K} \alpha_{ikj} r_{ikj}}, \forall i \in I_j$;
8 $\quad\quad (i^*, k^*) = \text{argmax}_{\forall i \in I_j}(i, k)[P_{ik}]$;
9 $\quad\quad R_{i^*} = \sum_{k=1}^{K} \alpha_{i^*kj} r_{i^*kj}$;
10 $\quad\quad$ **if** $R_{i^*} < r_{i^*j}$ **then**
11 $\quad\quad\quad \alpha_{i^*k^*j} = 1$;
12 $\quad\quad$ **else**
13 $\quad\quad\quad G = G + 1$;
14 $\quad\quad\quad K_d = K_d \cup \{i^*\}$;
15 $\quad\quad$ **end**
16 $\quad$ **end**
17 $\quad s = s + 1$;
18 **end**
19 return α_j;

Algorithm 3 CB Allocation for Cell j Based Upon Proportional Fairness

1 Task inputs: I_j, $(f_{ij}, \forall i \in I_j)$, F_j^{max};
2 $G = 0$, $F_d = \{\}$, $s = 0$;
3 **while** $(G < \lceil \mu \times I_j \rceil) \& (s < S_{max})$ **do**
4 $\quad F_i(t+1) = \frac{(t_c-1)F_i(t)+\sum_{f=1}^{F_j^{max}} \beta_{ifj} \Delta f}{t_c}, \forall i \in I_j$;
5 $\quad \beta_{i,k,j} = 0, (\forall i \notin F_d, \forall k \in K)$;
6 $\quad$ **while** $(\exists f \in F_j^{max} \rightarrow \sum_{i \in I_j} \beta_{ifj} = 0)$ **do**
7 $\quad\quad P_{if} = \frac{f_{ij}}{(t_c-1)F_i(t+1)+\sum_{f=1}^{F_j^{max}} \beta_{ifj} \Delta f}, \forall i \in I_j$;
8 $\quad\quad (i^*, f^*) = \text{argmax}_{\forall i \in I_j}(i, f)[P_{if}]$;
9 $\quad\quad F_{i^*} = \sum_{f=1}^{F_j^{max}} \beta_{i^*fj} \Delta f$;
10 $\quad\quad$ **if** $F_{i^*} < f_{i^*j}$ **then**
11 $\quad\quad\quad \beta_{i^*f^*j} = 1$;
12 $\quad\quad$ **else**
13 $\quad\quad\quad G = G + 1$;
14 $\quad\quad\quad F_d = F_d \cup \{i^*\}$;
15 $\quad\quad$ **end**
16 $\quad$ **end**
17 $\quad s = s + 1$;
18 **end**
19 return β_j;

e.g. $r_{i,j}$ of ij-th UE, and the data rate of the candidate RB, e.g. $r_{i,k,j}$ of RB k, as the numerator. This makes sure the UE with higher data rate requirement being more prone to get high quality RB allocated. The denominator of the proportional fairness value is to guarantee the RB allocation follows the proportional fairness way to avoid RB allocation bias [24]. The denominator is the sum of the data rate of the objective ij-th UE in last time slot t_c (i.e. $(t_c - 1)R_i(t + 1)$) and the real time data rate of ij-th UE (i.e. $\sum_{\forall k \in K} \alpha_{i,k,j} r_{i,k,j}$).

On finishing one outer loop, Algorithm 2 will check whether the number of the UEs successfully obtained enough RBs is higher than a threshold i.e $G \geqslant \lceil \xi \times I_j \rceil$ or not at step 3. ξ is the ratio of success UEs to the total number of UEs, which is a pre-defined parameter to determine the converge condition of Algorithm 2. If G is higher than the allowed threshold, Algorithm 2 will finish (converge) and return the output α_j at step 19. However, if G is not higher than the threshold, Algorithm 2 will invalid the RB allocations to the UEs that not obtained enough RBs (step 5) and carry on working out the RB allocations of those UEs in further steps. In addition, to guarantee algorithm finishing in finite steps in case of condition $G \geqslant \lceil \xi \times I_j \rceil$ not easily reached, Algorithm 2 will terminate if running out of allowed steps S_{max}.

