RESOURCE ALLOCATION AND COMPUTATION OFFLOADING IN CLOUD-ASSISTED WIRELESS NETWORKS

By

AYMAN NASIH SALMAN YOUNIS

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Professor Dario Pompili

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2022

ABSTRACT OF THE DISSERTATION

Resource Allocation and Computation Offloading in Cloud-assisted Wireless Networks

By Ayman Nasih Salman Younis

Dissertation Director:
Professor Dario Pompili

The next radio generations of mobile networks, Fifth Generation (5G) and beyond, are expected to speed up the transition from monolithic and inflexible networks to agile and distributed networking elements that rely on "virtualization", "softwarization", openness, intelligent and yet fully interoperable Radio Access Network (RAN) components. At the same time, the increasing popularity of computation-intensive applications on mobile devices has contributed to the overwhelming mobile traffic volume that is pushing against the boundary of the current communication networks' capacity. Besides, mobile platforms are becoming the predominant medium of access to Internet services due to a tremendous increase in their computation and communication capabilities.

In light of this, cloud-assisted wireless networks are promising solutions that unite wireless networks and cloud computing to deliver cloud services directly from the network edges. The three emerging paradigms for cloud-assisted wireless networks are: Cloud Radio Access Network (C-RAN), in which the baseband resources are pooled at a Base Band Unit (BBU) based on the fundamentals of centralization and virtualization; Mobile-Edge Computing (MEC), which provides cloud computing and storage capabilities to enable rich services and applications in close proximity to the end-users; and Next Generation RAN (NG-RAN), in which the functional splitting technique is utilized to flexibly balance the radio and computation processes at Central Units (CUs) and Distributed Units (DUs).

These paradigms are complementary and have unique justifications within the 5G and beyond ecosystem. The centralized nature of C-RAN provides a higher degree of cooperation in the network to address the capacity fluctuation and to increase the spectral and energy efficiency, whereas the MEC paradigm is useful in reducing service latency and improving localized user experience; on the other hand, NG-RAN provides flexible distribution of computation and radio capabilities at Base Stations (BSs).

The goal of this research is to leverage the emerging C-RAN, MEC, and NG-RAN paradigms and to design disruptive innovations for wireless access networks in such a way as to always make the best use of the resources available so as to satisfy the service requests from the end-users. To this end, novel resource-allocation schemes and computation-offloading policies are proposed in this thesis to minimize the service latency and to improve the users' Quality of Experience (QoE). The proposed innovative solutions include: (i) a novel resource-allocation solution that aims at optimizing the energy consumption of a C-RAN, (ii) a novel resource-allocation scheme that aims at maximizing the network energy efficiency of a C-RAN subject to practical constraints including Quality of Service (QoS) requirement, transmission power, and fronthaul capacity, (iii) a dynamic Video-streaming QoE Maximization (VQM) that takes into account the video Distortion Rate (DR) and the coordination among MEC server in order to enhance Adaptive Bitrate (ABR)-video streaming in a MEC network, (iv) a joint task offloading, latency, and Quality Loss of Result (QLR) framework, which helps improve users' computation experience by offloading their computation tasks to the edge servers, and (v) a novel Deep Reinforcement Learning-based Resource Allocation (ReLAX) framework to deal with the joint optimization of end-user association and power allocation in NG-RAN systems. The proposed innovations in this research can benefit a wide range of mobile applications and services such as video streaming, augmented reality (AR)/virtual reality (VR), Internet-of-Things (IoTs), and public safety operations.

Acknowledgements

I would like to thank my adviser, Dr. Dario Pompili, for constantly supporting me throughout my doctoral studies and for always leading me forward both academically and research wise. He has taught me to define interesting and relevant research problems, to look at them from multiple angles, and to try and find the best possible engineering solutions. He will always have my sincere gratitude and respect for his keen mentoring and expert advice.

I would like to extend my gratitude to Drs. Emina Soljanin, Zoran Gajic, Kristin Dana, Richard P. Martin, Bo Yuan, Saman Zonouz, and Sumit Maheshwari for serving as committee members in my PhD qualifying exam, thesis proposal, and dissertation defense.

I would like to thank all the CPS Lab members and my collaborators—Tuyen X. Tran, Mehdi Rahmati, Abolfazl Hajisami, Brian Qiu, Chuanneng Sun, and Parul Pandey—for the many discussions and encouragement. My sincerest thanks to all the co-authors of my publications and to the many anonymous reviewers of my peer-reviewed papers.

I am grateful for the financial support from the Higher Committee for Education Development in Iraq (HCED), the Department of Electrical and Computer Engineering–Rutgers University, and the US National Science Foundation (NSF) grant (No. ECCS-2030101), which have provided with the necessary resources to carry on my research and write this doctoral dissertation.

Last, but not least, I would like to thank my parents, Mrs. Khairyah Younis and Mr. Nasih Younis, my wife, Adwhaa AL-Chaab, and my daughters, Mila Younis and Layan Younis, for their continued understanding and encouragement. Their unconditional love and support have given me the strength to chase my dreams and aspirations. To them, I dedicate this dissertation.

