IEEE TRANSACTIONS ON
WIRELESS
COMMUNICATIONS

IEEE Xplore® Digital Library
http://ieeexplore.ieee.org/Xplore

Bandwidth and Energy-aware Resource Allocation for Cloud Radio Access Networks

Ayman Younis, *Student Member, IEEE*, Tuyen X. Tran, *Member, IEEE*, and Dario Pompili, *Senior Member, IEEE*

***Abstract*—Cloud Radio Access Network (C-RAN) is emerging as a transformative paradigmatic architecture for the next generation of cellular networks. In this article, a novel resource allocation solution that optimizes the energy consumption of a C-RAN is proposed. First, an energy consumption model that characterizes the computation energy of the Base Band Unit (BBU) pool is introduced based on empirical results collected from a programmable C-RAN testbed. Then, the resource allocation problem is split into two subproblems—namely the Bandwidth Power Allocation (BPA) and the BBU Energy-Aware Resource Allocation (EARA). The BPA, which is first cast via Mixed-Integer Nonlinear Programming (MINLP) and then reformulated as a convex problem, aims at assigning a feasible bandwidth and power to serve all users while meeting their Quality of Service (QoS) requirements. The second subproblem, i.e., the BBU EARA, is defined as a bin-packing problem that aims at minimizing the number of active Virtual Machines (VMs) in the BBU pool to save energy. Simulation results coupled with real-time experiments on a small-scale C-RAN testbed show that the proposed resource allocation solution optimizes the energy consumption of the network while meeting practical constraints and QoS requirements, and outperforms competing algorithms such as Best Fit Decreasing (BFD), RRH-Clustering (RC), and SINR-based.**

***Index Terms*—C-RAN; Testbed; Software Defined Radio; Network Virtualization; Profiling; Resource Allocation.**

I. Introduction

Over the last few years, the proliferation of personal mobile-computing devices such as tablets and smart phones, along with a plethora of data-intensive mobile applications, has resulted in a tremendous increase in demand for ubiquitous and high data-rate wireless communications. To cope with this exponentially growing rate, it is expected that cellular wireless systems would need $100\times$ increase in Spectral Efficiency (SE) and $1000\times$ improvement in Energy Efficiency (EE) by 2020, which calls for a technological revolution. While the current cellular network architecture was not originally designed for such capabilities, Cloud Radio Access Network (C-RAN) [2] has been introduced recently as a revolutionary paradigmatic redesign of the cellular architecture to address the huge increase in data traffic as well as to reduce the capital expenditure (CAPEX) and operating expenditure (OPEX) [3].

The idea of C-RAN is to decouple the computational functionalities from the distributed Base Station (BS, a.k.a. eNodeB in LTE) and to consolidate them in a centralized processing center. A typical C-RAN is composed of: (i) light-weight, distributed Radio Remote Heads (RRHs) plus antennae, which are located at the remote site and are controlled by a centralized virtual base station pool, (ii) the Base Band Unit (BBU) composed of high-speed programmable processors and real-time virtualization technology to carry out the digital processing tasks, and (iii) low-latency high-bandwidth optical fibers, which connect the Remote Radio Heads (RRHs) to the BBU pool. In a centralized BBU pool, since all the information about the network resides in a common place, the BBU can exchange control data at Gbps rate. This centralized characteristic—along with virtualization technology and low-cost relay-like RRHs—provides a higher degree of freedom to make optimized decisions, and has made C-RAN a promising technology candidate to be incorporated into the Fifth Generation (5G) wireless network, especially for urban/high-density areas. For instance, based on the global view of the network condition and on the traffic demand information available at the BBU pool, dynamic provisioning and allocation of spectrum, computing, and radio resources can improve network performance [4]–[9]. Furthermore, fueled by the strong computing capabilities and storage resources at the BBU pool, C-RAN can provide a central port for traffic offloading and content management via edge caching [10]–[12]. In some respect, C-RAN paves the way for bridging the gap between two so-far disconnected worlds: wireless cellular communications and cloud computing.

In a BBU pool, most of the communication functionalities are implemented in part or fully in a virtualized environment hosted over general-purpose computing servers that are housed in one or more racks in a nearby cloud datacenter. It is therefore crucial to design and provision the virtualized environment properly in order to make it flexible and energy efficient while also capable of handling intensive computations. Such a virtualized environment can be realized via the use of Virtual Machines (VMs). The flexible reconfigurability of the virtualized BBU allows for it to be dynamically resized 'on the fly' in order to meet the fluctuations in capacity demands. This *elasticity* will enable significant improvement in user Quality of Service (QoS) as well as efficiency in energy and computing resource utilization in C-RANs. However, determining the computational resources of a virtualized BBU that is capable of providing adequate processing capabilities with respect to the traffic load presents non-trivial engineering challenges.

Our Vision: Although C-RAN offers many crucial updated features that make it possible to transform conventional Radio Access Networks (RAN) from hardware-defined infrastruc-

A preliminary shorter version of this work is in the *Proc. of the IEEE International Conference on Autonomic Computing (ICAC).* July'17 [1].

The authors are with the Department of Electrical and Computer Engineering, Rutgers University–New Brusnwick, NJ, USA; E-mails: {a.younis, tuyen.tran, pompili}@rutgers.edu.

This work was partially supported by the US National Science Foundation (NSF) Grant No. CNS-1319945.

Digital Object Identifier: 10.1109/TWC.2018.2860008

tures to a software-defined environment—such services are referred to as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) [13]—there are several critical hardware challenges needed to be identified and addressed in order to achieve the full benefits of using C-RAN in 5G systems. Obviously, the energy consumption of a C-RAN can be significantly reduced if we optimize the computational BBU resources such as the computation frequency (CPU cycles per second). On the other hand, the communication resources depend on multiple parameters including radio signal bandwidth and the Modulation and Coding Scheme (MCS) index.

In this article, *we seek to design an efficient resource allocation solution, considering the computational requirements of the virtualized BBU over a real-world implementation of a small-scale C-RAN system*. Software implementations and real hardware are essential to understand the runtime complexity as well as the performance limits of the BBU in terms of processing throughput and latency and how they translate to mobile-user QoS metrics. The realization of the C-RAN emulation testbed on virtualized general-purpose computing servers will allow for *profiling* of the computational complexity of the different communication functionalities implemented in software. In particular, such profiling results will provide a "mapping" from the number and combination of different types of user traffic to VM computational capacity. *Hence, we aim at establishing empirical models for the estimation of processing time and CPU utilization with respect to different radio-resource configurations and traffic loads*. Our model will provide researchers and practitioners with real-world insights and the necessary tool for designing advanced and efficient resource provisioning and allocation strategies in C-RANs.

Related Work: There has been a considerable number of recent works studying the benefits of C-RAN from the cooperative communications perspectives. For instance, the works in [14]–[17] consider the power minimization problem by jointly optimizing the set of active RRHs and precoding or beamforming design. The considered power models consist of the RRH transmission power [14], and additionally the user transmission power in [16], transport network power in [15], and power consumption at the BBU pool in [17]. In addition, the tradeoff between transmission power and delay performance is investigated in [18]–[20] via different approaches. Furthermore, the works in [9], [21]–[23] address the fronthaul uplink compression problem in C-RAN. While showing promising performance gains brought by the centralization and optimization of C-RAN, *these works often overlook the system issues and mostly rely on simplified assumptions when modeling the computational resources of the BBU.*

From the system perspectives, several LTE RAN prototypes have been implemented over General-Purpose Platforms (GPPs) such as the Intel solutions based on hybrid GPP-accelerator [24], Amarisoft solution [25] , and OpenAirInterface (OAI) platform [26]. Studies on these systems have demonstrated the preliminary potential benefits of C-RAN in improving statistical multiplexing gains, energy efficiency, and computing resource utilization. Field-trial results in [2], [27] show the feasibility of deploying C-RAN fronthaul using Common Packet Radio Interface (CPRI) compression—which specifies the interface between two LTE functional blocks, i.e., the baseband processing and the radio—single fiber bidirection, and wavelength-division multiplexing. The authors in [28] focus on minimizing computational and networking latencies by VMs or containers. Kong et al. [29] present the architecture and implementation of a BBU cluster testbed to improve energy efficiency in C-RAN. Wu [30] shows a high-level architecture for programmable RAN (PRAN) that centralizes BSs' L1/L2 processing of BBU pool onto cluster of commodity servers. This approach shows the feasibility of fast data path control and efficiency of resource pooling. The work in [31] presents the cross-layer resource allocation problem as a Mixed-Integer Nonlinear Programming (MINLP), which considers elastic service scaling, RRH selection, and beamforming. In [32], the authors propose a workload consolidation framework for minimizing energy consumption in C-RAN by reducing the number of baseband processing servers used.

In summary, these works perform theoretical studies of resource allocation problems, overall system architecture, feasibility of virtual software BS stacks, performance requirements, and analysis of optical links between the RRHs and the BBU cloud. However, most of these systems are either proprietary or ad-hoc based, and do not provide a generalized characterization that can be used for the design of new algorithms. *In contrast, our work is based on real-world C-RAN testbed experiments that allowed us to derive a realistic empirical model for the processing power consumption at the BBU pool. Based on such models, we formulated and solved two subproblems to achieve an optimized tradeoff between energy consumption and user QoS.*

Our Contributions: The objective of this article is to propose an efficient resource allocation scheme that aims at minimizing the overall energy consumption of C-RAN, including the power consumption of the BBU pool and the RRHs. In particular, using empirical data collected from our real-time OAI testbed, we modeled the network energy consumption in a C-RAN system, which consists of two main parts: the computation energy consumed in the BBU pool and the Radio Frequency (RF) energy transmitted by RRHs. We established BBU computation model via testbed experiments and propose resource allocation techniques to optimize the number of active servers while ensuring QoS requirements for the users in a downlink C-RAN system. Given the importance of designing effective resource management solutions in C-RAN and the lack of experimental studies for the computational performance and requirements of the BBU pool, we make the following contributions in this article.

- We design and implement a programmable C-RAN testbed comprising of a virtualized BBU connected to multiple eNodeBs (eNBs). The BBU is implemented using an open-source software platform that allows for simulation and emulation of the LTE protocol stack. The eNBs are realized using programmable USRP Software Defined Radio (SDR) boards.
- We perform extensive experiments with transmissions between the eNB and the UE under various configurations to identify the runtime complexity and performance limits

of the BBU in terms of processing time, throughput, and latency. It is shown that the processing time and CPU utilization of the BBU increase with the Modulation and Coding Scheme (MCS) index and with the number of allocated Physical Resource Blocks (PRBs).