3) ALGORITHM 3

In Algorithm 3, similar to the RB allocation, in each session of proportional fairness based CB allocation, a pair of CB f^* to UE i^* that has the highest proportional fairness value will be found and UE i^* will be allocated CB f^* (step 8 of Algorithm 3). However, if a candidate UE i^* already has enough CBs allocated to have computation capacity $F_{i^*} = f_{i^*j}$, this UE will not be allocated CB any more to avoid greedy (step 10 of Algorithm 3). The proportional fairness values of each CB to UE pair e.g.P_{if} are calculated by step 7 of Algorithm 3. A proportional fairness value takes the computation capacity requirement of a UE, e.g. f_{ij}, as the numerator to make sure the UE with higher computation capacity requirement being more prone to get CB allocated. The denominator of the proportional fairness value is to guarantee the CB allocation follows the proportional fairness way to avoid CB allocation bias. The denominator is the sum of the computation capacity of the objective ij-th UE in last time slot t_c (i.e. $(t_c - 1)F_i(t + 1)$) and the real time computation capacity of ij-th UE (i.e. $\sum_{f=1}^{F^{max}} \beta_{ifj} \Delta f$).

Similar to Algorithm 2, Algorithm 3 will converge if the number of the UEs successfully obtained enough CBs is higher than the allowed threshold i.e G $\geqslant \lceil \mu \times I_j \rceil$. Algorithm 3 will invalid the CB allocations to the UEs that not obtained enough CBs and carry on working out the CB allocations of those UEs in further steps, if the algorithm not able to converge yet. μ is the ratio of success UEs to the total number of UEs, which is a pre-defined parameter to determine the converge condition of Algorithm 3. Algorithm 3 will finish if the converge condition is satisfied or the algorithm running out of allowed steps S_{max}.

B. 3C-RA FOR THE WHOLE MULTI-LAYER CRAN

According to Algorithm 1, the 3C-RA algorithm can work out the resource allocation for a single cell of Multi-layer CRAN. 3C-RA can further utilize a cell coloring based distributed resource allocation approach to enable every RRH

cell in Multi-layer CRAN to carry out Algorithm 1 in an efficient and distributed way. When 3C-RA completes, the optimized resource allocation for the whole Multi-layer CRAN is obtained.

In traditional distributed resource allocation, when a RRH cell j carries out Algorithm 1, RRH cell j has to ask its neighbor cells in Q_j to stay static and not to carry out Algorithm 1 at the same time. Otherwise, The running of Algorithm 1 in RRH cell j will not able to converge and obtain the output. This is because if neighbor cells in Q_j not static and having their resource allocation changing all the time, cell j then does not able to have a deterministic prerequisite to work out its resource allocation. The way putting the neighbor cells to stay static during the resource allocation of a cell will tremendously decrease the resource allocation efficiency, and cause the overall resource allocation taking incredibly long time to finish. This problem will escalate in the case that the size of Multi-layer CRAN increases. To solve this issue, this paper proposes Algorithm 4 for 3C-RA working out resource allocation for the whole Multi-layer CRAN in distributed way with efficiency.

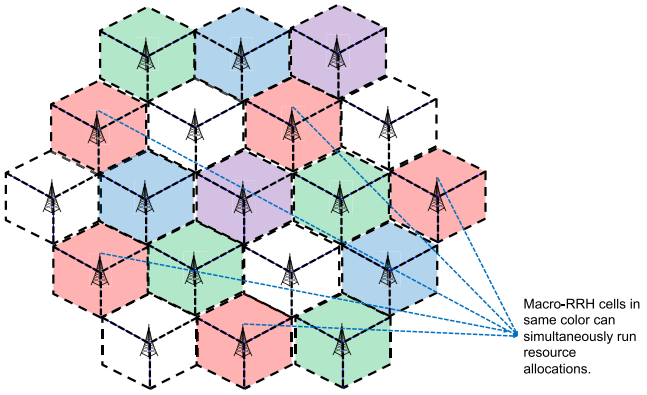

FIGURE 2. Color RRH cell nodes by DRCC guaranteeing no adjacent node having same color.

The cell coloring based distributed resource allocation approach was first introduced by the authors of [25], [26]. As shown in Fig.2, the non-neighbor RRH cells (in same color) can carry out the Algorithm 1 simultaneously, each of which can surely converge and get the resource allocation output. This is because the same colored RRH cells are geographically far away from each other, and any two of the running cells are not each other's neighbor. Therefore, during the simultaneous running, the neighbors of a running cell are static for the moment, which provides a deterministic prerequisite for the running cell to work out the resource allocation. By this simultaneous resource allocation within RRH cells, 3C-RA can effectively decrease the running time.