Dedication

To my parents,

my wife Adhwaa,

and our daughters Mila and Layan

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 5G and Beyond Systems: Key Features and Requirement

The explosive increase in demand for wireless broadband services, such as Internet of Things (IoT), wireless health services, and smart city applications, has placed severe demands on cloud infrastructure and wireless network that need for massive connectivity of devices, and also ultra-low-latency connectivity over Internet Protocol (IP). At the same time, mobile platforms are becoming the predominant medium of access to Internet services due to a tremendous increase in their computation and communication capabilities. To meet these performance criteria, there is ongoing work in many areas of the upcoming Fifth Generation (5G) and beyond wireless systems to realize breakthroughs in the transformation of Information and Communications Technology (ICT). 5G and beyond networks are expected to be heterogeneous networks that involve multiple modes and a unified air interface tailored to the needs of specific applications. Besides, enabling new techniques such as network resource allocations and computing-intensive task offloading framework are expected to be major features of a 5G and beyond network.

Requirements: The rapid growth in wireless data services driven by mobile Internet and computation-intensive devices has triggered the study and consideration of the 5G and beyond cellular network. It is expected that the next generation systems will have to support multimedia and real-time applications with a wide variety of requirements, including user higher capacity and data rates, lower end-to-end latency, massive device connectivity, enhanced Quality of Experience (QoE), and improved energy efficiency. These challenges are briefly discussed as follows.

- *Capacity and Data Rate:* With increased mobile data traffic such as video consumption

and real-time applications, system capacities data rates will be crucial requirements in the 5G era. In general, mobile wireless communication would need a 1000-fold increase in traffic capacity by 2020 relative to 2010 levels, and a 10- to 100-fold increase in data rates specifically at high mobility and crowded areas, with extremely height peak data rates of 10 Gbit/s [3].

- *End-to-end Latency:* With emerging several real-time applications such as Vehicle-to-Vehicle (V2V) systems, and Augmented and Virtual Reality (AR and VR) applications, the system latency factor would be an important factor in designing such latency-critical applications. For example, traffic safety applications for cars and humans, built around V2V and vehicle-to-Infrastructure (V2I) communication, require very fast request-response and feedback control cycles with high availability and reliability. In order to realize these applications, wireless networks are expected to support a target of 1 ms end-to-end latency with high reliability [3].

- *Connection Density:* Providing the possibility for the massive number of connected devices and sensors communicating with each other will be the key driver by upcoming generation infrastructure. These will range from devices with limited resources that require only intermittent connectivity for reporting (e.g., sensors) to devices that require always-on connectivity for monitoring and/or tracking (e.g., traffic safety and control, monitor and control of infrastructure). Hence, a challenge for 5G and beyond wireless access is to support the diversity of devices and service requirements in a scalable and efficient manner.

- *QoE:* Another significant metric required in the context of next generation is QoE, which describes the subjective perception of the users as to how well applications or services are working. For instance, the QoE of video streaming applications depends on the quality of the encoded and delivered video in the context of the display on which the video is shown. Delivering an application with low QoE leads to user dissatisfaction, whereas high QoE unnecessarily drains resources on both the customer (e.g., device battery, memory, CPU cycle) and operator (e.g., radio and transport network resource, Base Station (BS) power) sides. Hence, a challenge for 5G and

beyond is to support user services with a consistent level of QoE and optimal policy for exploiting mobile device resources.

- *Network Energy Efficiency:* Another key demand required in the 5G and beyond standardization is network energy efficiency. Increased network energy efficiency is one critical factor to reduce the optional cost of a network. Therefore, 5G is expected to significantly improve network energy efficiency by enabling and modifying new 5G architectures and protocols.

Key Enabling Technologies: To maintain the sustainable development of the wireless communication industry, novel solutions should be developed to meet the 5G and beyond requirements. These necessitate disruptive solutions that could lead to both architecture and hardware design changes, as listed in the following [4, 5].

- *Cloud-based Wireless Networks:* To deal with the high growth in cellular data, a large number of small cells (e.g, microcell, picocell, femtocell, relay nodes, Wi-Fi access points) are required to be installed indoors or outdoors, giving rise to Ultra-Dense Networks (UDN), which pave the way for the development of 5G and beyond. With such large-scale UDNs, network providers meet several major challenges in terms of operation and management, network deployment, and inter-cell interference mitigation. To overcome those challenges, cloud-based platforms are emerged to optimize the deployment, operation and management, and facilitate the network's overall performance.

- *Millimeter Wave (mmWave):* The requirement of the 5G wireless communication for high throughput motivates the wireless industry to use the mmWave bands, ranging from 3 to 300 GHz. Many bands therein seem promising, including most immediately the local multipoint distribution service at $28 - 30$ GHz, the license-free band at 60 GHz, and the E-band at $71 - 76$ GHz, $81 - 86$ GHz, and $92 - 95$ GHz. The mmWave are emerged to increase bandwidth, compensate the heavy path loss, and improve the communications capacity. Considering the commercial requirements, 5G mmWave large array systems should be implemented in an energy- and cost-efficient way with a small form factor.

- *Massive Multiple-Input Multiple-Output (MIMO):* Massive MIMO is typically comprised of a few hundred inexpensive antenna components, which can focus transmission energy in certain directions and consequently increase throughput and save energy significantly. Moreover, it can also facilitate concurrent transmissions to serve multiple users at the same time. Massive MIMO may require major architecture changes, practically in the design of macro BSs and it may also lead to new types of deployments.

- *Cognitive Radio (CR):* Spectrum and energy scarcity are two main constraints of wireless communication systems. CR has been presented as a powerful technique to increase spectrum efficiency by enabling unlicensed users to access unused spectrum opportunistically [6]. Two main paradigms to efficiently utilize spectra are spectrum sensing and spectrum database. For the former, unlicensed users sense the spectrum to detect the availability of channels before transmission and access the channels only when idle. For the latter, unlicensed users can acquire the availability of channels through spectrum databases before accessing the channels. Accordingly, work needs to be done on a cross-layer view (e.g, Physical-layer (PHY) and Medium Access Control (MAC) layers) on designing CR networks.