- Using empirical data from testbed experiments, we model the BBU processing time as a function of the CPU frequency, MCS, and PRBs; and the BBU's CPU utilization as a linearly increasing function of the maximum downlink data rate. These models provide insights and key inputs to formulate/design/evaluate resource-management strategies in C-RAN.
- We split the resource allocation problem into two subproblems—namely the Bandwidth Power Allocation (BPA) and the BBU Energy-Aware Resource Allocation (EARA). The BPA, which is first cast via MINLP and then reformulated as a convex problem, aims at assigning a feasible bandwidth and power to serve all users while meeting their QoS requirements. The second subproblem, the BBU EARA, is defined as a bin-packing problem that aims at minimizing the number of active Virtual Machines (VMs) in the BBU pool to save energy.
- Our approach leverages the established BBU computation model and introduces novel techniques to optimize dynamically the UE-RRH and RRH-BBU associations. Simulation results coupled with real-time experiments on a small-scale C-RAN testbed show that the proposed resource allocation solution minimizes the energy consumption of the network while meeting practical constraints and QoS requirements, and outperforms competing algorithms such as Best Fit Decreasing (BFD), RRH-Clustering (RC), and SINR-based.

Article Organization: In Sect. II, we describe the C-RAN architecture and the related system challenges; in Sect. III, we introduce the system and power consumption models considered throughout this work; in Sect. IV, we formulate the resource allocation optimization problem, discuss its properties, decompose it into two simpler subproblems, and propose two algorithms to solve them; in Sect. V, we present our testbed experiment results as well as numerical simulation results to evaluate performance of our proposed algorithms. Finally, we conclude the article in Sect. VI.

II. C-RAN Architecture and Implementation Challenges

We describe here the C-RAN system architecture and the OAI software platform that is capable of realizing a virtualized C-RAN system. We then discuss the critical issues of a C-RAN implementation and virtualization.

A. C-RAN Architecture

The general architecture of C-RAN mainly consists of two parts: the distributed RRHs plus antennae deployed at the remote site and the centralized BBU pool hosted in a cloud datacenter. The BBU pool consists of multiple BBUs, each hosted on a VM and connected to the corresponding RRH via high-bandwidth low-latency media (e.g., use of optic fibers allows for maximum distance separation of $40\ \mathrm{Km}$ between the RRH and its BBU [2]). Packet-level processing, Medium Access Control (MAC), Physical-layer (PHY) baseband processing, and RF functionalities may be split between the BBU and the RRHs depending on the specific C-RAN implementation [33]. In this article, we consider the full centralization of C-RAN to exploit fully the potential of this paradigm where only RF functionalities are deployed at the RRHs. Based on the network performance and system implementation complexity, each BBU can be assigned to one RRH, as shown in Fig. 1(a), or the BBUs can be consolidated into one entity, called BBU pool, that takes care of performing baseband PHY- and MAC-layer processing, as depicted in Fig. 1(b).

B. Emulation Platform

We choose an open-source software implementation of LTE standard called OAI [26] to realize the virtualized C-RAN system. This is a wireless communication platform developed by EUROCOM that provides a complete flexible cellular ecosystem towards an open-source 5G implementation. OAI can be used to build and customize mobile network operators consisting of eNBs and commercial Off-The-Shelf (COTS) UEs as well as software-defined UEs. In addition, OAI offers tools to configure and monitor the RAN in real time via a software radio front-end connected to a host computer for processing. This approach is similar to other SDR prototyping platforms in the wireless networking research community such as OpenBTS [34].

The structure of OAI mainly consists of two components: one, called $\mathrm{Openairinterface5g}$, is used for building and running eNB units; the other, called $\mathrm{Openair\text{-}cn}$, is responsible for building and running the Evolved Packet Core (EPC) networks, as shown in Fig. 1(c). The $\mathrm{Openair\text{-}cn}$ component provides a programmable environment to implement and manage the following network elements: Mobility Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (S-GW), and PDN Gateway (P-GW). Figure 2(a) shows a downlink functional block diagram of an eNB. The RRH includes only time-domain RF and analog-to-digital functionalities while the BBU contains all the other functions. Furthermore, it can be observed that the overall processing is the sum of per User Processing (UP) and Cell Processing (CP). The UP depends only on the MCS and on the resource blocks allocated to the users as well as on the Signal-to-Noise Ratio (SNR) and channel conditions; whereas the CP depends on the channel bandwidth, thus imposing a constant base processing load on the system.

C. C-RAN Implementation Challenges

Although C-RAN offers many key features for RAN systems, there are several critical hardware challenges needed to be identified and addressed to achieve the benefits of using C-RAN in 5G systems. These challenges are listed as follows.

1) Testbed capacity: a typical C-RAN testbed should be implemented to deal with tens to hundreds of RRHs at the same time; hence, the testbed must be equipped with high computational resources and low-latency operation

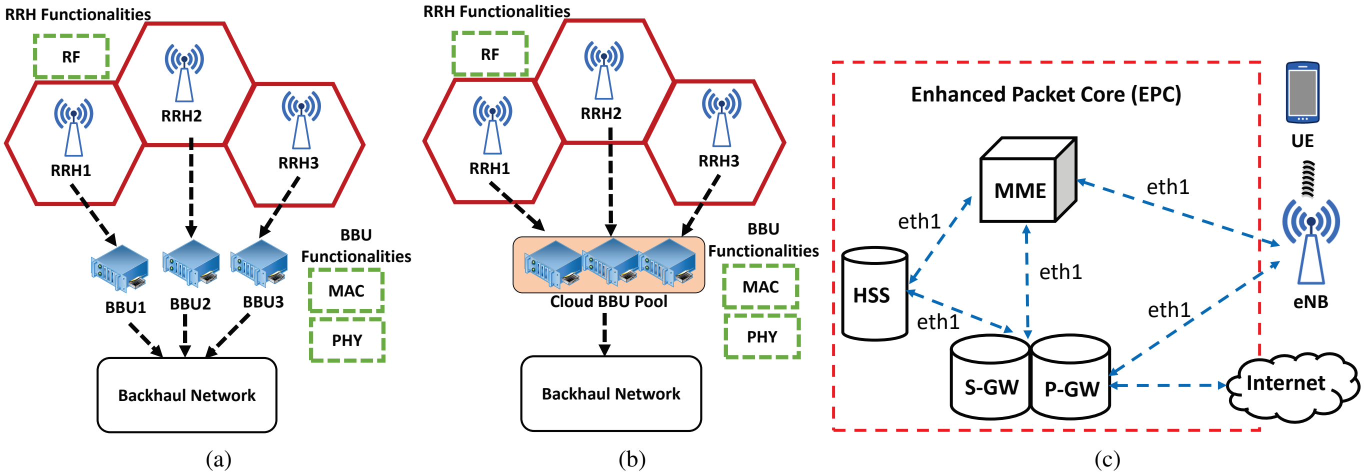


Fig. 1. (a) Each Base Band Unit (BBU) is assigned to one Remote Radio Head (RRH); (b) Consolidated BBU; and (c) Evolved Packet Core (EPC) network topology diagram.

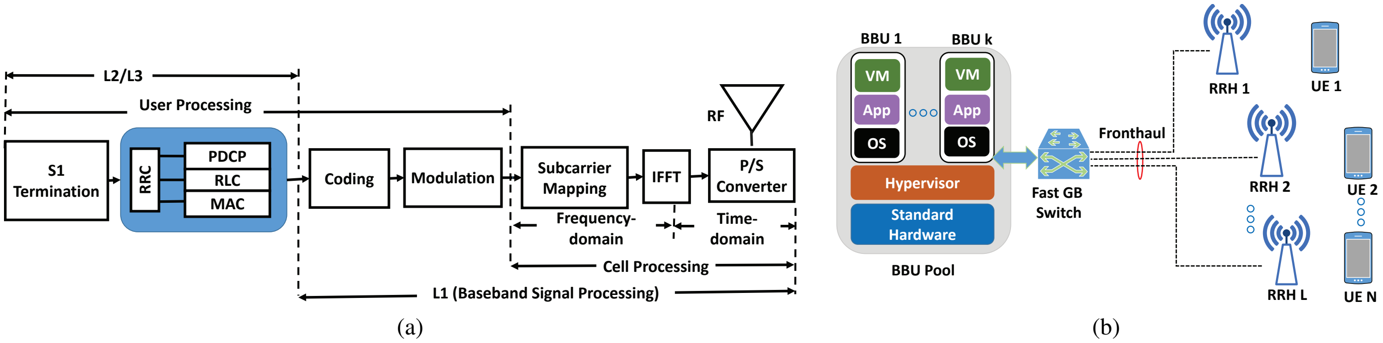


Fig. 2. (a) Downlink-BBU pool physical processing blocks; and (b) A conceptual structure of C-RAN network where the BBU pool can be realized by real-time VMs.

system. Moreover, reliable synchronization between the BBU pool and the RRHs over the front-haul links should be achieved.

2) Testbed latency: the Frequency Division Duplex (FDD) LTE Hybrid Automatic Repeat Request (HARQ) requires a Round Trip Time (RTT) of at most 8 ms; therefore, a real-time requirement for hardware and software environments must be provisioned for the BBU pool. Furthermore, the testbed hardware should provide the capabilities to enable dynamic resource provisioning and sharing in order to address the geographical and temporal variation of the traffic load in the network.
3) Other testbed requirements: other issues related to system front-haul multiplexing and the back-haul link cost, energy optimization, and channel estimation should also be considered in the testbed implementation.

In this work, we put a particular focus on LTE FDD, which consists of the following layers: i) LTE PHY with symbol-level processing, ii) MAC layer, which supports wide-band multiuser scheduling and HARQ.

III. System Model

In this section, we first describe the network setting, communication and BBU computation models; then, we describe the network energy consumption model.

TABLE I
Summary of key notations.