3C-RA for the whole Multi-layer CRN is designed in Algorithm 4. At step 1, 3C-RA colors RRH cells of Multi-layer CRAN to guarantee that no adjacent RRH cell has the same color, by using the DRCC algorithm. Afterwards, the resource allocation of RRH cells is carried out

Algorithm 4 3C-RA for the Whole Multi-Layer CRAN

```
1  Color RRH cells through Distributed RRH Cell Coloring
   Algorithm (DRCC)[25] by colors from group Z;
2  i=1, G_old = 0, G_new = 1, s = 0;
3  while (G_new > G_old)&(s < S_max) do
4      G_old = G_new ;
5      for (i ≤ |Z|) do
6          Simultaneously run Algorithm 1 for each of the
           cells with color Z[i];
7      end
8      G_new = Σ_{∀j∈C} G_j;
9      i = (i + 1)%|Z|;
10     s = s + 1;
11 end
12 return (α_j, β_j), ∀j ∈ C;
```

iteratively (step 3 to step 11). In one iteration, only one of the colors is considered and Algorithm 1 is simultaneously run to the cells in that color (step 6). In the next iteration, the 3C-RA algorithm moves to another color in set Z, and cycles until the end (step 9). Then 3C-RA works on the cells of that color simultaneously. The 3C-RA algorithm moves from one color to another color step by step until no resource allocation of a cell leading to the increment of the utility of network task latencies: G_{new} any more (3C-RA converges) or 3C-RA running out of allowed iteration counts (S_{max})(step 3). Finally, step 12 returns the results: (α_j, β_j), $\forall j \in C$, which is the resource allocation of every RRH cell of Multi-layer CRAN.

V. 3C-RA CONVERGE TO OPTIMAL SOLUTION

In 3C-RA, Algorithm 1 applies 3 mechanisms to make sure itself working towards the optimal resource allocation for each RRH cell of Multi-layer CRAN. First, 3C-RA applies the requested data rate and computation capacity of each UE task to calculate the proportional fairness value during the resource allocation (see step 7, 8 of Algorithm 2 and step 7, 8 of Algorithm 3). The requested data rate and computation capacity of a UE are calculated by solving (14) and (2). This enables the proportional fairness based resource allocation to be more prone to allocate resource to the UE tasks with intense data rate and computation capacity requests. Second, Algorithm 2 and 3 called by Algorithm 1 do not allocate any RB or CB to the UEs that already have enough resource allocated to void greedy (see step 10 of Algorithm 2 and step 10 of Algorithm 3). Third, Algorithm 2 and Algorithm 3 define their converge condition to make sure the number of UEs successfully allocated enough RBs and CBs is higher than a pre-defined threshold (see step 3 of Algorithm 2 and step 3 of Algorithm 3). If not able to converge to the conditions, Algorithm 2 or Algorithm 3 will invalid the resource allocation output of the UEs that not able to receive enough RBs or CBs, and try to work out the resource allocation for those UEs in further steps. This enables Algorithm 2 and Algorithm 3 to work out the communication and computation resource allocations closer to the optimal solution.

These 3 mechanisms make sure 3C-RA working out the resource allocation of each RRH cell based on a way of heuristic searching, which can closely approach the optimal solution. However, the resource allocation results of 3C-RA are highly effected by the pre-defined parameters including E_{ij}^T, T_{ij}^T, ξ, μ and S_{max}. 3C-RA uses converge parameters ξ and μ and maximal allowed steps S_{max} to make sure Algorithm 1, 2 and 3 converge after finite steps. In theory, the higher ξ, μ and S_{max} are, the longer time and more computing resource 3C-RA algorithm will cost to reach convergence and work out the resource allocation closer to the optimal solution, vice versa. Another thing is, how easily 3C-RA converges to the optimal solution is close related to the latency threshold T_{ij}^T and energy cost threshold E_{ij}^T, according to which the data rate and computation capacity requests of a UE task is calculated by solving (14) and (2). Basically, the lower E_{ij}^T or T_{ij}^T is, the harder 3C-RA will converge to the optimal solution, as lower E_{ij}^T or T_{ij}^T leading to higher data rate and computation capacity requirements of each UE task. Finally, 3C-RA may reach an sub-optimal solution without to the optimal one even running in incredibly long time, such as running with $\xi = 100\%$ and $\mu = 100\%$ and S_{max} as a very large number. This is because the optimization problem $\mathcal{P}$ is non-convex, and the heuristic search of 3C-RA may work in back and forth without reaching the optimal solution.