- *Device-to-device Communication (D2D):* As a promising technique, D2D communication has attracted much attention in 5G and beyond systems. With D2D mode, the communication paradigm enables devices in close proximity to communicate with each other directly without sending data to the BS or the core network. Using D2D communication, it can significantly improve spectral efficiency, save transmission power, and reduce network latency.

1.2 Cloud-assisted Wireless Networks

Increasing demand for wireless data traffic and mitigating energy consumption in different types of applications are inevitable for the 5G networks. Therefore, 5G and beyond systems

will imply major developments in the implementation of networking infrastructure. Cloud-assisted wireless networks, Software-Defined Networking (SDN) and Network Functions Virtualization (NFV), are becoming promising solutions in today's business due to the benefits such as greater flexibility, increased security, scalability and low cost. The three emerging paradigms for cloud-assisted wireless networks are Cloud Radio Access Network (C-RAN), which aims at the centralization of the BSs functionalists (e.g, encoding, decoding) via network virtualization and optical fronthaul technologies; Mobile-Edge Computing (MEC), which provides cloud-computing capabilities at the edge of the mobile network, within the RAN and in close proximity to mobile subscribers; and Next-Generation RAN (NG-RAN), which aims to transition from inflexible and monolithic networks to agile and decentralized elements.

1.2.1 Cloud Radio Access Network

The C-RAN is introduced as a cost-efficient potential solution to enhance spectrum efficiency and energy efficiency of wireless networks. In addition, C-RAN has the potential to decrease the cost of network operation, including capital expenditure (CAPEX) and operating expenditure (OPEX), by reducing power and energy consumption in the network. The centralized nature of processing within C-RAN enables flexible management of the spectrum and computing resources as well as real-time collaborative communications among the BBUs. These characteristics bring extra degrees of freedom and facilitate the deployment of advanced cooperative technique as well as resource-allocation mechanisms for improved energy efficiency for both the mobile devices and the cellular systems, and increased spectrum efficiency for the overall systems.

C-RAN Architecture. As illustrated in Fig 1.1, 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 RRHs to the BBU pool. Packet-level processing, MAC, PHY baseband processing, and Radio Frequency (RF) functionalities may be split

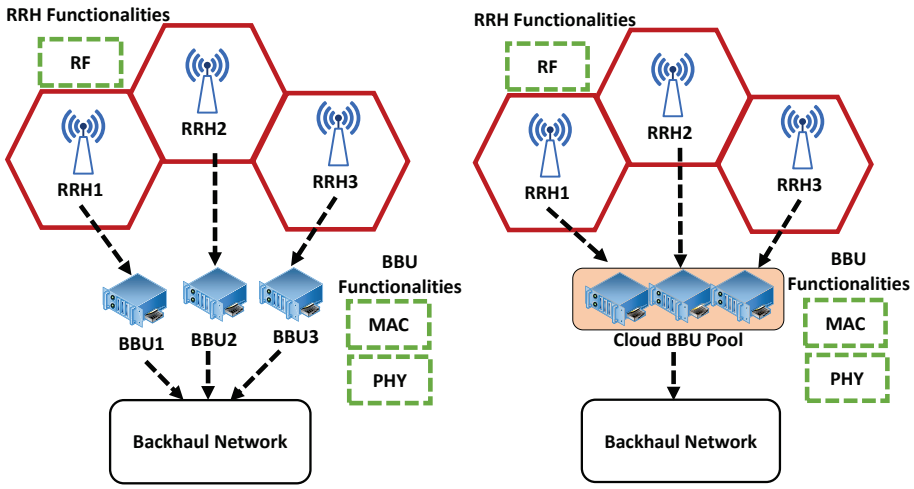


Figure 1.1: (a) Each BBU is assigned to one RRH; and (b) Consolidated BBU.

between the BBU and the RRHs depending on the specific C-RAN implementation [7]. Based on the network performance and system implementation complexity, each BBU can be assigned to one RRH, as shown in Fig. 1.1(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. 1.1(b).

Advantages of C-RAN. The centralized processing in C-RAN permits the implementation of efficient computation and resource allocation algorithms between the RRHs and the BBU pool. It also enable the optimization of the radio access performance at the traffic level, for instance, through joint multi-cell processing and intercell interference coordination (ICIC). Resource allocation and ICIC techniques can significantly enhance wireless network performance by mitigating interference between adjacent BSs. At the network level, centralized processing is demanded to deal with ultra-dense networks (e.g., to dynamically adapt to spatial and temporal fluctuations by turning on/off RRHs) by adding spectrum resources and configuring the network to fine-tune user data traffic delivery. The list of other benefits enabled by C-RAN can be listed as follows [8].

- *Increasing energy efficiency:* With C-RAN architecture, the number of cell sites can be reduced several folds. Thus, the costs of onsite infrastructures, such as air conditioning and other power-consuming equipment, can be significantly reduced. Besides, to improve the overall network energy efficiency, it is much easier to deploy small cells

with lower transmission power.

- *Throughput Improvement:* Because of the pooling of BBU resources in a C-RAN, the mechanisms introduced for LTE-Advanced (LTE-A) to increase spectral efficiency and throughput, such as coordinated Multi-Point (CoMP) and enhanced Inter-Cell Interference Coordination (eICIC) schemes, are greatly facilitated. As signal processing from many cells can be done over one BBU pool in C-RAN, the energy processing and transmitting delays are also reducing by utilizing implementing load balancing methods between the cells, such as adaptive resource allocation and beamforming.