Notation	Definition
$\mathcal{N}$	the set of UEs
$\mathcal{L}$	the set of RRHs
$\mathcal{K}$	the set of BBUs
h_{ij}	the channel gain between UE i and RRH j
B_{ij}	the bandwidth allocated to UE i from RRH j
g_{ij}	the transmit power allocated to UE i from RRH j
r_{ij}	the data rate of UE i when it associated with RRH j
C_i	the computation capacity allocated for UE i in BBU pool
f_i^{CPS}	the computation CPU frequency in BBU pool for UE i
G_i, D_i	the positive testbed constants depending on the setup
$\mathcal{E}_k^{bbu}$	the computation power consumption of BB k
$\mathcal{E}^a$, $\mathcal{E}^s$	the static and sleeping power of a VM , respectively
$\mathcal{E}_i(C_i)$	the CPU power consumption
P_{pon}	the power consumption of PON
P_{olt}	the acquired power of OLT
P_j^{fh}	the transport link power consumption of fronthaul link j
P_j^a, P_j^s	the power of RRH j in active and sleep state, respectively
P_{net}	the total power consumption of C-RAN network
P^{tr}, P^c	the RRH transmit power and constant power, respectively
λ_i, μ_i	the Lagrange multipliers
U	the maximum number of BBU k in the cloud

A. Network Description

As illustrated in Fig. 2(b), we consider a C-RAN system consisting of a set $\mathcal{N} = \{1, 2, ..., N\}$ of N UEs and a set $\mathcal{L} = \{1, 2, ..., L\}$ of L RRHs. Each UE is equipped with single antenna while each RRH has $A > 1$ antennas. All the RRHs are connected through Fast GB switch to a common processing center comprising of a set of K BBUs denoted as $\mathcal{K} = \{1, 2, ..., K\}$. The BBU pool is composed of high-speed programmable processors and real-time VMs to carry out PHY/MAC-layer functionalities. The BBUs could serve each UE by generating a VM to provide computation resource as common datacenters do in a cloud-based system. The number of VMs generated on BBU pool is limited, which means that one BBU can only support a limited number of UEs. We assume that each UE can only be supported by one VM.

We consider that each BBU can serve one or more RRHs, and the RRHs can cooperate with each other for downlink transmissions to the UEs. We assume that $h_{ij} \in \mathbb{C}^{A\times 1}$ is the channel gain between UE i and RRH j. The bandwidth and transmit power allocated to UE i from RRH j are denoted as B_{ij} and g_{ij}, respectively. The Signal-to-Interference-plus-Noise Ratio (SINR) for UE i when receiving signal from RRH j is given by,

$$\gamma_{ij} = \frac{g_{ij} h_{ij}}{B_{ij}\left(N_0 + I_i\right)}, \forall i \in \mathcal{N}, j \in \mathcal{L}, \tag{1}$$

where N_0 is the Power Spectral Density (PSD) of the Additive White Gaussian Noise (AWGN); I_i is the maximum interference introduced by other active RRHs with unit bandwidth, which can be rewritten as,

$$I_i = \sum_{k\in\mathcal{L}\setminus\{j\}} g_k^{max} h_{ik} / B_k^{max}, \forall i \in \mathcal{N}, \tag{2}$$

where g_k^{max} and B_k^{max} are the maximum transmission power and bandwidth, respectively. Hence, the achievable data rate of UE i when it is associated with RRH j can be calculated as,

$$r_{ij} = B_{ij} \log_2\left[1 + \gamma_{ij}\right], \forall i \in \mathcal{N}, j \in \mathcal{L}. \tag{3}$$

Interference management mechanisms, such as Enhanced Inter Cell Interference Coordination (eICIC) and Coordinated Multipoint (CoMP) [35], [36], can be employed to reduce interference in the network. Benefiting from centralizing BBU resources in a C-RAN, those schemes reduce processing and transmitting delays since signal processing from many cells can be done over one BBU pool.

B. Active-Sleep Network Power Model

In C-RAN, when BBU and RRH are separated, the processing time at the BBU is reduced and the delay calculation has to consider propagation delays and interface latencies between RRH and BBU. However, we are going to model the energy consumption according to the following requirements, which are mentioned in [37]. The first requirement is that the RTT between RRH and BBU equipped with a CPRI link cannot exceed 700 μs for LTE and 400 μs for LTE-Advanced. Hence, the length of a BBU RRH link should not exceed 15 km to avoid too high round trip-delays while the speed of light in a fiber is approximately 200 m/μs. Consequently, this leaves the BBU PHY layer only with around $2.3 - 2.6$ ms for signal processing at a centralized processing pool. The propagation delay, corresponding to timing advance, between RRH and UE, affects only the UE processing time. The timing advance value can be up to 0.67 ms (equivalent to a maximum cell radius of 100 km). Once the BBU received a subframe (1 ms duration) from the RRH, the BBU has to decode the subframe as well as assemble and return another subframe back to the RRH within a *hard deadline* ≤ 3 ms depending on the RRH–BBU distance.

In a real-world implementation of a C-RAN testbed, which we will describe in detail in Sect. V, the CPU utilization of the BBU linearly increase with the PRB and MCS index. Under this premise, we can consider the computation capacity C_i [cycles/s] that is allocated for UE i in the BBU pool as a linearly increasing function of the user downlink data rate. Specifically, the computation capacity for the data of UE i can be modeled as,

$$C_i = G_i f_i^{\mathrm{CPS}} + D_i, \forall i \in \mathcal{N}, \tag{4}$$

where G_i and D_i are positive constants that can be estimated via offline profiling of the C-RAN testbed. We use f_i^{CPS} to refer to the computation frequency (CPU cycles per second) in the BBU pool for UE i.

In this work, we assume that the total power consumption in a downlink C-RAN network system contains two main parts: the computation power spent in the BBU pool and the power consumption of RRHs in the downlink transmission. In practice, the BBU pool can dynamically adjust the VMs' computation capacities to handle the dynamics of user traffic demand and channel states. The power consumption of the BBU pool is closely related to computing workloads for the baseband signal processing [38]. We use a_{ki} to indicate whether UE i is served by the VM generated by BBU k, which can expressed as,

$$a_{ki} = \begin{cases} 1 & \text{UE } i \text{ is served by BBU } k, \\ 0 & \text{otherwise,} \end{cases} \forall k \in \mathcal{K}, i \in \mathcal{N}. \tag{5}$$

Hence, we can model the computation power consumption of BBU k corresponding to UE i as,

$$\mathcal{E}_k^{bbu} = \begin{cases} \mathcal{E}^a + \sum\limits_{i\in\mathcal{N}} a_{ki}\mathcal{E}_i(C_i) & \text{BBU } k \text{ is active,} \\ \mathcal{E}^s & \text{BBU } k \text{ is sleep,} \end{cases} \forall k \in \mathcal{K}, \tag{6}$$

where parameter $\mathcal{E}^a$ represents the statistic part of the power consumption of a VM in *working mode*, which is constant, while $\mathcal{E}_i(C_i)$ represents the CPU power consumption used to process baseband signal of UE i. Additionally, we use $\mathcal{E}^s$ to denote the power consumption of VM i in *sleeping mode*. The increasing cost power from switching from sleep to active mode can be formulated as,

$$\Delta_k^{bbu} = \mathcal{E}^a + \sum_{i\in\mathcal{N}} a_{ki}\mathcal{E}_i(C_i) - \mathcal{E}^s, \forall k \in \mathcal{K}. \tag{7}$$

According to [39], the amount of power consumption corresponding to UE i can be modeled as,

$$\mathcal{E}_i(C_i) = w_i C_i, \forall i \in \mathcal{N}, \tag{8}$$

where $w_i > 0$ is a constant.

Depending on Passive Optical Network (PON) model [40], the Optical Line Terminal (OLT), which connects a set of associated Optical Network Unites (ONUs) through fiber links, can be used for C-RAN. Therefore, the power consumption of C-RAN network can be denoted as,

$$P_{pon} = P_{olt} + \sum_{j \in \mathcal{L}} P_j^{fh}, \tag{9}$$

where P_{olt} is the acquired power of OLT and P_j^{fh} is the transport link power consumption of fronthaul link j, which is given by,

$$P_j^{fh} = \begin{cases} P_j^a & \text{RRH } j \text{ is active,} \\ P_j^s & \text{RRH } j \text{ is sleep,} \end{cases} \forall j \in \mathcal{L}, \tag{10}$$

where P_j^a and P_j^s, with $P_j^a > P_j^s$, are consumed power in active and sleep state, respectively. Specifically, if RRH j is in active state, the value of P_j^{fh} would increase by,

$$\Delta_j^{fh} = P_j^a - P_j^s, \forall j \in \mathcal{L}. \tag{11}$$

A binary variable b_j is introduced to indicate whether RRH j is active or not, i.e.,

$$b_j = \begin{cases} 1 & \text{RRH } j \text{ is active,} \\ 0 & \text{RRH } j \text{ is sleep,} \end{cases} \forall j \in \mathcal{L}. \tag{12}$$

Therefore, P_{pon} can be transformed as,

$$\begin{aligned} P_{pon} &= P_{olt} + \sum_{j \in \mathcal{L}} \big(P_j^a - P_j^s\big) b_j + \sum_{j \in L} P_j^s \\ &= P_{olt} + \sum_{j \in \mathcal{L}} \Delta_j^{fh} b_j + \sum_{j \in L} P_j^s. \end{aligned} \tag{13}$$

Hence, the total power consumption of the C-RAN includes the BBU pool, $\mathcal{E}_{bbu}$, the PON, and the transmit power in RRHs, P^{tr}, which can be written as,

$$\begin{aligned} P_{net} &= \mathcal{E}_{bbu} + P_{pon} + P^{tr} \\ &= \sum_{k \in \mathcal{K}} \mathcal{E}_k^{bbu} + P_{olt} + \sum_{j \in \mathcal{L}} \Delta_j^{fh} b_j + \sum_{j \in L} P_j^s + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{L}} g_{ij} \\ &= \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} \Delta_k^{bbu} a_{ki} + \sum_{k \in \mathcal{K}} \mathcal{E}^s + P_{olt} + \sum_{j \in \mathcal{L}} P_j^s b_j \\ &\qquad + \sum_{j \in \mathcal{L}} P_j^s + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{L}} g_{ij} \\ &= \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} \left(\mathcal{E}^a + \mathcal{E}_i(C_i)\right) a_{ki} + \sum_{j \in \mathcal{L}} \big(P_j^a - P_j^s\big) b_j \\ &\qquad + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{L}} g_{ij} + P_c, \end{aligned} \tag{14}$$

where $P_c = P_{olt} + \sum\limits_{k \in \mathcal{K}} \mathcal{E}^s + \sum\limits_{j \in \mathcal{L}} P_j^s$ is a constant.

