On efficient offloading control in cloud radio access network with mobile edge computing

Tong Li[†¶‡], Chathura Sarathchandra Magurawalage[‡], Kezhi Wang[‡], Ke Xu[*†¶], Kun Yang[‖‡], Haiyang Wang[§]
[†]Tsinghua University, Beijing, China. Email: (litong12@mails, xuke@).tsinghua.edu.cn
[‡]University of Essex, Colchester, United Kingdom. Email: csarata, kezhi.wang, kunyang@essex.ac.uk
[§]University of Minnesota at Duluth, Minnesota, United States. Email: haiyang@d.umn.edu
[¶]Tsinghua National Laboratory for Information Science and Technology, Beijing, China.
[‖]ISN National Key Lab, Xidian University, Xian, Shanxi, China.

***Abstract*—Cloud radio access network (C-RAN) and mobile edge computing (MEC) have emerged as promising candidates for the next generation access network techniques. Unfortunately, although MEC tries to utilize the highly distributed computing resources in close proximity to user equipments equipments (UE), IN C-RANS suggests to centralize the baseband processing units (BBU) deployed in radio access networks. To better understand and address such a conflict, this paper closely investigates the MEC task offloading control in C-RANs environments. In particular, we focus on perspective of matching problem. Our model smartly captures the unique features in both MEC and C-RAN with respect to communication and computation efficiency constraints. We divide the cross-layer optimization into the following three stages: (1) matching between remote radio heads (RRH) and UEs, (2) matching between BBUs and UEs, and (3) matching between mobile clones (MC) and UEs. By applying the Gale-Shapley Matching Theory in the duplex matching framework, we propose a multi-stage heuristic to minimize the refusal rate for user's task offloading requests. Trace-based simulation confirms that our solution can successfully achieve near-optimal performance in such a hybrid deployment.**

***Index Terms*—Computation Offloading, Cloud Radio Access Network, Mobile Edge Computing, Offloading Control**

I. Introduction

User equipment (UE) (*e.g.*, smartphone, tablet, wearable device, and digital camera) is playing an important role in new application scenarios including virtual reality (VR), augmented reality (AR) and surveillance system, *etc*. While resource-constrained UEs (CPU, GPU, memory, storage capacity, and battery lifetime) have driven a dramatic surge in developing new paradigms to handle computation intensive tasks [1] (for example, computation intensive applications requiring huge computing capacity are not suitable to run in mobile or portable devices). Mobile cloud computing (MCC) [2] provides a solution where UEs offload computation to the remote resourceful cloud (*e.g.*, EC2 [3]), thereby saving processing power and energy. However, the cloud in MCC scenarios is usually in a wide area network (WAN), and it is difficult to control delays and jitters at the WAN scale. Therefore, offloading tasks to the public cloud may suffer from high latency via the Internet [4]. For example, AR requires low latency in order to provide correct information according to user location and orientation, while offloading tasks to remote cloud may incur information distortion due to delayed data transmission. To accomplish this, mobile edge computing (MEC) [5] is proposed where UEs offload computation intensive tasks to a computing resource-rich location, within radio access networks and in close proximity to UEs.

On the other hand, task offloading generates data intensive workloads, which may become one of the main influential factors of the unprecedented mobile traffic growth. It has been predicted that mobile traffic will increase exponentially to 100 times by the year 2020 [6] [7]. The dynamics of substantially increased data rates requires that cellular infrastructure must be flexible and reconfigurable, supporting simplified deployment and management. As conventional radio access network may incur high cost, high latency and data exchange inefficiency [8], it lacks the efficiency to support centralized interference management and the flexibility to migrate services to the edge for computation intensive applications.

To ensure highly efficient network operation and flexible service delivery when handling mobile Internet traffic surging, cloud radio access network (C-RAN) [8] brings cloud computing technologies into mobile networks by centralizing baseband processing units (BBU) of radio access network. It moves BBU from traditional base stations to the cloud and leaves remote radio heads (RRH) distributed geographically. RRHs are connected to the BBU pool via high bandwidth and low-latency fronthaul. The BBU pool can be realized by virtual machines (VM) in data centers, and the centralized baseband processing enables BBU to be dynamically configured and shared on demand [9]. In this case, with the transition from a conventional hardware based environment to a software based infrastructure, C-RAN can achieve flexible matching between RRHs and BBUs, according to the quality of service (QoS) requirement.