VI. SIMULATION RESULTS

To validate the contributions of this paper, we run a series of system simulations on Multi-layer CRAN with 3C-RA running in different configurations, then compare the UE tasks latency and UE throughput as the outputs. We also compare 3C-RA to the Proportional Fairness based Resource Allocation (PF-RA) solution proposed in [24], which carries out the resource allocation without cooperative control. The PF-RA is implemented as the default resource allocation solution in the Vienna LTE System Level Simulator [27] employed with related augments by this paper. The simulation is based on the Monte Carlo method, and is a time driven process. The simulation configurations, 3C-RA settings and UE scenarios are listed in Table.1. The 3C-RA setting 1 in Table.1 is to set 3C-RA running with harsh converge condition to allocate resource to UEs closer to the optimal solution. In contrast, 3C-RA setting 2 and 3 are to set 3C-RA running with medium and easy converge conditions.

The simulation results are shown in Fig.3, Fig.4 and Table.2. In Fig.3, the cumulative latency utility of each RRH cell, e.g $G_j(\alpha_j, \beta_j)$ of cell j as formulated in (9), is demonstrated. Compared to PF-RA, 3C-RA in Fig.3 universally enables higher cumulative latency utility of each RRH cell, considering all of the UE scenarios in Multi-layer CRAN. That means, with 3C-RA, Multi-layer CRAN better handles UEs tasks with low-latency. Considering different settings, 3C-RA in setting 1 gives the best performance on task latency, as it works with harsh converge condition. 3C-RA in setting 2 and setting 3 give the medium and worst performance on task latency.

TABLE 1. System configuration on simulations.

Parameter	Value
Bandwidth/RB numbers	20MHz / 100
RRH inter distance	250
RRH transmitter height	16m
RRH transmitter power	20w
UE transmitter height	1.5m
Number of RRHs	19
RRH antenna gain pattern	TS36.942[28]
RRH cells geometry	Regular hexagonal grid
RRH number of Antenna	1*1 SISO
Pathloss model	TS36.942: Urban [28]
Shadow fading	type Claussen [29]
UE task CPU cycles: F_{ij}	Voice: (1 to 5); Data: (5 to10)
UE task data request: D_{ij}	Voice: (1k to 5k); Data: (5k to10k)
E_j^{max} in (11)	5J/UE task
T_{ij}^{max} in (8)	Voice: (0.1s to 0.5s); Data: (0.5s to 1s)
$F_{j,LEC}^{max}$ and $f_{j,HEC}^{max}$	(25 to 50) and (50 to 100)
Simulation time TTI [27]	20
UE Scenario 1	5 UEs/RRH(Sparse)
UE Scenario 2	15 UEs/RRH (Medium)
UE Scenario 3	25 UEs/RRH (Congest)
3C-RA setting 1	$\xi = 0.8$, $\mu = 0.8$, $S_{max} = 20$, $T_{ij}^T = (1/2) * T_{ij}^{max}$, $E_{ij}^T = 5j$
3C-RA setting 2	$\xi = 0.6$, $\mu = 0.6$, $S_{max} = 15$, $T_{ij}^T = (2/3) * T_{ij}^{max}$, $E_{ij}^T = 5j$
3C-RA setting 3	$\xi = 0.5$, $\mu = 0.5$, $S_{max} = 10$, $T_{ij}^T = T_{ij}^{max}$, $E_{ij}^T = 5j$

TABLE 2. Performance comparisons on average RRH throughput.

UEs Scenario	Cumulative Latency Utility of The Whole Network /Average RRH Throughput(Mb/s)			
	PF-RA	3C-RA setting 3	3C-RA setting 2	3C-RA setting 1
No.1	311/25.50	322/29.43	363/32.17	371/36.65
No.2	680/26.88	713/31.57	790/33.81	836/38.00
No.3	985/27.05	1063/32.32	1138/34.17	1214/38.63

In Fig.4, the comparisons on average UE throughput according to Empirical Cumulative Distribution Function (ECDF) are shown. Compared to PF-RA, the ECDF outputs in Fig.4 show that 3C-RA universally enables higher UE throughput in all of the UE scenarios in Multi-layer CRAN. Similar to Fig.3, 3C-RA in setting 1 gives the best performance on UE throughput, and 3C-RA in setting 2 and setting 3 give the medium and worst performance on UE throughput.