IV. Proposed Solution

We present now a novel resource allocation framework that optimizes the total energy consumption in the computation and transmission parts of the C-RAN network. We formulate the network power consumption minimization problem, followed by our solution approach.

A. Resource Allocation Problem for Network Power Consumption Minimization

The network power consumption in (14) suggests the two following strategies to reduce the network power consumption: (i) reduce the number of active VMs in BBU pool and (ii) reduce the transmission power consumption. Our goal is to propose an adaptive resource allocation strategy in C-RAN that minimizes the total energy consumption at the BBU pool and at the distributed RRHs. The energy minimization problem can be mathematically formulated as,

$$\begin{aligned} \mathcal{P}0: \quad & \underset{b_j, a_{ki}, B_{ij}, g_{ij}}{\text{minimize}} \quad P_{net} & \text{(15a)} \\ \text{s.t.} \quad & r_{ij} \geq r_i^{\min}, \forall i \in \mathcal{N}, j \in \mathcal{L} & \text{(15b)} \\ & \sum\nolimits_{i \in \mathcal{N}} B_{ij} \leq b_j B_j^{\max}, \forall j \in \mathcal{L}, & \text{(15c)} \\ & \sum\nolimits_{i \in \mathcal{N}} g_{ij} \leq g_j^{\max}, \forall j \in \mathcal{L}, & \text{(15d)} \\ & B_{ij} \geq 0, g_{ij} \geq 0, \forall i \in \mathcal{N}, \forall j \in \mathcal{L}, & \text{(15e)} \\ & b_j, a_{ki} \in \{0,1\}, \forall i \in \mathcal{N}, j \in \mathcal{L}, k \in \mathcal{K}, & \text{(15f)} \end{aligned}$$

where constraint (15b) guarantees the QoS of UEs by keeping the data rate of UE i above or equal the target; and (15c) and (15d) account for the bandwidth and power budgets of RRH j, respectively. Problem $\mathcal{P}0$ is a MINLP, which is NP-hard and difficult to solve [41]. Our goal in this article is to design a low-complexity, suboptimal solution to minimize the network energy consumption in C-RAN, as will be presented in the next subsections.

B. A Divide-and-conquer Approach: Decomposing the Resource Allocation Problem

As previously stated, the optimization problem defined by $\mathcal{P}0$ is a MINLP since both classes of binary variables b_j and a_{ki} take values in discrete sets. Most solutions for MINLP relax the integer variables into continuous ones so that appropriate linear/nonlinear optimization methods can be applied. Intuitively, exhaustive search could obtain optimal solutions for $\mathcal{P}0$; however, the complexity of finding the optimal solution is too high even for medium-scale cases. Hence, we need to follow a different approach based on a divide-and-conquer strategy involving breaking the original problem into simpler subproblems that can be solved directly; the solutions to the subproblems are then combined to give a solution to the original problem. Because of the properties of the objective function in (15), in fact, we can split the energy minimization problem into two subproblems. The first subproblem, the BPA, aims at assigning a feasible bandwidth and power to serve all UEs while meeting their QoS requirements; while the second subproblem, the BBU EARA, consists (i) in deciding which UEs in each RRH-UE cluster should be served by which BBU in the pool and (ii) in minimizing the number of working VMs in the BBU pool. We will formulate the second subproblem as a bin-packing problem. The two subproblems will be elaborated in detail in the following subsections.

C. Bandwidth and Power Allocation Algorithm (BPA)

In this subproblem, our goal to assign the bandwidth and power budgets of the RRHs so as to satisfy the rate requirements of all users. Given a set of users $\mathcal{N}_j$ served by RRH j, we can formulate a feasible bandwidth and power allocation to serve all users as,

$$\mathcal{P}1: \text{find } \ B_{ij}, g_{ij} \tag{16a}$$

$$\text{s.t. } \ r_{ij} \geq r_i^{\min}, \forall i \in \mathcal{N}_j, j \in \mathcal{L}, \tag{16b}$$

$$\sum\nolimits_{i \in \mathcal{N}_j} B_{ij} \leq B_j^{\max}, \forall j \in \mathcal{L}, \tag{16c}$$

$$\sum\nolimits_{i \in \mathcal{N}_j} g_{ij} \leq g_j^{\max}, \forall j \in \mathcal{L}, \tag{16d}$$

$$B_{ij} \geq 0, g_{ij} \geq 0, \forall i \in \mathcal{N}_j, \forall j \in \mathcal{L}. \tag{16e}$$

We claim that all UEs in $\mathcal{N}_j$ can be served by RRH j if a feasible solution to (16) exists. However, solving the feasibility problem $\mathcal{P}1$ is not straightforward; therefore, we reformulate $\mathcal{P}1$ into an equivalent form that is easier to address. Specifically, considering that the UEs in $\mathcal{N}_j$ consume all bandwidth $B_j^{\max}$, we aim at finding the minimum power consumption of RRH j with QoS requirements so that the optimization subproblem can be represented as,

$$\mathcal{P}2: \underset{B_{ij}, g_{ij}}{\text{minimize}} \ \sum\nolimits_{i \in \mathcal{N}_j} g_{ij} \tag{17a}$$

$$\text{s.t. } \ (16\text{b}) \sim (16\text{e}). \tag{17b}$$

From constraint (16b) and the definition of r_{ij} in (3), we conclude that,

$$g_{ij} = \frac{N_0 B_{ij}}{h_{ij}} \left(2^{\frac{r_i^{\min}}{B_{ij}}} - 1 \right), \forall i \in \mathcal{N}_j, j \in \mathcal{L}. \tag{18}$$

Finally, by substituting (18) into (17), the optimization subproblem can be recast as,

$$\mathcal{P}3: \underset{B_{ij}}{\text{minimize}} \ \sum\nolimits_{i \in \mathcal{N}_j} \frac{N_0 B_{ij}}{h_{ij}} \left(2^{\frac{r_i^{\min}}{B_{ij}}} - 1 \right), \tag{19a}$$

$$\text{s.t. } \ \sum\nolimits_{i \in \mathcal{N}_j} B_{ij} = B_j^{\max}, \forall j \in \mathcal{L}, \tag{19b}$$

$$B_{ij} \geq 0, \forall i \in \mathcal{N}_j, \forall j \in \mathcal{L}. \tag{19c}$$

Lemma 1. *Problem $\mathcal{P}3$ in* (19) *is convex.*

Proof. Please see the Appendix. □

We can define the Lagrangian associated with $\mathcal{P}3$ as,

$$\begin{aligned} \mathcal{L}(B_{ij}, \lambda_i, \mu_i) = & \sum_{i \in \mathcal{N}_j} \frac{N_0 B_{ij}}{h_{ij}} \left(2^{\frac{r_i^{\min}}{B_{ij}}} - 1 \right) \\ & + \sum_{i \in \mathcal{N}_j} \lambda_i \left(B_{ij} - B_j^{\max} \right) - \sum_{i \in \mathcal{N}_j} \mu_i B_{ij}, \end{aligned} \tag{20}$$

where λ_i and μ_i are Lagrange multipliers. Suppose B_{ij}^*, λ_i^*, and μ_i^* are the primal and dual points with zero dual gap [42]; by applying Karush-Kuhn-Tucker (KKT) conditions [42], the following optimal values can be obtained as,

$$\lambda_i^* = -\frac{N_0}{h_{ij}} \left[\left(1 - \frac{r_i^{\min} \ln 2}{B_{ij}^*} \right) 2^{\frac{r_i^{\min}}{B_{ij}^*}} - 1 \right], \tag{21}$$

$$\sum\nolimits_{i \in \mathcal{N}_j} B_{ij}^* = B_j^{\max}, \forall j \in \mathcal{L}, \tag{22}$$

$$\mu_i^* = 0, B_{ij}^* > 0, \forall i \in \mathcal{N}_j, \forall j \in \mathcal{L}. \tag{23}$$

Algorithm 1 BPA Algorithm for UE-RRH Clustering

```
Initialize: $x = 0$, $\lambda_i^{(x)} = 0$, $\lambda_i^{\min} = 0$, and $\lambda_i^{\max} = \Lambda_i$, $\forall i \in \mathcal{N}$
Output: $B_{ij}^*$, $g_{ij}^*$
repeat
    $x = x + 1$, $\lambda_i^{(x)} = (\lambda_i^{\max} + \lambda_i^{\min})/2$
    for $i \in \mathcal{N}$ do
        Determine $B_{ij}$ from $\mathcal{P}3$
    end for
    if $\sum_{i \in \mathcal{N}} B_{ij} > B_j^{\max}$ then
        $\lambda_i^{\min} = \lambda_i^{(x)}$
    else
        $\lambda_i^{\max} = \lambda_i^{(x)}$
    end if
until $|\lambda_i^{(x)} - \lambda_i^{(x-1)}| \leq \epsilon$
for $i \in \mathcal{N}$ do
    $B_{ij}^* = B_{ij}$
    Determine $g_{ij}^*$ from (18)
end for
```

The BPA algorithm is detailed in Algorithm 1, where ϵ and Λ_i are a tolerance and an appropriately large number, respectively. Suppose Algorithm 1 needs a total number of T iterations to converge or the maximum number of iterations is set to T, then the computational complexity can be approximately given as $\mathcal{O}\left(T \cdot N^2\right)$.

As the system knows the network optimal bandwidth and power budgets for the RRH from Algorithm 1, we can determine the set of active RRHs $\mathcal{L}_j$ required to serve the given set of users $\mathcal{N}$. The policy of optimal variable b_j^* can be written as,

$$b_j^* = \begin{cases} 1 & \text{RRH } j \text{ is active, if } g_i(\{j\}) \geq g_i^*(\{j\}) \\ & \text{and } B_i(\{j\} \geq B_i^*(\{j\}) \ , \forall j \in \mathcal{L}_j, i \in \mathcal{N}, \\ 0 & \text{RRH } j \text{ is sleep, otherwise} \end{cases} \tag{24}$$

D. BBU Energy-Aware Resource Allocation (EARA)

The BBU power consumption can be formulated as bin-packing problem, which seeks to assign a set of items in different sizes into the minimum number of bins. Each bin has a fixed capacity, so that the sum size of items assigned to one bin cannot exceed the bin capacity. In our case, each BBU is regarded as bin, and the UE-RRH association users are considered as items.