It is worth mentioning that C-RAN uses centralized BBU to do baseband processing, while MEC handles distributed task offloading by shifting computation capacity from a public cloud to an edge cloud, which can significantly reduce offloading latency. Since MEC usually works with distributed base stations in conventional radio access network, it is quite interesting to see if MEC mobile offloading still works in C-RANs environments. Figure 1 shows the hybrid deployment of C-RAN with MEC for computation offloading. Connected with geographically distributed RRHs and centralized BBUs, UEs get access to VMs, called mobile clones (MC), in a mobile

*Ke Xu is the corresponding author (Email: xuke@tsinghua.edu.cn).

cloud for computation offloading. For computation offloading requests, data is first transmitted by base stations (RRHs and BBUs) via uplinks. Once processed by an MC in a mobile cloud, the results will be returned to UEs via downlinks. As this paper mainly focuses on uplink optimization, we calculate the completion time of task offloading as the sum of the data transmission latency via wireless communication and the task processing time on MCs.

Assume RRHs, BBUs and MCs are heterogeneous (*e.g.*, different loads and amount of resources), then the different matching among UEs, RRHs, BBUs and MCs results in different task offloading efficiencies. In particular, data transmission latency depends on the assignment of both RRHs and BBUs, and task processing time depends on the MC assignment. However, the UE interaction makes it challenging to directly assign a UE's most satisfied RRH, BBU or MC to them. This interaction may affect the task offloading efficiency in two aspects: (1) the wireless transmission data rate will decrease with poor channel qualities between UEs and RRHs, (2) while the baseband processing speed of BBUs and task processing speed of MCs will be slowed down when overloaded. The former is called *communication efficiency*, and the latter is called *computation efficiency*.

For offloading control, we define *refusal ratio* as the proportion of offloading tasks that are not able to meet their deadlines. Then this paper is devoted to the efficient offloading control by addressing the assignment problem: how to assign RRHs, BBUs and MCs to UEs to minimize the refusal ratio among all the offloading requests? Different from the prior solutions of resource allocation [10], [11], [12], [13] and admission control [14] [15], we focus on the matching problem. Moreover, we take into account the task offloading efficiency not only in wireless transmission but also in cloud computing, which is new and challenging in achieving efficient MEC task offloading control in C-RANs environments.

Motivated by these observations, we first formulate the joint assignment among UEs, RRHs, BBUs and MCs, which is unfortunately NP-Hard. By applying the duplex matching framework based on the classic Gale-Shapley Matching Theory, a multi-stage heuristic is finally given to minimize the refusal rate for UE's task offloading requests. Our major contributions are summarized as follows. 1) We handle the offloading control with a new perspective that focuses on the joint RRH, BBU and MC matching problem, where a 0-1 programming model capturing the unique features in both MEC and C-RAN is proposed (Section IV). 2) We divide the optimization problem into three stages including the UE-to-RRH stage, the UE-to-BBU stage, and the UE-to-MC stage, and a multi-stage heuristic for efficient offloading control is proposed (Section V). 3) We conduct a trace-based evaluation to show that our solution can achieve near-optimal performance for MEC task offloading control in C-RANs environments (Section VI).

II. Related Work

Cai *et al.* [16] enabled cloud services in the Internet, serving UEs by using a split-TCP proxy. However, the Internet may introduce large latency to the transmission, which may not be able to complete tasks within the required time limits. Wang *et al.* [10], [11], [12] studied the joint resource allocation in C-RANs with MCC under the time constraints of the given tasks. Also, Sardellitti *et al.* [13] studied joint optimization of radio and computational resources for MEC combined with cellular networks. Tang *et al.* [9] studied the cross-layer resource allocation with elastic service scaling in C-RANs. Nevertheless, all the above work fell in the general category of resource allocation optimization, without considering the optimal matching between users (*e.g.*, UE), communication resource (*e.g.*, BBU) and computing resource (*e.g.*, MC).

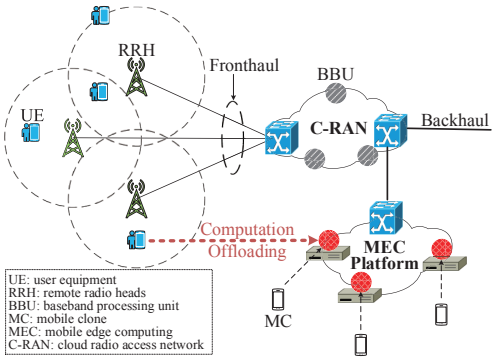

Fig. 1: Computation offloading architecture

Moreover, Ha *et al.* [14] proposed cooperative transmission in C-RANs considering cloud processing constraints by allocating different BBUs and RRHs to different UEs. Ha [15] moved a step further by considering admission control in C-RANs under the fronthaul constraints. However, those two papers [14] [15] only consider communication efficiency, other than considering cloud service computation efficiency as well. Thus, to address the above challenges, we focus on the perspective of multi-stage RRH, BBU and MC assignments, and design a duplex matching framework based on the classic Gale-Shapley Matching Theory.

III. Offloading Control: Background and Framework

This section clarifies the computation offloading background and the offloading control framework in C-RANs with MEC.