- *Adaptability to Nonuniform Traffic and Scalability:* In each BS, daily traffic distribution generally fluctuates, and the peaks of traffic happen at various hours. In C-RAN, the overall utilization rate can be improved since most baseband processes are executed in the BBU pool. The required baseband processing capacity of the pool is expected to be smaller than the sum of capacities of single base stations. In addition, coverage upgrades demand the connection of extra RRHs to the already existing BBU pool. Therefore, existing cells can then be split, or additional RRHs can be added to the BBU pool to handle non-uniformly distributed traffic.

1.2.2 Mobile-Edge Computing

The recent advances in IoT have enabled a paramount of new applications (e.g., VR, AR, and object tracking and recognition) to provide real-time machine-to-machine and machine-to-human interactions. Such complex applications necessitate higher computing power, memory and battery lifetime on mobile devices. However, due to physical size constraint, mobile devices are generally resource-hungry; in fact, the limited energy supply from battery has been one of the most challenging issues for mobile devices. At the same time, with the development of wireless communication technologies such as Wi-Fi, 4G or even 5G, Mobile Edge Computing (MEC) has emerged as a promising approach to address such a challenge. Specifically, MEC servers are owned by the network operator and are implemented directly at the cellular BSs or at the local wireless access points using a generic-computing platform. With this position, MEC allows for the resource-limited mobile devices to offload

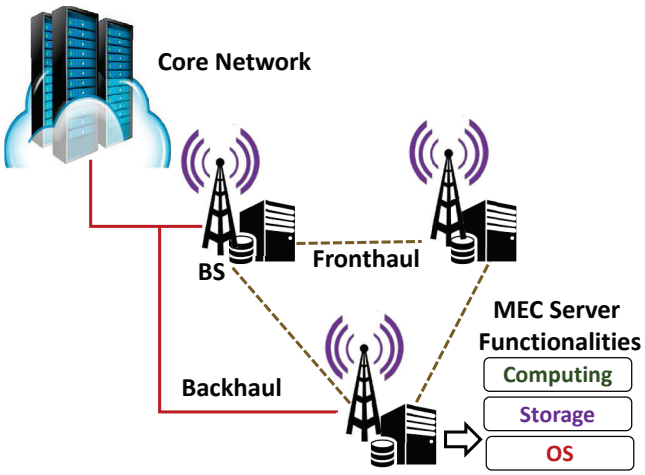


Figure 1.2: Illustration of a MEC network.

their computation tasks to the BSs at which the offloaded tasks can be executed by the co-located MEC servers. This substantially helps reduce the task completion delay and release the burden on the backhaul networks [9]. Additionally, MEC has the potential to empower the network with different benefits, including: (i) optimization of mobile resource by running compute-intensive applications at the network edge, (ii) context awareness support with providing the RAN information such as cell load, user mobility and location, and (iii) QoE's user improvement by enabling various techniques at the network edge such as video caching and high throughput browsing. Figure 1.2 represents a typical scenario of applying MEC technique for mobile networks, in which the computation-intensive applications can be offloaded from mobile user devices to the MEC servers. Recently, several MEC concepts with the purpose of smoothly integrating cloud capabilities into the mobile network architecture have been proposed in the literature. Some of these network architectures are; Small Cell Cloud Cloud (SCC) [10], Fast Moving Personal Cloud (MobiScud) [11], and European Telecommunications Standards Institute (ETSI) MEC [12, 13].

1.2.3 Next-Generation Radio Access Network

To support the diverse requirements of richer, more demanding applications, NG-RANs will need to lever- age a novel architecture to transition from inflexible and monolithic networks to agile and decentralized elements. This architecture will enable new networking functionalities to: (i) provide on-demand virtual and RAN functional splitting network

options in terms of software and hardware environments; (ii) manage the network physical infrastructure in near real time via open, intelligent, virtualized software interfaces; and (iii) enable cloud services, including cloud computing and task offloading seamlessly intensive computation tasks to nearby edge servers.

NG-RAN Concept and Architecture. The NG-RAN architecture, defined by 3GPP, comprises a Distributed Units (DUs) located in the close proximity to the Base Station (BS) tower that able to communicate with a Central Unit (CU) via Next Generation Fronthaul Interface (NGFI) standard [14], in which the PHY/MAC layers of the network flexibly splits between the CU and DU locations. In this way, the NGFI interface brings enormous advantages to the fronthaul network in terms of the system flexibility and the network latency.

Another key feature of NG-RAN design is to flexibly move the main signal processing functions performed by the digital baseband (PHY/MAC) processing to the CU while maintaining the radio access and low levels of communication functionalities at the cell sites in the form of Distributed DUs. Cooperation between two main units in an efficient way will open a path to enhance the overall network significant metrics, including architecture planning, network operation, resource utilization, and back/mid/front-haul management. Consequently, multiple wireless 5G and beyond services, such as massive Machine-Type Communication (mMTC), enhanced Mobile Broadband (eMBB), and ultra-Reliable Low-Latency communication (uRLLC), can dynamically deploy and manage to satisfy the emerging requirements of a variety of 5G and beyond applications. Figure 1.3 describes the main components in NG-RAN system.