The main goal for this subproblem is to assign UE-RRH associations to different BBUs in the pool so as to set up the fronthaul link between BBUs and RRHs, and to minimize the number of BBUs in working mode to save more energy.

According to (14), we can cast the EARA problem as,

$$\mathcal{P}4: \underset{\hat{a}_{ir}, y_k}{\text{minimize}} \ \sum\nolimits_{k \in \mathcal{K}} \mathcal{E}_k^{bbu} y_k \tag{25a}$$

$$\text{s.t. } \ \sum\nolimits_{k \in \mathcal{K}} a_{ki} = 1, \forall i \in \mathcal{N}_j, \tag{25b}$$

$$\sum\nolimits_{i \in \mathcal{N}_j} s_i a_{ki} \leq U y_k, \forall k \in \mathcal{K}, \tag{25c}$$

$$a_{ki}, y_k \in \{0, 1\}, \forall k \in \mathcal{K}, i \in \mathcal{N}_j, \tag{25d}$$

where U is a fixed value representing the maximum number of the BBU k in the cloud; s_i accounts for the required baseband resources of UE i. Parameter y_k indicates whether BBU k is in working mode or not, i.e., $y_k = 1$ if BBU k is in working mode, and $y_k = 0$ otherwise.

Constraint (25b) ensures that the data from one UE can only be processed by one BBU; constraint (25c) guarantees that the number of UEs supported by the BBU is less than a maximum threshold. With these positions, we claim that $\mathcal{P}4$ is a typical 1-D bin-packing problem.

While bin packing is a classical NP-hard problem, several very efficient heuristic algorithms exist to find suboptimal solutions [43], [44]. *In this article, we propose a heuristic algorithm based on Best Fit Decreasing (BFD) method, called EARA, which is another bin-packing approximate algorithm and has better performance without increasing the complexity*. The idea is that we sort all of the UE-RRH clusters into a decreasing order based on the CPU power consumption metric, as defined in (8), for each UE i, $\forall i \in \mathcal{N}_j$ in the cluster. Then, we will try to put the UE i associated with each RRH j into the most full BBU where it fits, or activate a new BBU to serve it when no existed BBU in the active mode has enough space ability, until all the users in UE-RRH clusters are assigned to BBUs. The computation complexity of solving Algorithm 2 is the same with BFD bin-packing solution, which is $\mathcal{O}(N_j \log N_j)$, where N_j represents the number of user associated with RRH j.

Algorithm 2 EARA Algorithm for BBU Scheduling

```
Input: $\mathcal{N}$, $\mathcal{L}$, $\mathcal{K}$, $U$,$G_i$, $D_i$, $f_i^{\mathrm{CPS}}$, $\forall i \in \mathcal{N}, k \in \mathcal{K}$
Output: reallocated BBUs set
while $\mathcal{L} \neq \emptyset$ do
    Associate set $\mathcal{N}_j$ UE with active RRHs by using Algorithm 1
    Compute $\mathcal{E}_i(C_i)$, $\forall i \in \mathcal{N}_j$ from (8)
    Select $j^* = \operatorname{argmin}\{\mathcal{E}_{\mathrm{i}}(\mathrm{C_i})\}$ $\forall i \in \mathcal{N}_j, j^* \in \mathcal{L}$
    Find the most-loaded BBU $k$ in $\mathcal{K}$ which can be serve $j^*$
    if BBU $k$ is exists then
        Put $j^*$ into BBU $k$
    else
        Find empty BBU $m$ in $\mathcal{K}$, put $j^*$ into BBU $m$
    end if
end while
```

V. Performance Evaluation

In this section, we first detail the experimental setups and results for the programmable C-RAN testbed. Then, we evaluate the performance of our proposed resource allocation algorithms, BPA and EARA, via numerical simulation results.

A. Testbed Experiment

We present here our C-RAN testbed using OAI, including the testbed architecture, configuration, and experiment methods. Then, we analyze the performance of the virtualized BBU, i.e., the OAI eNB, in terms of packet delay, CPU processing time, and utilization under various PRB and MCS configurations.

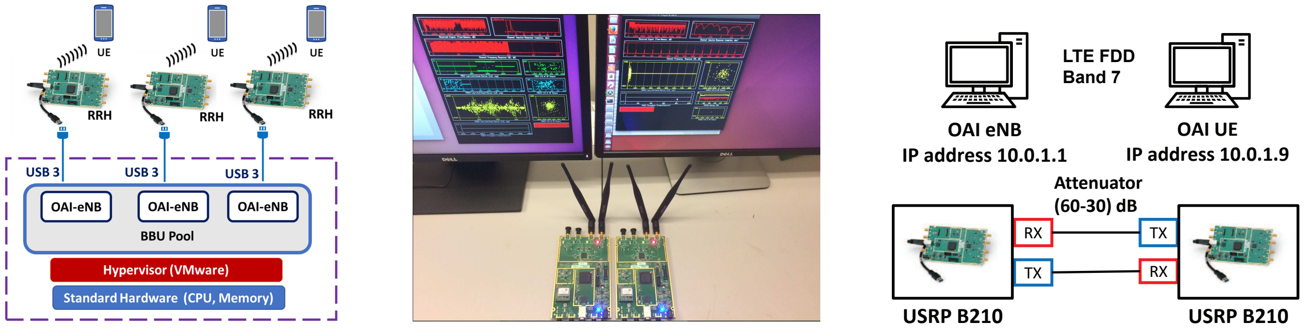
1) Testbed Architecture: Figure 3(a) illustrates the architecture of our testbed. The RRH front-ends of the C-RAN testbed are implemented using USRP SDR B210s, each supporting 2×2 MIMO with sample rate up to $62\ \mathrm{MS/s}$. In addition, each radio head is equipped with a GPSDO module for precise synchronization. Each instance of the virtual BBU is implemented using the OAI LTE stack, which is hosted in a VMware VM. All the RRHs are connected to the BBU pool (the physical servers hosting the VMs) via USB 3 connections. The Ubuntu 14.04 LTS with kernel $3.19.0\text{-}91\text{-}\mathrm{lowlatency}$ is used for both host and guest operating systems. In order to achieve real-time performance, all power-management features in the BIOS, C-states, and CPU frequency scaling have been turned off. The CPU should support the $\mathrm{ssse3}$ and $\mathrm{sse4.1}$ features. These flags must be exposed from the host to the guest, and can be checked by using the command $\mathrm{cat/proc/cpuinfo|grep\ flags|uniq}$. For the physical sever hosting the BBU, we use a Dell Precision T5810 workstation with Intel Xeon CPU E5-1650, 12-core at 3.5 GHz, and 32 GB RAM. There are several configurations that depend on the guest OS's specific setup that should be calibrated in order to boost the performance of the testbed. Most importantly, the maximum transmit power at the eNB and the UE can be calibrated as follows.

- **eNB:** The maximum transmit power at the eNB is signaled to the UE so that it can do its power control. The parameter is PDSCH Energy Per Resource Element (EPRE) $[\mathrm{dBm}]$ and is part of the configuration file, $\mathrm{pdsch\_referenceSignalPower}$. It should be measured using a vector signal analyzer with LTE option for the utilized frequency and then put in the configuration file.
- **UE:** At the UE, the maximum transmit power $[\mathrm{dBm}]$ is measured over the whole (usable) bandwidth. If the same hardware is used at the UE and at the eNB, the power is $\mathrm{max\_ue\_power} = \mathrm{PDSCH\_EPRE} + 10 \log_{10}(12\mathrm{N\_PRB})$.

2) Monitoring the OAI eNB and the UE: As illustrated in Fig. 3(b), our C-RAN experimental testbed consists of one unit of UE and one unit of eNB, both implemented using USRP SDR B210 boards and running on OAI. The OAI software instances of the eNB and UE run in separate Linux-based Intel x86-64 machines comprising of 4 cores for UE and 12 cores for eNB, respectively, with Intel i7 processor core at 3.6 GHz. The characteristics of the OAI software protocol stack are listed as follows:

- L1/L2 Implementation, Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP), GPRS Tunneling Protocol (GTP), and Radio Resource Control (RRC).
- Third Generation Partnership Project (3GPP) LTE MAC/PHY implementation.
- Frequency Division Duplex (FDD) and Time Division Duplex (TDD) modes supports.
- Built-in emulator and simulator.

OAI is supplied with useful monitoring tools such as network protocol analyzers, loggers, performance profilers,



(a) (b) (c)

Fig. 3. (a) Logical illustration of C-RAN testbed architecture; (b) C-RAN testbed implementation utilizing OAI; and (c) Configuration of the eNB-UE connection in an interference-free channel.

timing analyzers, and command line interfaces for performing the intended measurements and monitoring of the network. Specifically, the supported monitoring tools include:

- OAI Soft Scope, which monitors received-transmitted waveforms and also tracks the channel impulse response.
- WireShark Interface and ITTI Analyzer, which analyzes the exchanges between eNB and UE protocols.
- OpenAirInterface performance profiler, which is used for processing-time measurements.

We summarize the testbed configuration parameters in Table II. In particular, the eNB is configured in band 7 (FDD) using a DownLink (DL) carrier frequency of 2.66 GHz. The transmission bandwidth can be set to 5, 10, and 20 MHz, corresponding to 25, 50, and 100 PRBs, respectively. In order to determine the successful connection between eNB and UE, the RRC states should be observed in OAI software. Specifically, when the UE is successfully paired to the eNB, the RRC connection setup message can be seen in the OAI logger. Procedure 1 illustrates the OAI processing flow for building, running, and monitoring stages.

Procedure 1 OAI Setup Processing

Initialization: OAI installation, Kernel setup, CPU setting, USRP B210 configuration.