A. Computation Offloading

Figure 1 illustrates the overall architecture for task offloading in C-RANs with MEC. There are three basic components in the architecture: (1) Geographically distributed, RRHs are remote radio transceivers that bridge UEs and the operator radio control panel, performing lower layer analogue radio frequency (RF) functions. (2) Centralized in C-RANs, BBU is a unit for digital signal processing which can dynamically provision baseband processing for multiple distributed RRHs on demand. (3) MC is a VM deployed in a mobile cloud near the BBU pool, hosting various mobile edge applications (*e.g.*, edge health care, smart tracking). For the scenarios of C-RANs with MEC, the MEC platform hosts computation and services at the edge of radio access networks, reducing network latency and bandwidth consumption for subscribers. Furthermore, network operators allow third-party partners to run the MEC

platform, which will promote the rapid deployment of new applications and edge services to the mobile subscribers.

B. Computation and Communication Efficiency

Here we argue that not only communication efficiency but also computation efficiency should be considered in C-RANs with MEC scenarios, *i.e.*, there is interference among UEs both in wireless data transmission and cloud task processing. It is easy to understand that wireless channel quality will be influenced by user interaction. On the other hand, multiple tasks will compete for CPU time slices, which may lead to queueing delay. Moreover, based on the fact that the BBU processing during wireless communication can be regarded as computation intensive workload [8], multiple UEs will also compete for the computing resource in the BBU.

Computation Efficiency. With regard to task processing, we use q_v to denote the number of tasks (load) being processed in MC v. To capture the relationship between processing speed and task load, we introduce the Net Present Value (NPV) function [17], which is proved to fit the measurement results by Wang *et al.* [18]. We calculate the task processing speed as follow:

$$f_{GOPS}^{v} = \frac{\beta^{\gamma - q_v}}{\alpha} \tag{1}$$

where f_{GOPS}^{v} refers to the computation frequency (CPU cycles per second) with the unit of *giga operations per second* (GOPS) in MC v. The parameter α indicates the speed when MC is fully loaded (reaching the VM service limitation γ ($\gamma > \max\{\lfloor \frac{n}{k} \rfloor, \lfloor \frac{n}{m} \rfloor\}$). The service limitation depends on the resource allocated to the VM, reflecting the budget of network operators. The parameter β controls the skewness of the relationship between load and speed where $\beta \in (1, +\infty)$. It is easy to see that different VMs may have different α, β and γ. For example in [18], the features of the EC2 instances is captured as follows: α is around 105, β is around 1.04, and γ represents the resource amount purchased from EC2.

Communication Efficiency. On the other hand, the communication efficiency is influenced by multiple factors including radio signal bandwidth and the modulation and coding scheme (MCS) index. Alyafawi *et al.* [19] conducted a research to show that the decoding and encoding time for the LTE subframes grows with the increase of the MCS index. It is revealed that effective data rate over the air interface (throughput) is mainly controlled by MCS. For heterogeneous UEs and RRHs in C-RANs, the MCS index varies from 0 to 31, deciding the number of bits per symbol and defining the amount of redundant information inserted into data stream [20]. Hence, the communication efficiency mainly depends on the MCS index between RRHs and UEs (in this paper, we do not consider user interference in wireless channels, as it can also be reflected by the MCS index). Based on the prior related work [21] [22], we define the base station communication efficiency with the unit of giga operations per bit (GOPB). We use $f_{GOPB} = g(\theta_{ul})$ to denote the communication efficiency, where $g(\theta)$ is defined as a function of the MCS index.

Therefore, we derive the wireless transmission data rate (bit per second) as follow:

$$\rho_u = \frac{f_{GOPS}^{b}}{g(\theta_{ul})} \tag{2}$$

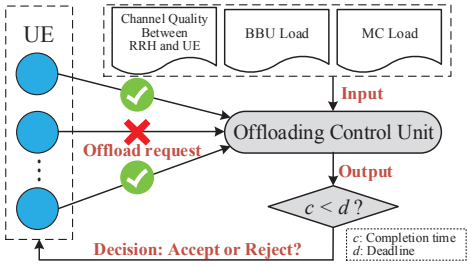

Fig. 2: Offloading control framework

where f_{GOPS}^{b} refers to the computation frequency with the unit of GOPS in BBU b.

C. Offloading Control Framework

We illustrate the deadline-aware offloading control framework in Fig. 2. In terms of heterogeneous RRHs, BBUs and MCs, we consider the channel qualities between RRHs and UEs, the BBU load and the MC load as the inputs. At first, UE generates tasks with offloading requests, then the offloading control unit (*e.g.*, the mobile cloud controller) assigns RRHs, BBUs and MCs to each UE. The expected completion time of each offloading task is obtained as the output. By estimating whether a task may exceed its deadline, we decide to accept or reject UE's offloading request. Note that our objective is to maximize the number of tasks meeting their deadlines, the operator may gain a better profit while satisfying most subscribers.