NG-RAN Functional Split Options. As a part of the NG-RAN study, 3GPP proposed several functional splits between CUs and DUs. Accordingly, it has been proposed 8 possible options shown in Fig 1.4 [2]. The choice of how to split the NG-RAN architecture depends on several factors related to radio network status, traffic size and network providers' services, such as low latency, high throughput, UE density, and the geographical location of DUs. By moving from Option 1 to Option 8, a tradeoff can be established between fronthaul latency and processing complexity. Basically, by adding more baseband functions at the DUs, the required fronthaul rate can be reduced, while the processing complexity will be

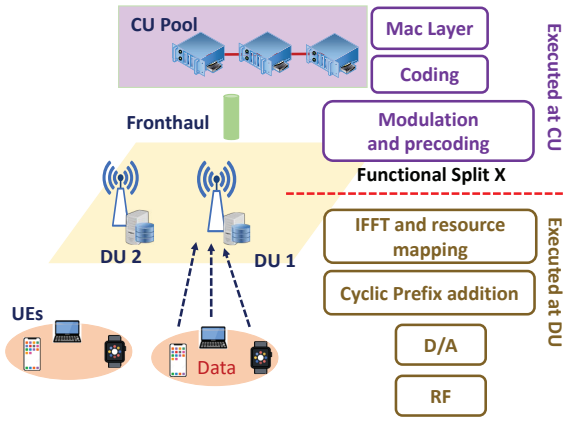


Figure 1.3: Example of NG-RAN architecture with enabling of function split option, in which computations can be performed at the CU and DU respectively.

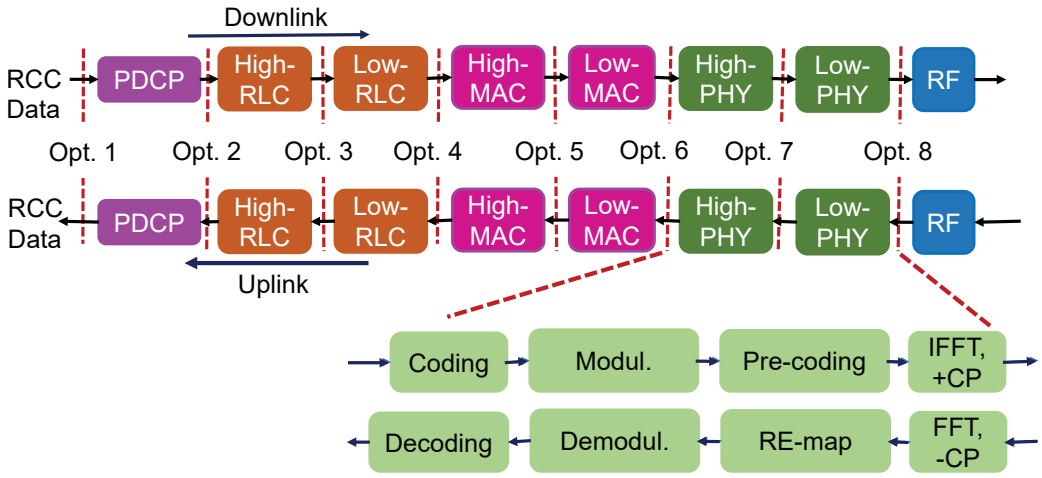


Figure 1.4: Logical diagram for uplink/downlink of gNB with eight functional split options.

increased, and the energy consumption at the DUs will be increased [15]. Specifically, computationally costly operations like Fast Fourier Transformation (FFT), Inverse Fast Fourier Transformation (IFFT), Rate Matching, and Turbo encoding/decoding are shifted to the CU side, resulting in variation in energy consumption at the CU and the DU.

1.3 Research Objectives and Contributions

The goal of this research is to design disruptive innovations for the wireless access network that always make the best use of the resources available to satisfy service requests from the users. The innovations should also make use of intelligence harvested from users and network context information such as the popular content being requested at a given time

in a given location, the computing resources required to process baseband data and to execute computation tasks for each user. This additional information can enable the network to make optimized control decisions both proactively and reactively so as to improve the users' communications and computation experiences. Fueled by the potential advantages of C-RAN, MEC , and NG-RAN, we aim at designing novel cooperative frameworks that optimize the control decisions for data transmissions, content provisioning, and computation in 5G systems. Specifically, our innovative solutions focus on improving downlink transmission throughput, reducing backhaul traffic load, and reducing end-to-end (e2e) latency for content delivery and mobile computation offloading. Our contributions in this dissertation are summarized in five specific topics as follows.

1. Bandwidth and Energy-Aware Resource Allocation for Cloud Radio Access Networks [16–18]: In this work, 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 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 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.

2. Energy-efficient Resource Allocation in C-RANs with Capacity-limited Fronthaul [18–20]: In this work, a novel resource allocation scheme that optimizes the network energy efficiency of a C-RAN is designed. First, an energy consumption model that characterizes the computation energy of the BBU is introduced based on empirical

results collected from a programmable C-RAN testbed. Then, an optimization problem is formulated to maximize the energy efficiency of the network, subject to practical constraints including QoS requirement, radio remote head transmit power, and fronthaul capacity limits. The formulated Network Energy Efficiency Maximization (NEEM) problem jointly considers the tradeoff among the network accumulated data rate, BBU power consumption, fronthaul cost, and beamforming design. To deal with the non-convexity and mixed-integer nature of the problem, we utilize successive convex approximation methods to transform the original problem into the equivalent Weighted Sum-Rate (WSR) maximization problem. We then propose a provably-convergent iterative method to solve the resulting WSR problem. Extensive simulation results coupled with real-time experiments on a small-scale C-RAN testbed show the effectiveness of our proposed resource allocation scheme and its advantages over existing approaches.