In Table.2, the numerical comparisons on UE task latency and UE throughput are listed. According to Table.2, 3C-RA can maximally improve the cumulative latency utility of the whole network by 23.25% and the average RRH throughput by 42.81% compared to PF-RA in UE scenario 3, where

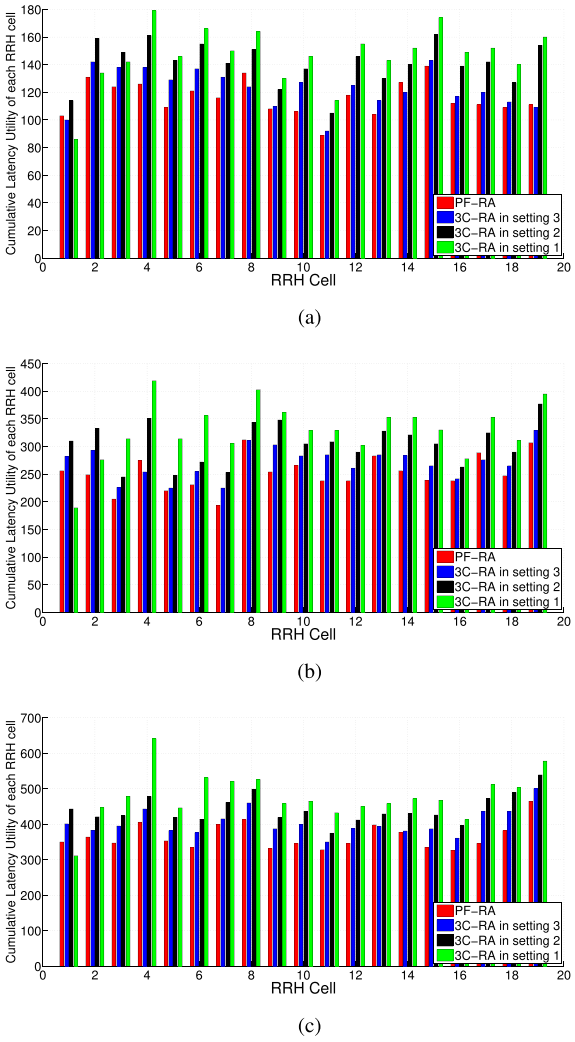

FIGURE 3. Comparisons on latency utilities. (a) UE scenario 1. (b) UE scenario 2. (c) UE scenario 3.

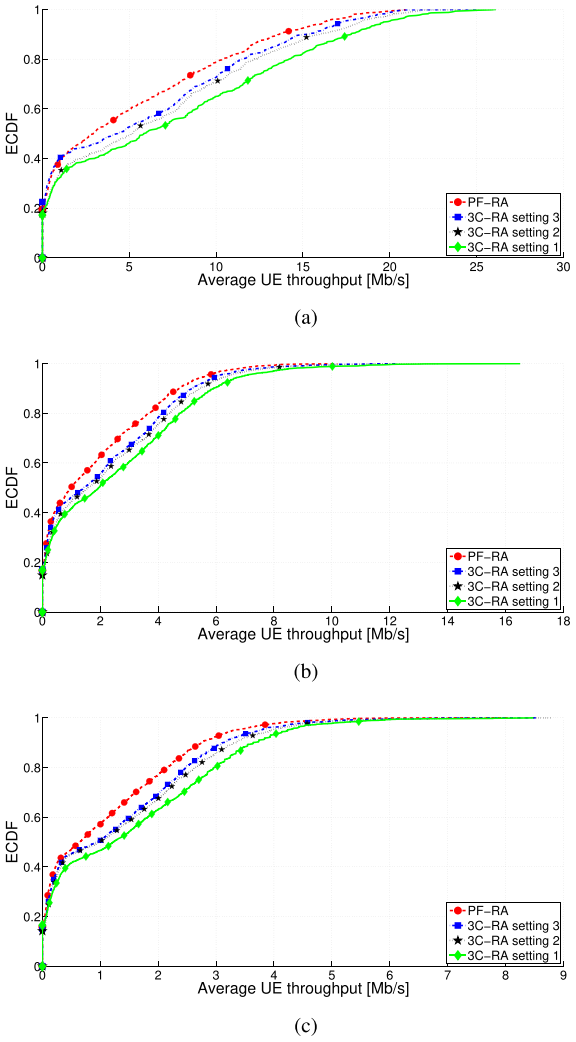

FIGURE 4. UE average throughput comparisons according to ECDF. (a) UE scenario 1. (b) UE scenario 2. (c) UE scenario 3.