IV. Problem Formulation

In this section, we formulate the matching problem among UEs, RRHs, BBUs and MCs to achieve the optimal offloading control in C-RANs with MEC. $\mathcal{U} = \{u_1, u_2, ..., u_n\}$, $\mathcal{L} = \{l_1, l_2, ..., l_o\}$, $\mathcal{B} = \{b_1, b_2, ..., b_k\}$ and $\mathcal{V} = \{v_1, v_2, ..., v_m\}$ denote the sets of UEs, RRHs, BBUs and MCs, respectively. n, o, k, and m denote the number of UEs, RRHs, BBUs and MCs, respectively. For a UE $u \in \mathcal{U}$ that requests task offloading, d_u refers to the deadline, and c_u refers to the completion time. According to Section III-B, we consider the constraints of computation and communication efficiency. We model the task processing time and the wireless transmission latency, and then model the assignment optimization problem.

A. Task Processing Time

As mentioned above, computation efficiency depends on the loads in MCs. According to Equation (1), we therefore obtain the processing time of UE's offloading tasks as follow:

$$T_C(u, v) = \frac{F_u}{f_{GOPS}^{v}} = \frac{\alpha F_u}{\beta^{\gamma - q_v}} \tag{3}$$

where F_u refers to the computing resource of the offloading task, which is denoted by the number of CPU operations.

B. Wireless Transmission Latency

In C-RANs, user data is transmitted by wireless communication via base stations, in which the fibre links between RRHs and BBUs allow more flexibility in network planning and deployment. On the other hand, the BBU pool is also a cloud-based platform in C-RANs. Thus, the wireless transmission latency is related to both communication efficiency and

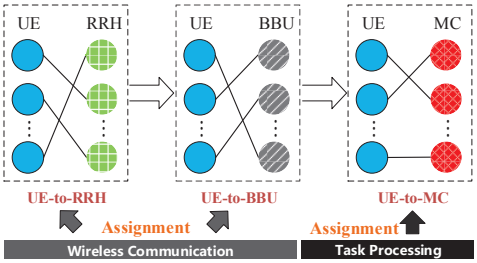

Fig. 3: Joint assignment among UEs, RRHs, BBUs and MCs

computation efficiency. As mentioned above, we use different MCS indexes to estimate the communication efficiency. For the BBU baseband computation efficiency, we again use the NPV function to capture the relationship between baseband processing speed and the BBU load, *i.e.*, $f^b_{GOPS} = \frac{\beta^{\gamma - q_b}}{\alpha}$. Then based on Equation (2), we obtain the wireless transmission latency for UE u as follow:

$$T_N(u,l,b) = \frac{D_u}{\rho_u} = \frac{\alpha \cdot D_u \cdot g(\theta_{ul})}{\beta^{\gamma - q_b}} \tag{4}$$

where D_u refers to the traffic size to be transmitted to the cloud for UE u.

C. Joint Assignment Optimization

Figure 3 illustrates the RRH, BBU and MC assignments. We define x_{uv}, z_{ul}, y_{ub} as the decision variables. In particular, $x_{uv}, z_{ul}, y_{ub} = 1$ if MC v, RRH l and BBU b are assigned to UE u, respectively, otherwise $x_{uv}, z_{ul}, y_{ub} = 0$. Since the offloading scheme depends on whether the task is able to meet its deadline, our objective becomes minimizing the refusal ratio for the UE's offloading requests, *i.e.*, maximizing the amount of UEs whose completion time is less than their deadlines. As mentioned before, we focus on the uplink completion time, which can be calculated by

$$c_u = T_C(u, v(u)) + T_N(u, l(u), b(u)) \tag{5}$$

where $v(u)$, $l(u)$, and $b(u)$ denote the MC, RRH and BBU assigned to UE u, respectively. Defining $\{x\}^+ = \begin{cases} 1, & x > 0 \\ 0, & \text{otherwise} \end{cases}$, we therefore obtain the number of UEs that will miss their deadlines, *i.e.*, $Z = \sum_{u \in \mathcal{U}} \{c_u - d_u\}^+$ (refusal ratio is $\frac{Z}{n}$). Then the joint assignment optimization model is proposed as follows:

$$\min \quad \sum_{u \in \mathcal{U}} \{c_u - d_u\}^+ \tag{6}$$

$$\text{s.t.} \quad \sum_{v \in \mathcal{V}} x_{uv}, \sum_{b \in \mathcal{B}} y_{ub}, \sum_{l \in \mathcal{L}} z_{ul} = 1 \quad \forall u \in \mathcal{U} \tag{7}$$

$$\sum_{u \in \mathcal{U}} x_{uv} \le \gamma_v - \gamma^0_v, \quad \forall v \in \mathcal{V} \tag{8}$$