3. On-demand Video-streaming in Mobile-Edge Computing [21, 22]: In this work, we aim at optimizing the QoE for dynamic adaptive video streaming that takes into account the Distortion Rate (DR) characteristics of videos and the coordination among MEC servers. Specifically, the Video-streaming QoE Maximization (VQM) problem is cast as a Mixed-Integer Nonlinear Program (MINLP) that jointly determines the integer video resolution levels and video transmission data rates. Due to the challenging combinatorial and non-convex nature of this problem, the Dual-Decomposition Method (DDM) is employed to decouple the original problem into two subproblems, which can be solved efficiently using standard optimization solvers. Real-time experiments on a wireless video streaming testbed have been performed on a FDD downlink LTE emulation system to characterize the performance and computing resource consumption of the MEC server under various conditions. Emulation results of the proposed strategy show significant improvement in terms of users' QoE over traditional approaches.

4. Latency and Quality-Aware Task Offloading in Multi-Node Next Generation RANs [18, 23–25]: In this work, we propose a multi-edge node task offloading system, i.e., QLRan, a novel optimization solution for latency and quality tradeoff task allocation in Next-Generation Radio Access(NG-RANs). Considering constraints on service

latency, quality loss, edge capacity, and task assignment, the problem of joint task offloading, latency, and Quality Loss of Result (QLR) is formulated in order to minimize the User Equipment (UEs) task offloading utility, which is measured by a weighted sum of reductions in task completion time and QLR cost. The QLRan optimization problem is proved as a Mixed Integer Nonlinear Program (MINLP) problem, which is a NP-hard problem. To efficiently solve the QLRan optimization problem, we utilize Linear Programming (LP)-based approach that can be later solved by using convex optimization techniques. Additionally, a programmable NG-RAN testbed is presented where the Central Unit (CU), Distributed Unit (DU), and UE are realized by USRP boards and fully container-based virtualization approaches. Specifically, we use OpenAirInterface (OAI) and Docker software platforms to deploy and perform the NG-RAN testbed for different functional split options. Then, we characterize the performance in terms of data input, memory usage, and average processing time with respect to QLR levels. Simulation results show that our algorithm performs significantly improves the network latency over different configurations.

Deep Reinforcement Learning-based Resource Allocation for Next Generation Radio Access Networks: In this work, we introduce a novel Deep Reinforcement Learning Based Resource Allocation (ReLAX) framework to deal with the joint optimization of UE association and power allocation in NG-RAN systems. Considering the dynamic nature of the NG-RAN environment, ReLAx problem has been formulated to maximize the network EE under the constraints of QoS, fronthaul link, functional split configuration and transmit power budget. The optimization problem is cast via a Mixed-Integer Non-Linear Programming (MINLP), which is in general non-convex and NP-complete. A multi-task Deep Deterministic Policy Gradient (DDPG) method is proposed to solve the NG-RAN resource allocation optimization problem, in which two actors are trained to generate UE association and power allocation, respectively. We introduce the soft multi-task learning as a constraint during training so that one model would not drift too far away from the other one. Our real-time experiments on a fully containerized NG-RAN testbed show the effect of functional splits on CPU utilization and system latency. Besides, simulation results show that the proposed resource allocation solution outperforms competing traditional algorithms.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows.

Chapter 2 describes our proposed efficient resource-allocation solution that aims at minimizing the energy consumption of SDN-based C-RAN, including the power consumption of the BBU pool and the RRHs. including. We also discuss the limitation and practical considerations for realizing the virtualized BBU pool over a real-world implementation of a small-scale C-RAN system. Numerical simulations show that our proposed algorithm significantly improves the energy consumption of the network over traditional approaches.

Chapter 3 presents our proposed resource-allocation solution that optimizes the network energy efficiency of a C-RAN subject to practical constraints including meeting the users' QoS requirements, RRH transmission power, and frounthaul capacity limits. The performance of our proposed iterative algorithm is evaluated under different network conditions. Extensive simulation results coupled with testbed experiments showed that the proposed resource allocation solution optimizes C-RAN energy efficiency under practical physical constraints while significantly outperforms existing approaches.

Chapter 4 describes our Video-streaming QoE Maximization problem which is cast as a Mixed-Integer Nonlinear Program that jointly determines the integer video resolution levels and video transmission data rates. Real-time experiments on a wireless video streaming testbed have been performed on a FDD downlink LTE emulation system to characterize the performance and computing resource consumption of the MEC server under various conditions. Emulation results of the proposed strategy show significant improvement in terms of users' QoE over traditional approaches.

Chapter 5 presents a multi-edge node task offloading system, i.e., QLRan, a novel optimization solution for latency and quality tradeoff task allocation in NG-RANs. Considering constraints on service latency, quality loss, edge capacity, and task assignment, the problem of joint task offloading, latency, and Quality Loss of Result (QLR) is formulated in order to minimize the User Equipment (UEs) task offloading utility, which is measured by a weighted sum of reductions in task completion time and QLR cost. The QLRan optimization problem is proved as a Mixed Integer Nonlinear Program (MINLP) problem, which is a NP-hard

problem. To efficiently solve the QLRan optimization problem, we utilize Linear Programming (LP)-based approach that can be later solved by using convex optimization techniques. Additionally, a programmable NG-RAN testbed is presented where the Central Unit (CU), Distributed Unit (DU), and UE are realized by USRP boards and fully container-based virtualization approaches. Specifically, we use OpenAirInterface (OAI) and Docker software platforms to deploy and perform the NG-RAN testbed for different functional split options. Then, we characterize the performance in terms of data input, memory usage, and average processing time with respect to QLR levels. Simulation results show that our algorithm performs significantly improves the network latency over different configurations.