Output: OAI monitoring tools, OAI Soft Scope, Wireshark/PCAP interface, OAI timing analyzer, OAI message sequence Chart.

```
repeat
    Configure CPU
    Run eNB
    Run UE
    if RRC-RECONFIGURED message at UE is observed then
        Observe Ping and Iperf Tests
    end if
until Testbed Stability
```

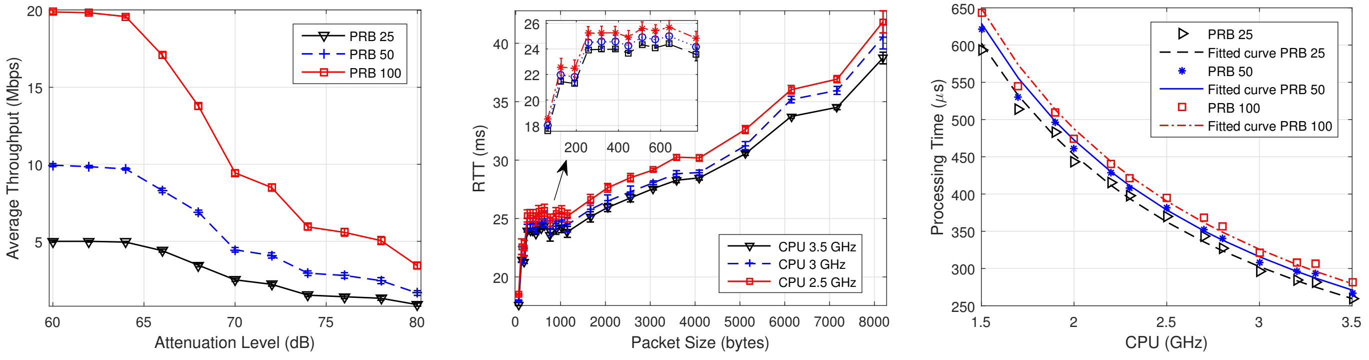
TABLE II
TESTBED CONFIGURATION PARAMETERS FOR eNB AND UE.

Duplexing mode	FDD	**Mobility**	Static
Frequency	2.66 GHz	**PRB**	$25, 50, 100$
Transm. power	$[150 \div 170]$ dBm	**Rad. pattern**	Isotropic
MCS	$[0 \div 27]$	**VM**	VMware

3) Interference-free Testbed Environment: We set up the experiment environment to emulate a "quiet" transmission between the eNB and UE in which there is no interference from other devices (so to have control of the environment). To accomplish this, we use two configurable attenuators, model name Trilithic Asia 35110D-SMA-R, which connect the Tx and Rx ports of the eNB to the Rx and Tx ports of the UE, respectively. Figure 3(c) shows the configuration of the eNB-UE connection in the interference-free channel. In order to establish a stable connection, the transmitter and received gains in the downlink have been selected to be 90 and 125 dB, respectively.

We use iperf to generate 500 packets to send from the eNB to the UE. Figure 4(a) illustrates the throughput performance versus the attenuation level between the eNB and UE. The attenuation was varied between 60 and 80 dB. We observe that the connection will fail when the attenuation level goes beyond 80 dB. The results in Fig. 4(a) show that when the attenuation level is 60 dB the achievable throughputs are around 5, 10, and 20 Mbps when using 25, 50, and 100 PRBs, respectively. On the other hand, at an attenuation of 80 dB, the throughputs are much lower, i.e., 0.98, 1.64, and 3.40 Mbps, respectively.

4) Delay Performance: To test the delay in the C-RAN testbed, we focus on measuring the RTT when sending packets between the eNB and the UE. The VM hosting the BBU is configured with 4 virtual cores and 8 GB RAM in a VMware hypervisor, running on a physical machine with 12 cores, 3.5 GHz CPU, and 16 GB RAM. The OAI UE runs on a low-latency Ubuntu physical machine with 3.0 GHz CPU and 8 GB RAM. Figure 4(b) illustrates the relationship between RTT and packet size when the BBU is set at different CPU frequencies. For each experiment, we sent 500 Internet Control Message Protocol (ICMP) echo request packets from the eNB to the UE. It can be seen that the RTT exponentially increases as the packet size increases. Moreover, we have also noted that the RTT is greater when OAI eNB runs on a VM than on a physical machine, which may be due to the overhead incurred when running the VM. In addition, there is a correlation be-



(a) (b) (c)

Fig. 4. (a) Downlink throughput performance at different attenuation levels; (b) RTT measurement for different packet sizes; and (c) Processing time of LTE subframes against CPU frequency with $\mathrm{MCS} = 27$ and various PRB allocations.

tween the CPU frequency and the OAI software performance. We have recorded that the minimum CPU threshold frequency to run OAI in our scenario is 2.5 GHz. Below the threshold value, we observed that the synchronization between eNB and UE is occasionally missed. By controlling the CPU frequency using the Cpupower tool, we have noticed that the RTT can be improved by increasing the CPU frequency steps.

5) Processing Time of LTE Subframes: We study here the BBU processing time of each LTE subframe with respect to different CPU frequency configurations in the VMware environment. The execution time of each signal processing module in the downlink is measured using *timestamps* at the beginning and at the end of each subframe. OAI uses the RDTSC instruction implemented on all x86 and x64 processors as of the Pentium processors to achieve precise timestamps [37]. The cpupower tool in Linux is used to control the available CPU frequencies. To avoid significant delay and to not miss the synchronization between eNB and UE hardware, we recommend to run the experiment within a $2.8 \div 3.5$ GHz CPU frequency range.

In Fig. 4(c), we depict the processing time of the eNB given different CPU-frequency steps, in which the MCS index is set to 27 for both UL and DL, and observed that the processing time dramatically decreases when the CPU frequency increases. To model the subframe processing time against the CPU frequency and radio-resource configuration, we repeat the experiment in Fig. 4(c) with different MCS indexes. The subframe processing time $T_{\mathrm{sub}}\,[\mu s]$ can be well fitted as a function of CPU frequency, MCS, and PRB as,

$$T_{\mathrm{sub}}\,[\mu s] = \frac{\alpha_{\mathrm{PRB}}}{f^{CPS}} + \beta_{\mathrm{MCS}} + 2.508, \tag{26}$$

where f^{CPS} [Hz] is the CPU frequency, and α_{PRB} and β_{MCS} are two parameters that increase with PRB and MCS values as reported in Table III.

According to (26), we can derive the computation frequency that appears in (4) as,

$$f^{\mathrm{CPS}} = \frac{\alpha_{\mathrm{PRB}}}{T_{\mathrm{sub}} - \beta_{\mathrm{MCS}} - 2.508} \tag{27}$$

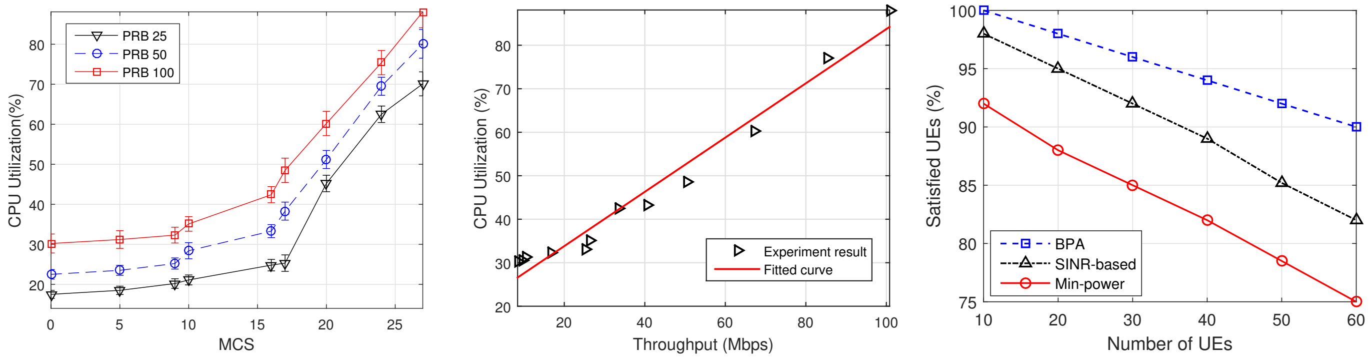
6) CPU Utilization: In C-RAN, it is of critical importance to understand the CPU utilization of the BBU in order to design efficient resource provisioning and allocation schemes. In the previous subsections, we have seen the relationship between MCS and CPU usage for different values of PRBs. In this experiment, the CPU utilization percentage is calculated using the top command in Linux, which is widely used to display processor activities as well as various tasks managed by the kernel in real time. We repeatedly send UDP traffic from the eNB to the UE with various MCS and PRB settings. The CPU utilization percentage has been recorded as in Fig. 5(a). By setting the CPU frequency of the OAI eNB to 3.5 GHz, we have seen that the highest CPU consumption occurred at MCS 27, which corresponded to 72%, 80%, and 88% when PRBs are 25, 50, and 100, respectively. We can conclude that the total processing time and computing resources were mainly spent on the modulation, demodulation, coding, and decoding. These tasks played the bigger roles in terms of complexity and runtime overhead in the BBU protocol stack.

TABLE III
VALUES OF PARAMETERS α_{PRB} AND β_{MCS}.

PRB	25	50	100
$\alpha_{\mathrm{PRB}}[\mu s]$	900	940	970

MCS	0	9	10	16	17	24	27
$\beta_{\mathrm{MCS}}[\mu s]$	0	9.7	11.8	37.5	39.7	64.8	75

To understand better the BBU computational consumption in C-RAN with respect to the users' traffic demand, we will now establish the relationship between the DL throughput and the percentage of CPU usage at the BBU. To begin, we learn that OAI supports 28 different MCSs with index ranging from 0 to 27. In the downlink direction, MCSs with the index 0 to 9 are modulated using QPSK, index 10 to 16 are modulated using 16-QAM, and the rest are based on 64-QAM. For instance, in LTE FDD system with PRB 100, corresponding to bandwidth of 20 MHz, we can get $12 \times 7 \times 2 = 168$ symbols per ms, in case of normal Cyclic Prefix (CP) [45]. Therefore, there are $16,800,000$ symbols per s, which is equivalent to a data rate of 16.8 Mbps. Based on the MCS index used in each experiment, we can calculate the corresponding DL throughput by multiplying the bit rate by the number of bits in the modulation scheme.



(a) (b) (c)

Fig. 5. (a) CPU utilization of the BBU at different values of MCS and PRB; (b) Percentage of CPU usage versus the downlink throughput; and (c) Percentage of satisfied UEs for the different algorithms.

Figure 5(b) shows the CPU utilization percentage at the BBU corresponding to different DL throughputs. Using the calculated results, we have fitted the CPU utilization as a linear function of the DL throughput as,

$$\mathrm{CPU}\,[\%] = 0.6237\phi + 21.3544, \tag{28}$$

where ϕ is the throughput measured in Mbps.

B. Numerical Simulations

We present now simulation results to evaluate the performance of our proposed solutions to the two subproblems discussed earlier: (i) bandwidth and power allocation and (ii) BBU power allocation. The simulations are carried out using a MATLAB implementation with optimization solvers (MOSEK).