$$\sum_{u \in \mathcal{U}} y_{ub} \le \gamma_b - \gamma^0_b, \quad \forall b \in \mathcal{B} \tag{9}$$

$$x_{uv}, y_{ub}, z_{ul} = 0 \text{ or } 1, \quad \forall u \in \mathcal{U}, v \in \mathcal{V}, b \in \mathcal{B} \tag{10}$$

where the constraint (7) refers to that every UE only selects one MC, every UE only selects one BBU, and every UE only selects one RRH. Note that the load of MC v can be calculated as $q_v = \sum_{u \in \mathcal{U}} x_{uv} + \gamma^0_v$, and the load of BBU b can be calculated as $q_b = \sum_{u \in \mathcal{U}} y_{ub} + \gamma^0_b$, where γ^0 denotes the initial load. Then (8) and (9) refer to the service limitation constraints of MC and BBU, respectively.

According to Formulas (3)-(6) and (10), the joint assignment optimization is a non-linear integer programming problem, which essentially turns out to be NP-Hard [23] [24]. Thus we are devoted to seeking efficient heuristics towards the optimal solution, which will be detailed in the next section.

V. Multi-stage Duplex Matching

By exhaustively searching all the possible combination of x_{uv}, y_{ub} and z_{ul}, the optimal solution can be achieved. However, the practical usefulness of this method is limited considering the real-time user demands. We thus propose a tri-level heuristic, which divides the optimization problem into three stages: the UE-to-RRH stage, the UE-to-BBU stage, and the UE-to-MC stage. We define the assignment A_1, A_2 and A_3 as the matchings from $\mathcal{U}$ to $\mathcal{L}$, $\mathcal{U}$ to $\mathcal{B}$ and $\mathcal{U}$ to $\mathcal{V}$, respectively. A_1 is the optimal assignment in the UE-to-RRH stage, while A_2 and A_3 are heuristic solutions obtained by applying the Matching Theory. Note that these three stages are correlative during multi-stage matching. In particular, A_2 is obtained according to A_1, and A_3 is obtained according to A_1 and A_2.

A. UE-to-RRH Stage

We assume that all the BBUs are the same and fully loaded, *i.e.*, $\alpha_b = \alpha$ and $q_b = r_b$ ($b \in \mathcal{B}$). According to Equation (2), the expected transmission latency of UE u becomes $\alpha \cdot D_u \cdot g(\theta_{ul})$. In this case, each UE just selects the RRH with the minimal expected transmission latency. Thus, based on the MCS index θ_{ul}, we can get the optimal assignment A_1 between UEs and RRHs.

B. UE-to-BBU Stage

In this stage, different UEs have different deadlines d. Meanwhile, different BBUs have different loads q and service limitations γ. The BBU assignment problem to minimize the transmission latency can be transformed into a 0-1 Multiple Knapsack problem with the non-linear objective function, which is known to be NP-Hard [23]. Since it is hard to get the optimal assignment here, we design a heuristic duplex matching algorithm based on the Gale-Shapley Matching Theory [25], in which Gale *et al.* discussed the real-life college admission problem (CAP). Similarly, we can regard UEs as students and BBUs as colleges. Since both UEs and BBUs own diverse properties and their preference lists are variable within a large range. To apply the Gale-Shapley Matching Theory, the challenge here is how to define the preference lists reasonably and efficiently [26].

BBU Availability. We assume that all the MCs are the same and fully loaded, *i.e.*, $\alpha_v = \alpha$ and $q_v = r_v$ ($v \in \mathcal{V}$). According to Equation (3), the expected processing time of UE becomes αF_u. Since the BBU load affects the transmission latency according to Equation (2), when we add a UE to the BBU, the performance of the UEs that are already assigned to this BBU will be affected. We therefore define that a BBU is available

to a UE if those existing UEs can still meet their deadlines after the new UE is added in, *i.e.*, $T_N(u(b)) \leq d_{u(b)} - \alpha F_{u(b)}$, where $u(b)$ denotes the UE assigned to BBU b.

Algorithm 1 PreferenceListGeneration()

Get θ_{ul} according A_1;
for all u that $u \in \mathcal{U}$ **do**
  Get the set of the available BBU set $\mathcal{B}_u^*$;
end for;
for all u b that $u \in \mathcal{U}$, $b \in \mathcal{B}$ **do**
  $T_N(u, l, b) = GetExpectedTransLatency(q_u, \theta_{ul})$;
  Get $\mathcal{P}_u$ by sorting $\mathcal{B}_u^*$ by ascending order of $T_N(u, l, b)$;
  Get $\mathcal{Q}_b$ by sorting $\mathcal{U}$ by ascending order of $d_u - \alpha F_u$;
end for;
return $\mathcal{Q}$ and $\mathcal{P}$;