Chapter 6 introduces our novel Deep Reinforcement Learning Based Resource Allocation (ReLAX) framework to deal with the joint optimization of UE association and power allocation in NG-RAN systems. Considering the dynamic nature of the NG-RAN environment, ReLAx problem has been formulated to maximize the network Energy Efficiency (EE) under the constraints of Quality of Service (QoS), fronthaul link, functional split configuration and transmit power budget. The optimization problem is cast via a Mixed-Integer Non-Linear Programming (MINLP), which is in general non-convex and NP-complete. A multi-task Deep Deterministic Policy Gradient (DDPG) method is proposed to solve the NG-RAN resource allocation optimization problem, in which two actors are trained to generate UE association and power allocation, respectively. However, using two separate models for two closely-related variables could be a waste of training time and resources. As such, we introduce the soft multi-task learning as a constraint during training so that one model would not drift too far away from the other one. Our real-time experiments on a fully containerized NG-RAN testbed show the effect of functional splits on CPU utilization and system latency. Besides, simulation results show that the proposed resource allocation solution outperforms competing traditional algorithms, such as ordinary DDPG and Weighted Minimum Mean Square Error (WMMSE).

Chapter 7 summarizes our main contributions and provides suggestions on future research directions that will push the state-of-the-art in cloud-assisted wireless networks.

Chapter 2

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

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.

2.1 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× increase in Spectral Efficiency (SE) and 1000× 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) [26] 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) [27]. 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 [19, 28–32]. 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 [33–35]. 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 infrastructures 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) [36]—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.

2.2 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 [37–40] 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 [37], and additionally the user transmission power in [39], transport network power in [38], and power consumption at the BBU pool in [40]. In addition, the tradeoff between transmission power and delay performance is investigated in [41–43] via different approaches. Furthermore, the works in [19, 44–46] address the front-haul 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 [47], Amarisoft solution [48] , and OpenAirInterface (OAI) platform [49]. 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 [26, 50] 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 [51] focus on minimizing computational and networking latencies by VMs or containers. Kong *et al.* [52] present the architecture and implementation of a BBU cluster testbed to improve energy efficiency in C-RAN. Wu [53] 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 [8] presents the cross-layer resource allocation problem as a Mixed-Integer Nonlinear Programming (MINLP), which considers elastic service scaling, RRH selection, and beamforming. In [54], 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.*

Main Contributions: The objective of this chapter 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,

- 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.

Chapter Organization: In Sect. 2.3, we describe the C-RAN software plateform and the related system challenges; in Sect. 2.4, we introduce the system and power consumption models considered throughout this work; in Sect. 2.5, 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. 2.6, we present our testbed experiment

results as well as numerical simulation results to evaluate performance of our proposed algorithms. Finally, we conclude the chapter in Sect. 2.7.

2.3 C-RAN Software Platform and Implementation Challenges

We describe here 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.

2.3.1 Emulation Platform

We choose an open-source software implementation of LTE standard called OAI [49] 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 [55]. The structure of OAI mainly consists of two components: one, called Openairintereface5g, is used for building and running eNB units; the other, called Openair-cn, is responsible for building and running the Evolved Packet Core (EPC) networks, as shown in Fig. 2.1. The Openair-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 2.2 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.

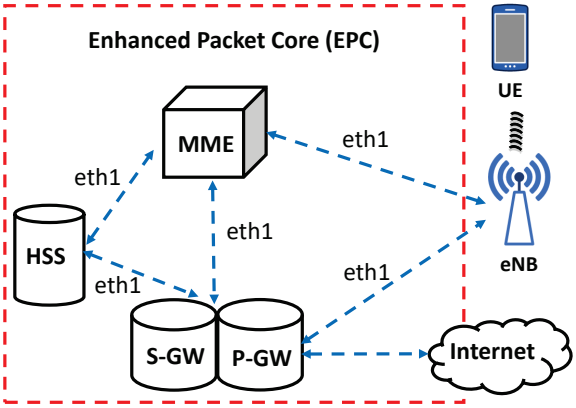


Figure 2.1: Evolved Packet Core (EPC) network topology diagram.

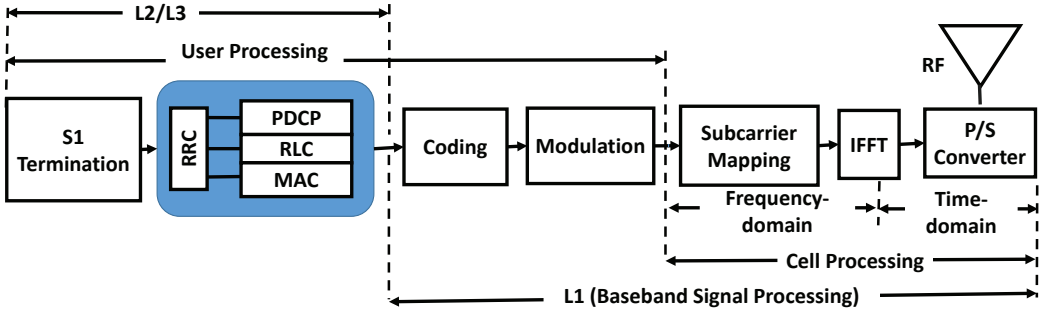


Figure 2.2: Diagram of downlink-BBU pool physical processing blocks.