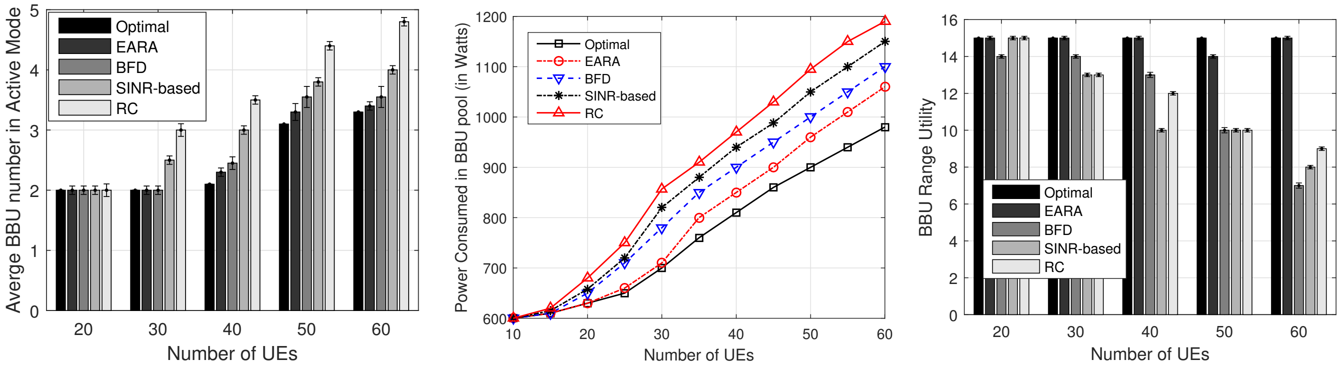
1) Simulation Setup: We consider a C-RAN system consisting of multiple hexagonal cells with a RRH in the center of each cell. The neighboring RRHs are 1 Km apart from each other. We assume that all the wireless channels in the system experience *block fading* such that the channel coefficients stay constant during scheduling interval but can vary from interval to interval, i.e., the *channel coherence time* is not shorter than the scheduling interval. We assume that all the RRHs have the same number of transmit antenna $A = 2$ and maximum transmit power $g_j^{max} = 20$ W, $\forall j$, while the desired data rate $r_i^{min} = 5$ Mbps, $\forall i$. We adopt the distance-dependent path-loss model given as L_p $[\mathrm{dB}] = 148.1 + 37.6 \log_{10} \mathrm{d}_{[\mathrm{Km}]}$, and the log-normal shadowing variance set to 8 dB. In addition, the wireless transmission bandwidth B_j^{max} is set to 10 MHz and the noise power is set to -100 dBm.

2) Performance of BPA Algorithm: For comparison, we introduce two user-association schemes that have been discussed in the literature: SINR-based scheme [46] and Min-power scheme [47]. The SINR-based simply assumes that each UE only associates with one RRH that provides the best channel gain. The Min-power scheme differs from our proposed algorithm, BPA, in the local search procedure. It is designed to minimize the total consumption power of the network without considering the bandwidth constraint, where UEs are associated with the nearest RRHs and the BBU pool allocates the same computing power to all the UEs. Figure 5(c) illustrates the percentage of satisfied users versus different number of users for the three algorithms. It is shown that the BPA algorithm has better performance compared against SNIR-based and Min-power schemes; it is also observed that the percentage of satisfied users decreases while the number of users increases for three algorithm.

3) Resource Allocation Results and Discussions: We evaluate the performance of our EARA scheme through numerical simulations. The number of BBUs in the cloud is $K = 5$, and the maximum number of VMs in each BBU is set to 15.

- *Optimal Algorithm:* The optimal bin-packing-based BBU scheduling algorithm needs to traverse all feasible solutions, and then chooses the solution with minimum number of BBUs in working mode. Since bin-packing problem is a typical NP-hard problem, the complexity of the algorithm is $\mathcal{O}\left(2^{N_j}\right)$, where N_j represents the number of user associated with RRH j.
- *BFD Algorithm:* The BFD is a bin-packing approximate algorithm that aims at finding a set of users that fits the BBUs capacity. It has lower complexity, $\mathcal{O}\left(N_j \log N_j\right)$, than other existing algorithms such as Next Fit and First Fit Decreasing [48].
- *RRH-Clustering (RC) Algorithm:* It is proposed in [43] to assign each RRH to BBUs. Compared to the previous BBU scheduling algorithms (e.g., BFD and NFB), it reduces the complexity with however a loss of performance. In each iteration, the algorithm assigns the cluster with maximum number of UEs and the cluster with minimum number of UEs to one BBU. The complexity of this algorithm is $\mathcal{O}(N_j)$.
- *SINR-based Algorithm:* In this algorithm [46], the users are associated with a RRH that provides the maximum SINR; in other words, UE i, $\forall i \in \mathcal{N}$ is assigned to RRH $j = \arg\max_{j\in\mathcal{L}} g_j^{\max} H_{ij}$. The complexity of this algorithm is $\mathcal{O}(N_j \log N_j)$.

Figure 6(a) depicts the average number of BBUs in active mode under different traffic loads in the network. It is obvious that, when the number of UEs in the cell is small, all the algorithms—Optimal, EARA, BFD, SINR-based, and RC—have the same performance. That is because one BBU can support all users in the cell that has the number of UEs less



(a) (b) (c)

Fig. 6. (a) Average number of BBUs in active/working mode under different numbers of UEs; (b) BBU pool power consumption under different UE number with $K = 5$, $U = 15$, $\mathcal{E}_k^{bbu} = 200$W in active mode and 100W in sleep mode; and (c) BBU pool range under different number of UEs.

than or equal to the capacity of a single BBU. As the number of UEs increases, the performance difference among the BBU scheduling algorithms becomes clearer and clearer. However, except for the optimal algorithm, our algorithm shows better performance compared to other competing ones, as it chooses the best fit BBU for those cases with lowest capacity loss.

Figure 6(b) illustrates the energy consumption in BBU pool with different number of UEs. Since the BBU pool carries the baseband signal processing, a large number of UEs lead to heavy computing workloads in the pool. Therefore, the consumption of the cloud platform, where the BBU pool is implemented, increases significantly with the increase of the number of UEs. This indicates that the energy consumption of the BBU pool is closely related to the network traffic load.

In Fig. 6(c), we define the BBU range utility metric as a number calculated by subtracting the maximum number of active VMs in the BBU k, $\forall k \in \mathcal{K}$ from the minimum number of active VMs in the BBU m, $\forall m \in \mathcal{K}$ under different numbers of UEs. It can be seen from the plot that all evaluated algorithms have the same performance at low traffic load; however, except for the optimal algorithm, our proposed algorithm shows better performance compare with the others.

VI. Conclusions

We proposed a novel resource-allocation scheme that optimizes the energy consumption of a Cloud Radio Access Network (C-RAN), one of the key technologies towards 5G wireless cellular networks. An energy consumption model that characterizes the computation energy of the Base Band Unit (BBU) pool is proposed based on empirical results collected from our programmable C-RAN testbed. Then, the resource allocation problem is decomposed into two subproblems: the Bandwidth Power Allocation (BPA) problem, cast via Mixed-Integer Nonlinear Programming (MINLP), that aims at assigning a feasible bandwidth and power allocation to serve all users with QoS requirements; and the BBU Energy-Aware Resource Allocation (EARA) problem, cast as a bin-packing problem, that aims at minimizing the number of active Virtual Machines (VMs) in the BBU pool to increase energy saving. We addressed the BPA problem by transforming it into a convex problem and proposed a novel heuristic algorithm to the BBU EARA problem based on the Best Fit Decreasing (BFD) method. Testbed experiments were carried out to evaluate the BBU performance under various computing and radio-resource configurations. Experimental results showed that the frame processing time and CPU utilization of the BBU increase with the Modulation and Coding Scheme (MCS) index and with the number of allocated Physical Resource Blocks (PRBs). Additionally, simulation results were presented to evaluate the performance of our two proposed algorithms, BPA and EARA, and their improvement over existing algorithms under a variety of network conditions.

Future Work: Our testbed is a first step towards the realization of a real-time operating system and virtualization environment in C-RAN. The OAI testbed can be built in different environments such as linux container LXC and Docker. This procedure will help give better understanding of the OAI's performance inside different virtualization environments. Another line of investigation is on IaaS, where virtualized BBU resources can be adjusted on-demand by a hypervisor. In our case, we can create more than one OAI VM working on top of a single physical computer implemented by using a VMware hypervisor that manages several virtual servers. This approach will be a practical solution for testing parallel OAI eNBs on one hypervisor.

Appendix

Proof of Lemma 1: We rewrite the objective of $\mathcal{P}3$ as,

$$f\left(B_{ij}\right) = \sum\nolimits_{i \in \mathcal{N}_j} \frac{N_0 B_{ij}}{h_{ij}} \left(2^{\frac{r_i^{\min}}{B_{ij}}} - 1 \right), \forall j \in \mathcal{L}. \qquad (29)$$

The objective function of (29) is convex *if and only if* its Hessian matrix is positive semi-definite [42]. In our case, the Hessian matrix can be calculated as,

$$H = \begin{bmatrix} \frac{\partial f}{\partial(B_1)\partial(B_1)} & \frac{\partial f}{\partial(B_1)\partial(B_2)} & \cdots & \frac{\partial f}{\partial(B_1)\partial(B_N)} \\ : & : & : & : \\ \frac{\partial f}{\partial(B_L)\partial(B_1)} & \frac{\partial f}{\partial(B_L)\partial(B_2)} & \cdots & \frac{\partial f}{\partial(B_L)\partial(B_N)} \end{bmatrix},$$

$$\frac{\partial f}{\partial(B_j)\partial(B_i)} = \begin{cases} 0, & \forall i \neq j, \\ \frac{N_0 (r_i^{\min})^2 (\ln 2)^2}{h_{ij}(B_{ij})^3} 2^{\frac{r_i^{\min}}{B_{ij}}} > 0, \forall B_{ij} > 0, & \forall i = j \end{cases} \tag{30}$$

It can be seen that the Hessian matrix H is a diagonal matrix where all the diagonal entries are positive. Hence, H is positive semidefinite and thus $\mathcal{P}3$'s objective function is convex. Moreover, since constraints (19b) and (19b) are affine, we can state that problem $\mathcal{P}3$ is convex. The proof is complete.