Preference List Generation. The preference for UE u to select the available BBUs results in preference list $\mathcal{P}_u$ ($\mathcal{P} \subseteq \mathcal{B}$). Also, every BBU owns a preference list $\mathcal{Q}_b$ ($\mathcal{Q} \subseteq \mathcal{U}$). As depicted in Algorithm 1, we calculate $\mathcal{P}_u$ and $\mathcal{Q}_b$ as follows. 1) For UE u to select a BBU, $\mathcal{P}_u$ is obtained by sorting BBU set $\mathcal{B}$ by an ascending order of the expected transmission latency. To calculate the expected transmission latency for UE u assigning BBU b (Algorithm 1. *Step 6*), we add 1 to q_b and get θ_{ul} based on the assignment A_1 before calculating T_N according to Equation (2). 2) Similarly, for BBU b to select a UE, $\mathcal{Q}_b$ is obtained by sorting $\mathcal{U}$ by an ascending order of $d_u - \alpha F_u$, where d_u denotes the deadline of UE u.

Algorithm 2 DuplexMatchingAlgorithm()

$i \leftarrow 1$, $\varphi_b \leftarrow 1$, $\mathcal{Y}_b \leftarrow \emptyset (b \in \mathcal{B})$;
while $(\bigcup_{b \in \mathcal{B}} \mathcal{Y}_b \neq \mathcal{U})$ **do**
  for all u b that $u \in (\mathcal{U} \setminus \cup_{b \in \mathcal{B}} \mathcal{Y}_b)$, $b \in \mathcal{B}$ **do**
    $(\mathcal{P}_u, \mathcal{Q}_b) = PreferenceListGeneration()$;
    $\mathcal{Y}_b = \mathcal{Y}_b \bigcup \{u | GetTopItem(\mathcal{P}_u) = b\}$;
  end for;
  for all b that $b \in \mathcal{B}$ **do**
    Sort $\mathcal{Y}_b$ according to UE ranking in $\mathcal{Q}_b$;
    if $GetElementCount(\mathcal{Y}_b) > \varphi_b$ **then**
      Define a temporary set $\mathcal{H}$ as the set of the bottom $(GetElementCount(\mathcal{Y}_b) - \varphi_b)$ UE(s) in $\mathcal{Y}_b$;
      $\mathcal{Y}_b = \mathcal{Y}_b \setminus \mathcal{H}$;
      $\varphi_b = \varphi_b + \triangle\varphi$;
    end if;
  end for;
  $i = i + 1$;
end while;
return A_2 according to $\mathcal{Y}_b$;

UE-to-BBU Duplex Matching Algorithm. Defining $\mathcal{Y}_b$ as the prospective admission list of BBU b, and φ as the quota of a BBU. Regarding to the CAP, students are considered by a college which can admit a quota of only φ. Similarly, we set a dynamic quota for each BBU, which gradually increases during iterations. The duplex matching framework can be described as Algorithm 2.

C. UE-to-MC Stage

After wireless transmission through base station, the offloading tasks are processed in the mobile cloud. The final UE-to-MC stage handles the MC assignment for offloading requests. Similar to the UE-to-BBU stage, it is also feasible to apply the duplex matching framework in this stage, however, to achieve better performance we have to modify some steps.

Firstly, the preference list for UE u to select the available MCs is denoted by $\mathcal{P}'_u$ ($\mathcal{P}' \subseteq \mathcal{V}$), which can be obtained by sorting $\mathcal{V}$ in an ascending order of the expected processing time. To calculate the expected processing time for UE u assigned MC v, we add 1 to q_v, then calculate T_C according to Equation (3). Different from the UE-to-BBU stage, for MC v to select UEs, the preference list $\mathcal{Q}'_v$ ($\mathcal{Q}' \subseteq \mathcal{U}$) is obtained by sorting $\mathcal{U}$ in an ascending order of $d_u - T_N$, where T_N is calculated based on the assignment A_2. Secondly, the available MCs for a UE must meet the constraints of $T_C \leq d_u - T_N$ for all the UEs assigned to this MC. Note that during the UE-to-BBU stage, some of the offloading requests were probably not assigned to any BBU, due to task deadline constraints and BBU service limitations. As a result, in this stage we no longer assign MCs to the UEs that were not assigned a BBU.

By applying the modified Algorithms 1 and 2, we obtain the assignment A_3 through the duplex matching framework. The matchings between RRHs and BBUs, and BBUs and MCs are also obtained by combining A_1, A_2 and A_3.

VI. Performance Evaluation

In this section, we conduct the matlab-based implementation to estimate the duplex matching framework, where trace-based offloading requests are fed to these programs. In particular, task deadline, offloading traffic size and computation resource demand are randomly generated in a uniform distribution according to the prior works [21] [27] [28], *i.e.*, we set $d_u \in [4000, 6000]$ (ms), $D_u \in [1, 100]$ (MB), $F_u \in [1, 20]$ (G Hz). We also set parameters $\alpha = 102$, $\beta = 1.0026$, $\gamma = 400$, $\gamma^0 = 0$, $\Delta\varphi = 1$, and empirically set $g(\theta) = \frac{0.0156}{\theta}$.