Multi-layer CRAN is congested. In UE scenario 1 where UEs are sparse in Multi-layer CRAN, 3C-RA can increase the cumulative latency utility of the whole network by by 19.29% and the average RRH throughput by 43.73% compared to PF-RA. Table.2 also shows that 3C-RA improves the performance of Multi-layer CRAN in different settings.

In summary, based on the simulation results shown in Table.2 and Fig.3-4, it is obvious that Multi-layer CRAN as a new CRAN with Multi-layer MEC better utilizes local clouds to deliver low-latency computing and communication services. Further more, the 3C-RA algorithm as an effective resource allocation solutions enables Multi-layer CRAN to have UE tasks handled in lower latency and high network throughput according to the simulation setting and UE scenarios considered.

The 3C-RA overall complexity is $T(\sum_{\forall j \in C}(N_j * K * F_j^{max} * S_{max}))$. The specific computing time is determined by the number of RRH cells C, the number of RBs K, the number of CBs F_j^{max} and number of UEs N_j per cell j. As discussed in Algorithm 4, 3C-RA as a distributed algorithm can share the computation tasks within all the RRHs. Thus 3C-RA can run in efficiency and timely deliver the outputs.

VII. CONCLUSION

This paper proposes a new type of Mobile Edge Computing architecture named Multi-layer CRAN to provide dynamic computing offloading strategies. Multi-layer CRAN can provide low-latency computing and communication services through the helps of LECs as the proximity local clouds to UEs. This paper further designs a Cooperative Communication and Computation Resource Allocation (3C-RA) for Multi-layer CRAN. 3C-RA is designed to rationally allocate the communication and computation resource from RRHs, HEC and LECs to UEs in Multi-layer CRAN for low task latency. 3C-RA is designed to be distributed in real-time and to be scalable with respect to the network sizes. Through systematical simulations, the results validate that Multi-layer CRAN with 3C-RA has the theoretical performance gain.

In future work, we intend to improve the performance of Multi-layer CRAN by using fog computing to explore

the computing capabilities of UEs, deploying LECs based local clouds through better network planning, and implementing better resource allocation solutions etc. In current 3C-RA algorithm, the computation and communication resource allocations are in a loosely coupled format, where the communication resource allocation is sequentially built. Meanwhile, 3C-RA in this paper is based on a simple SISO antenna model, which does not take the cutting edge MIMO beam forming antenna model into consideration. This paper also does not consider any sophisticated UE to mobile edge cloud i.e. LEC and HEC selection method. Therefore, in future work, the 3C-RA algorithm for Multi-layer CRAN should be extended to have a more closely coupled cooperative communication and computation resource allocation and consider more advanced network scenarios.

REFERENCES

[1] "C-RAN the road towards green RAN," China Mobile Res. Inst., Tech. Rep. v2.5, 2011. [Online]. Available: http://labs.chinamobile.com/report/view_59826

[2] "C-RAN: The road towards green RAN," China Mobile Res. Inst., White Paper, Jun. 2014. [Online]. Available: http://labs.chinamobile.com/cran

[3] European Telecommunications Standards Institute. *Mobile Edge Computing: A Key Technology Towards 5G*. Accessed: Sep. 2015. [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf

[4] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[5] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, 2017.

[6] K. Zhang *et al.*, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[7] K. Wang, K. Yang, and C. Magurawalage, "Joint energy minimization and resource allocation in C-RAN with mobile cloud," *IEEE Trans. Cloud Comput.*, to be published.

[8] K. Wang, K. Yang, X. Wang, and C. S. Magurawalage, "Cost-effective resource allocation in C-RAN with mobile cloud," in *Proc. ICC*, May 2016, pp. 1–6.

[9] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, "Heterogeneous cloud radio access networks: A new perspective for enhancing spectral and energy efficiencies," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 126–135, Dec. 2014.

[10] H. Dahrouj, A. Douik, O. Dhifallah, T. Y. Al-Naffouri, and M.-S. Alouini, "Resource allocation in heterogeneous cloud radio access networks: Advances and challenges," *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 66–73, Jun. 2015.