References

[1] T. X. Tran, A. Younis, and D. Pompili, “Understanding the computational requirements of virtualized baseband units using a programmable cloud radio access network testbed,” in *Proc. IEEE ICAC*, pp. 221–226, 2017.

[2] China Mobile Research Institute, “C-RAN: The road towards green RAN,” *White Paper*, Sept. 2013.

[3] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, “Cloud radio access network (C-RAN): A primer,” *IEEE Network*, vol. 29, no. 1, pp. 35–41, 2015.

[4] T. X. Tran and D. Pompili, “Dynamic radio cooperation for user-centric cloud-RAN with computing resource sharing,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2379–2393, 2017.

[5] P. Luong, F. Gagnon, C. Despins, and L.-N. Tran, “Joint virtual computing and radio resource allocation in limited fronthaul green C-RANs,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2602–2617, 2018.

[6] T. X. Tran and D. Pompili, “Dynamic radio cooperation for downlink cloud-RANs with computing resource sharing,” in *Proc. IEEE MASS*, pp. 118–126, 2015.

[7] P. Luong, F. Gagnon, C. Despins, and L.-N. Tran, “Optimal joint remote radio head selection and beamforming design for limited fronthaul C-RAN,” *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5605–5620, 2017.

[8] D. Pompili, A. Hajisami, and T. X. Tran, “Elastic resource utilization framework for high capacity and energy efficiency in cloud RAN,” *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 26–32, 2016.

[9] A. Younis, T. X. Tran, and D. Pompili, “Fronthaul-aware Resource Allocation for Energy Efficiency Maximization in C-RANs,” in *Proc. IEEE ICAC 2018*.

[10] T. X. Tran and D. Pompili, “Octopus: A Cooperative Hierarchical Caching Strategy for Cloud Radio Access Networks,” in *IEEE MASS*, pp. 154–162, 2016.

[11] T. X. Tran, F. Kazemi, E. Karimi, and D. Pompili, “Mobee: Mobility-Aware Energy-Efficient Coded Caching in Cloud Radio Access Networks,” in *Proc. IEEE MASS*, pp. 461–465, 2017.

[12] T. X. Tran, D. V. Le, G. Yue, and D. Pompili, “Cooperative hierarchical caching and request scheduling in a cloud radio access network,” *IEEE Trans. Mobile Computing*, 2018.

[13] ORACLE Cloud, “Enterprise-grade cloud solutions: SaaS, PaaS, and IaaS,” 2017.

[14] V. N. Ha, L. B. Le, and D. Ngc-Dung, “Coordinated multipoint transmission design for cloud-RANs with limited fronthaul capacity constraints,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7432–7447, 2016.

[15] Y. Shi, J. Zhang, and K. B. Letaief, “Group sparse beamforming for green cloud-RAN,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2809–2823, 2014.

[16] S. Luo, R. Zhang, and T. J. Lim, “Downlink and uplink energy minimization through user association and beamforming in C-RAN,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 494–508, 2015.

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

[18] H. Wu, Y. Sun, and K. Wolter, “Energy-efficient decision making for mobile cloud offloading,” *IEEE Trans. Cloud Computing*, 2018.

[19] H. Wu, “Stochastic analysis of delayed mobile offloading in heterogeneous networks,” *IEEE Trans. Mobile Comput.*, vol. 17, no. 2, pp. 461–474, 2018.

[20] J. Li, M. Peng, A. Cheng, Y. Yu, and C. Wang, “Resource allocation optimization for delay-sensitive traffic in fronthaul constrained cloud radio access networks,” *IEEE Syst. J.*, vol. 11, no. 4, pp. 2267–2278, 2017.

[21] T. X. Vu, H. D. Nguyen, and T. Q. Quek, “Adaptive compression and joint detection for fronthaul uplinks in cloud radio access networks,” *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4565–4575, 2015.

[22] S.-H. Park, O. Simeone, O. Sahin, and S. Shamai, “Robust and efficient distributed compression for cloud radio access networks,” *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 692–703, 2013.

[23] T. X. Vu, H. D. Nguyen, T. Q. Quek, and S. Sun, “Adaptive cloud radio access networks: Compression and optimization,” *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 228–241, 2017.

[24] R. Schooler, “Transforming networks with NFV and SDN,” *Intel Architecture Group*, 2013.

[25] AMARISOFT, “Amarisoft LTE software base station.” Available: https://www.amarisoft.com/?p=amarilte, 2018.

[26] EURECOM, “OAI.” Available: http://www.openairinterface.org/, 2018.

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

[28] C.-N. Mao, M.-H. Huang, S. Padhy, S.-T. Wang, W.-C. Chung, Y.-C. Chung, and C.-H. Hsu, “Minimizing latency of real-time container cloud for software radio access networks,” in *Proc. IEEE CloudCom*, pp. 611–616, 2015.

[29] Z. Kong, J. Gong, C.-Z. Xu, K. Wang, and J. Rao, “ebase: A baseband unit cluster testbed to improve energy-efficiency for cloud radio access network,” in *Proc. IEEE ICC*, pp. 4222–4227, 2013.

[30] W. Wu, L. E. Li, A. Panda, and S. Shenker, “PRAN: Programmable radio access networks,” in *Proc. ACM Workshop HotNets*, 2014.

[31] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, “Cloud RAN for mobile networks—A technology overview,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 2015.

[32] T. Sigwele, A. S. Alam, P. Pillai, and Y. F. Hu, “Energy-efficient cloud radio access networks by cloud based workload consolidation for 5G,” *J. Netw. Comput. Appl.*, vol. 78, pp. 1–8, 2017.

[33] P. Rost, C. J. Bernardos, A. De Domenico, M. Di Girolamo, M. Lalam, A. Maeder, D. Sabella, and D. Wübben, “Cloud technologies for flexible 5G radio access networks,” *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 68–76, 2014.

[34] D. A. Burgess and H. S. Samra, “The OpenBTS project.” Available: http://openBTS. org, 2008.

[35] A. Davydov, G. Morozov, I. Bolotin, and A. Papathanassiou, “Evaluation of joint transmission CoMP in C-RAN based LTE-A HetNets with large coordination areas,” in *Proc. IEEE GC Wkshps*, pp. 801–806, 2013.

[36] K. I. Pedersen, Y. Wang, S. Strzyz, and F. Frederiksen, “Enhanced inter-cell interference coordination in co-channel multi-layer LTE-advanced networks,” *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 120–127, 2013.

[37] 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. IEEE ICC*, pp. 5523–5528, 2015.

[38] D. Sabella, A. De Domenico, E. Katranaras, M. A. Imran, M. Di Girolamo, U. Salim, M. Lalam, K. Samdanis, and A. Maeder, “Energy efficiency benefits of RAN-as-a-service concept for a cloud-based 5G mobile network infrastructure,” *IEEE Access*, vol. 2, pp. 1586–1597, 2014.

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

[40] A. R. Dhaini, P.-H. Ho, G. Shen, and B. Shihada, “Energy efficiency in tdma-based next-generation passive optical access networks,” *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 850–863, 2014.

[41] J. Tang, W. P. Tay, T. Q. Quek, and B. Liang, “System cost minimization in cloud ran with limited fronthaul capacity,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3371–3384, 2017.

[42] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[43] K. Boulos, M. El Helou, and S. Lahoud, “RRH clustering in cloud radio access networks,” in *Proc. IEEE ICAR*, pp. 1–6, 2015.

[44] K. Wang, W. Zhou, and S. Mao, “On joint BBU/RRH resource allocation in heterogeneous cloud-RANs,” *IEEE Internet of Things J.*, vol. 4, no. 3, pp. 749–759, 2017.

[45] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-advanced for mobile broadband*. Academic press, 2013.

[46] L. Liu and R. Zhang, “Downlink SINR balancing in C-RAN under limited fronthaul capacity,” in *Proc. IEEE ICASSP*, pp. 3506–3510, 2016.

[47] R. Zhang and S. Cui, “Cooperative interference management with MISO beamforming,” *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5450–5458, 2010.

[48] T. Sigwele, A. S. Alam, P. Pillai, and Y. F. Hu, “Evaluating energy-efficient cloud radio access networks for 5G,” in *Proc. IEEE DSDIS*, pp. 362–367, 2016.

Ayman Younis received the B.Eng. and M.Sc. degrees in Electrical Engineering from the University of Basrah, Iraq, in 2008 and 2011, respectively. He is currently pursuing the Ph.D. degree in Electrical and Computer Engineering (ECE) with Rutgers University, New Brunswick, NJ, USA under the guidance of Dr. Dario Pompili. His research interests include wireless communications and mobile cloud computing, with emphasis on software-defined radio testbed, cloud radio access networks and mobile-edge computing.

Dr. Tuyen X. Tran received his PhD degree in Electrical and Computer Engineering (ECE) from Rutgers University, NJ, USA, in May 2018. He had previously received the MSc degree in ECE from the University of Akron, OH, USA, in 2013, and the BEng degree (Honors Program) in Electronics and Telecommunications from Hanoi University of Science and Technology, Vietnam, in 2011. His research interests lie in the area of wireless communications, mobile cloud computing, and network optimization. He held research internships at Huawei Technologies R&D Center, Bridgewater, NJ, during the summers of 2015-2017. He received the Best Paper Award at the IEEE/IFIP Wireless On-demand Network systems and Services Conference (WONS) in 2017 and the Outstanding Graduate Student Award from the Rutgers School of Engineering in 2018.

Dr. Dario Pompili is an associate professor with the Dept. of ECE at Rutgers U., where he directs the Cyber-Physical Systems Laboratory (CPS Lab), which focuses on mobile computing, wireless communications and networking, acoustic communications, sensor networks, and datacenter management. He received his PhD in ECE from the Georgia Institute of Technology in 2007. He had previously received his ‘Laurea’ (combined BS and MS) and Doctorate degrees in Telecommunications and System Engineering from the U. of Rome “La Sapienza,” Italy, in 2001 and 2004, respectively. He is a recipient of the NSF CAREER’11, ONR Young Investigator Program’12, and DARPA Young Faculty’12 awards. In 2015, he was nominated Rutgers-New Brunswick Chancellor’s Scholar. He published more than 150 refereed scholar publications: with more than 8,000 citations, Dr. Pompili has an h-index of 32 and a i10-index of 67 (Google Scholar, June’18). He is a Senior Member of the IEEE Communications Society and of the ACM.