For optimality comparison, we summarize the algorithms as follows. *Optimal baseline* refers to the optimal solution obtained by brute-force searching. *Duplex matching* refers to our heuristic solution proposed in Section V. *Linear programming relaxation* refers to the solution that converts the integer constraint (Formula (10)) into the continuous one, *i.e.*, $x_{uv}, y_{ub}, z_{ul} \in [0, 1], \forall u \in \mathcal{U}, v \in \mathcal{V}, b \in \mathcal{B}$. By solving the relaxed linear programming, we obtain the rounded decision variables as a feasible solution.

We start with a small offloading scenario with 6 offloading requests (each request is generated by an individual UE), 5 RRHs distributed geologically, 3 BBUs in the BBU pool, and 3 MCs in the MEC platform, *i.e.*, $n = 6$, $o = 3$, $k = 3$, and $m = 3$. Figure 4 presents the performance of our optimal assignment among UEs, RRHs, BBUs and MCs (we test 10 randomly generated cases). We can see that the multi-stage duplex matching reduces the refusal ratio compared to the linear programming relaxation solution (whose refusal ratio can be as high as 68%). Note that both the optimal baseline and our solution can achieve 0 refusal ratio in Case 9. Eventually, this figure draws the conclusion that our approach can achieve near-optimal performance in 90% of cases.

To avoid measurement bias, we also test the scenarios with a larger number of UEs, RRHs, BBUs and MCs, *i.e.*, $n \geq 100$, $o = 50$, $k = 30$, and $m = 30$. Figure 5 illustrates the case when the number of UEs varies from 100 to 500 (refusal ratio is the average value calculated by running the same case 100 times). We can see that even though refusal ratio increases with the number of UEs, our solution can always bound the optimal assignment. Figure 6 further explores the cumulative distribution function (CDF) of task completion time for the UEs admitted (UEs that can meet their deadline). It is easy to

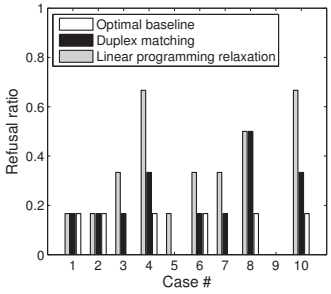

Fig. 4: Refusal ratio

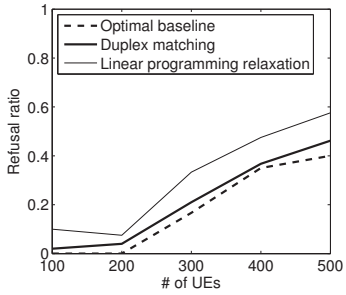

Fig. 5: Refusal ratio vs. # of UEs

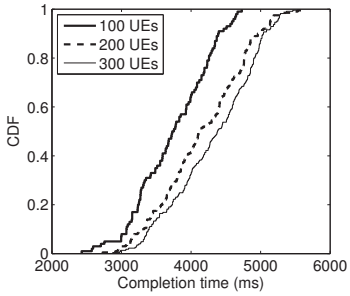

Fig. 6: Completion time

see that completion time increases with the growing number of UEs, due to the increased load on both BBUs and MCs, which affects the computation efficiency.

VII. Conclusion

This paper focuses on the perspective of matching problem in the hybrid offloading architecture of C-RANs with MEC. We design an efficient offloading control framework minimizing the refusal ratio of offloading requests. The joint assignment modeling shows the NP-Hardness of our problem, and a tri-level heuristic is therefore proposed, which divides the cross-layer optimization into three stages: the UE-to-RRH stage, the UE-to-BBU stage, and the UE-to-MC stage. By applying the Matching Theory, we propose the duplex matching framework. Evaluation shows that our solution can achieve near-optimal performance.

Future work includes the extension to environments with multi-resource management as well as further research on scenarios of mobility management [29].

Acknowledgment

This work was done during Tong Li's visit in University of Essex as a sponsored researcher and was supported by National Natural Foundation of China (61472212), National Science and Technology Major Project of China (2015ZX03003004), National High Technology Research and Development Program of China (863 Program) (2013AA013302, 2015AA015601), EU Marie Curie Actions CROWN (FP7-PEOPLE-2013-IRSES-610524), UK EPSRC Project NIRVANA (EP/L026031/1), EU H2020 Project iCIRRUS (GA-644526), and Natural Science Foundation of China (61620106011). Besides, grateful acknowledgement is made to Dr. Nikolaos Thomos who gave me considerable help by means of suggestions and comments.