[11] M. Gerasimenko *et al.*, "Cooperative radio resource management in heterogeneous cloud radio access networks," *IEEE Access*, vol. 3, pp. 397–406, 2015.

[12] X. He, A. He, Y. Chen, K. K. Chai, and T. Zhang, "Energy efficient resource allocation in heterogeneous cloud radio access networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.

[13] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, New York, NY, USA, 2010, pp. 49–62.

[14] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. 6th Conf. Comput. Syst. (EuroSys)*, New York, NY, USA, 2011, pp. 301–314.

[15] 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 *Proc. IEEE INFOCOM*, Mar. 2012, pp. 945–953.

[16] J. Oueis, E. C. Strinati, and S. Barbarossai, "Distributed mobile cloud computing: A multi-user clustering solution," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[17] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput. (MCC)*, 2012, pp. 13–16.

[18] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul./Aug. 2016.

[19] Y.-Y. Shih, W.-H. Chung, A.-C. Pang, T.-C. Chiu, and H.-Y. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE Netw.*, vol. 31, no. 1, pp. 52–58, Jan./Feb. 2017.

[20] S.-Y. Lien, S.-C. Hung, H. Hsu, and K.-C. Chen, "Collaborative radio access of heterogeneous cloud radio access networks and edge computing networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 193–199.

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

[22] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, 2010, p. 4.

[23] M. Peng, K. Zhang, J. Jiang, J. Wang, and W. Wang, "Energy-efficient resource assignment and power allocation in heterogeneous cloud radio access networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 11, pp. 5275–5287, Sep. 2014.

[24] Z. Sun, C. Yin, and G. Yue, "Reduced-complexity proportional fair scheduling for OFDMA systems," in *Proc. Int. Conf. Commun., Circuits Syst.*, Jun. 2006, pp. 1221–1225.

[25] P. Jiang, J. Bigham, and M. A. Khan, "Distributed algorithm for real time cooperative synthesis of wireless cell coverage patterns," *IEEE Commun. Lett.*, vol. 12, no. 9, pp. 702–704, Sep. 2008.

[26] H. Mei, J. Bigham, P. Jiang, and E. Bodanese, "Distributed dynamic frequency allocation in fractional frequency reused relay based cellular networks," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1327–1336, Apr. 2013.

[27] *Vienna LTE System Level Simulator v1.6*. Accessed: Dec. 2016. [Online]. Available: http://www.nt.tuwien.ac.at/ltesimulator/

[28] Technical Specification Group RAN, "E-UTRA; LTE RF system scenarios," 3GPP, Tech. Rep. TS 36.942, 2009.

[29] H. Claussen, "Efficient modeling of channel maps with correlated shadow fading in mobile radio systems," in *Proc. IEEE 16th Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2005, pp. 512–516.

HAIBO MEI received the B.Sc. and M.Sc. degrees from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, in 2005 and 2008, respectively, and the Ph.D. degree from the School of Electronic Engineering and Computer Science, Queen Mary University of London (QMUL), U.K., in 2012. He was a Post-Doctoral Research Assistant with QMUL and a Senior Research and Development Engineer with Securus Software Ltd., U.K. He is currently a Lecturer with the University of Electronic Science and Technology of China. His research interests include resource efficiency and self-organization of wireless communications, intelligent transportation system, and mobile cloud computing.

KEZHI WANG received the B.E. and M.E. degrees from the College of Automation, Chongqing University, China, in 2008 and 2011, respectively, and the Ph.D. degree from The University of Warwick, U.K., in 2015. He was a Senior Research Officer with the University of Essex, U.K. He is currently a Lecturer with the Department of Computer and Information Sciences, Northumbria University. His research interests include wireless communication, signal processing, and mobile cloud computing.

KUN YANG (SM'08) received the B.Sc. and M.Sc. degrees from the Computer Science Department, Jilin University, China, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, University College London (UCL), U.K. He is currently a Chair Professor with the School of Computer Science and Electronic Engineering, University of Essex, leading the Network Convergence Laboratory, U.K. Before joining in University of Essex at 2003, he was with UCL on several European Union (EU) research projects for several years. He manages research projects funded by various sources, such as UK EPSRC, EU FP7/H2020, and industries. He has authored over 80 journal papers. His main research interests include wireless networks, future Internet technologies, and mobile cloud computing. He is a fellow of the IET. He serves on the editorial boards of the IEEE and the non-IEEE journals.