References

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

[2] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wirel. Commun. Mob. Comput.*, vol. 13, no. 18, pp. 1587–1611, 2013.

[3] Amazon EC2. [Online]. Available: http://aws.amazon.com/ec2/

[4] F. Safaei, P. Boustead, C. Nguyen, J. Brun, and M. Dowlatshahi, "Latency-driven distribution: infrastructure needs of participatory entertainment applications," *Communications Magazine*, vol. 43, no. 5, pp. 106–112, 2005.

[5] Y. C. Hu, M. Patel, and D. Sabella, "Mobile edge computingła key technology towards 5g," *ETSI White Paper*, vol. 11, 2015.

[6] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, and A. C. Soong, "What will 5g be?" *JSAC*, vol. 32, no. 6, pp. 1065–1082, 2014.

[7] Y. Zhang, K. Xu, G. Yao, M. Zhang, and X. Nie, "Piebridge: A cross-dr scale large data transmission scheduling system," in *Proc. of SIGCOMM*. ACM, 2016, pp. 553–554.

[8] C. Mobile, "C-ran: the road towards green ran," *White Paper*, 2011.

[9] J. Tang, W. P. Tay, and T. Q. Quek, "Cross-layer resource allocation with elastic service scaling in cloud radio access network," *TWC*, vol. 14, no. 9, pp. 5068–5081, 2015.

[10] X. Wang, K. Wang, S. Wu, S. Di, K. Yang, and J. H, "Dynamic resource scheduling in cloud radio access network with mobile cloud computing," in *Proc. of IWQoS*. IEEE, 2016, pp. 1–6.

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

[12] K. Wang, K. Yang, and C. Magurawalage, "Joint energy minimization and resource allocation in c-ran with mobile cloud," *ToCC*, no. 99, pp. 1–1, 2016.

[13] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *TSIPN*, vol. 1, no. 2, pp. 89–103, 2015.

[14] V. N. Ha, L. B. Le *et al.*, "Cooperative transmission in cloud ran considering fronthaul capacity and cloud processing constraints," in *WCNC*. IEEE, 2014, pp. 1862–1867.

[15] V. N. Ha and L. B. Le, "Joint coordinated beamforming and admission control for fronthaul constrained cloud-rans," in *Proc. of GLOBECOM*. IEEE, 2014, pp. 4054–4059.

[16] Y. Cai, F. R. Yu, and S. Bu, "Cloud radio access networks (c-ran) in mobile cloud computing systems," in *Proc. of INFOCOM Workshops*. IEEE, 2014, pp. 369–374.

[17] S. A. Ross, "Uses, abuses, and alternatives to the net-present-value rule," *Financial management*, vol. 24, no. 3, pp. 96–102, 1995.

[18] H. Wang, R. Shea, X. Ma, F. Wang, and J. Liu, "On design and performance of cloud-based distributed interactive applications," in *Proc. of ICNP*. IEEE, 2014, pp. 37–46.

[19] I. Alyafawi, E. Schiller, T. Braun, D. Dimitrova, A. Gomes, and N. Nikaein, "Critical issues of centralized and cloudified lte-fdd radio access networks," in *Proc. of ICC*. IEEE, 2015, pp. 5523–5528.

[20] Modulation and Coding Scheme (MCS) Index. [Online]. Available: http://mcsindex.com/

[21] T. Sigwele, A. S. Alam, P. Pillai, and Y. F. Hu, "Evaluating energy-efficient cloud radio access networks for 5g," in *Proc. of DSDIS*. IEEE, 2015, pp. 362–367.

[22] J. Chen, T. Li, R. Du, and J. Fu, "Efficient reliable opportunistic network coding based on hybrid flow in wireless network," *China Communications*, vol. 8, no. 4, pp. 125–131, 2011.

[23] P. Brucker and S. Knust, "Complexity results for scheduling problems," http://www2.informatik.uni-osnabrueck.de/knust/class/, 2009.

[24] K. Xu, T. Li, H. Wang, and H. Li, "Modeling, analysis, and implementation of universal acceleration platform across online video sharing sites," *TSC*, pp. 1–15, 2016.

[25] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *American Mathematical Monthly*, pp. 9–15, 1962.

[26] T. Li, K. Xu, M. Shen, H. Wang, K. Yang, and Y. Zhang, "Towards minimal tardiness of data-intensive applications in heterogeneous networks," in *Proc. of ICCCN*. IEEE, 2016, pp. 1–9.

[27] K. Gardner, M. Harchol-Balter, and S. Borst, "Optimal scheduling for jobs with progressive deadlines," in *Proc. of INFOCOM*. IEEE, 2015, pp. 1113–1121.

[28] Measuring wireless networks and smartphone users in the field. [Online]. Available: http://livelab.recg.rice.edu/traces.html

[29] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proc. of AFIN*. Citeseer, 2014.