2.3.2 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 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

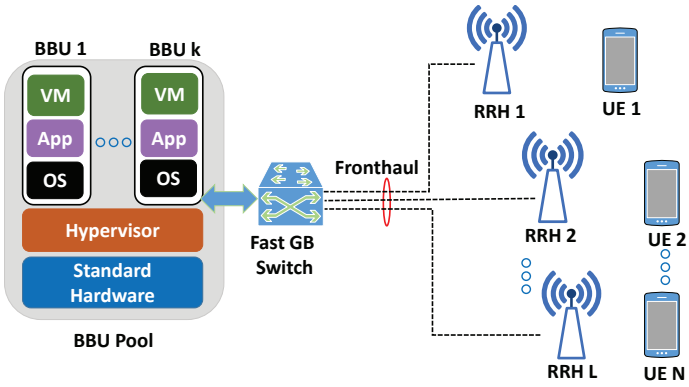


Figure 2.3: A conceptual structure of C-RAN network where the BBU pool can be realized by real-time VMs.

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.

2.4 System Model

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

2.4.1 Network Description

As illustrated in Fig. 2.3, 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

Table 2.1: 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

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,

$$\gamma_{ij} = \frac{g_{ij}h_{ij}}{B_{ij}\left(N_0 + I_i\right)}, \forall i \in \mathcal{N}, j \in \mathcal{L}, \tag{2.1}$$

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,

$$I_i = \sum_{k \in \mathcal{L} \setminus \{j\}} g_k^{max} h_{ik} / B_k^{max}, \forall i \in \mathcal{N}, \tag{2.2}$$

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,

$$r_{ij} = B_{ij} \log_2 \left[1 + \gamma_{ij}\right], \forall i \in \mathcal{N}, j \in \mathcal{L}. \tag{2.3}$$

Interference management mechanisms, such as Enhanced Inter Cell Interference Coordination (eICIC) and Coordinated Multipoint (CoMP) [56, 57], 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.

2.4.2 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 [58]. 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. 2.6.1, 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,

$$C_i = G_i f_i^{\mathrm{CPS}} + D_i, \forall i \in \mathcal{N}, \tag{2.4}$$

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 [59]. We use a_{ki} to indicate whether UE i is served by the VM generated by BBU k, which can expressed as,

$$a_{ki} = \begin{cases} 1 & \text{UE } i \text{ is served by BBU } k, \\ 0 & \text{otherwise,} \end{cases} \quad \forall k \in \mathcal{K}, i \in \mathcal{N}. \tag{2.5}$$

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

$$\mathcal{E}_k^{bbu} = \begin{cases} \mathcal{E}^a + \sum\limits_{i \in \mathcal{N}} a_{ki} \mathcal{E}_i(C_i) & \text{BBU } k \text{ is active,} \\ \mathcal{E}^s & \text{BBU } k \text{ is sleep,} \end{cases} \quad \forall k \in \mathcal{K}, \tag{2.6}$$

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,

$$\Delta_k^{bbu} = \mathcal{E}^a + \sum_{i \in \mathcal{N}} a_{ki} \mathcal{E}_i(C_i) - \mathcal{E}^s, \forall k \in \mathcal{K}. \tag{2.7}$$

According to [60], the amount of power consumption corresponding to UE i can be modeled as,

$$\mathcal{E}_i(C_i) = w_i C_i, \forall i \in \mathcal{N}, \tag{2.8}$$

where $w_i > 0$ is a constant.

Depending on Passive Optical Network (PON) model [61], 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,

$$P_{pon} = P_{olt} + \sum_{j \in \mathcal{L}} P_j^{fh}, \tag{2.9}$$

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,

$$P_j^{fh} = \begin{cases} P_j^a & \text{RRH } j \text{ is active,} \\ P_j^s & \text{RRH } j \text{ is sleep,} \end{cases} \forall j \in \mathcal{L}, \tag{2.10}$$

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,

$$\Delta_j^{fh} = P_j^a - P_j^s, \forall j \in \mathcal{L}. \tag{2.11}$$

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

$$b_j = \begin{cases} 1 & \text{RRH } j \text{ is active,} \\ 0 & \text{RRH } j \text{ is sleep,} \end{cases} \forall j \in \mathcal{L}. \tag{2.12}$$

Therefore, P_{pon} can be transformed as,

$$\begin{aligned} P_{pon} &= P_{olt} + \sum_{j\in\mathcal{L}} \left(P_j^a - P_j^s\right) b_j + \sum_{j\in L} P_j^s \\ &= P_{olt} + \sum_{j\in\mathcal{L}} \Delta_j^{fh} b_j + \sum_{j\in L} P_j^s. \end{aligned} \tag{2.13}$$

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,

$$\begin{aligned} P_{net} &= \mathcal{E}_{bbu} + P_{pon} + P^{tr} \\ &= \sum_{k\in\mathcal{K}} \left(\mathcal{E}^a + \sum_{i\in\mathcal{N}} a_{ki}\mathcal{E}_i(C_i) - \mathcal{E}_s \right) y_k + \sum_{k\in\mathcal{K}} \mathcal{E}s \\ &+ \sum_{j\in\mathcal{L}} \left(P_j^a - P_j^s\right) b_j + \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{L}} g_{ij} b_{ij} + P_c, \end{aligned} \tag{2.14}$$

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

2.5 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.

2.5.1 Resource Allocation Problem for Network Power Consumption Minimization

The network power consumption in (2.14) 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,

$$\mathcal{P}0: \underset{b_j, a_{ki}, B_{ij}, g_{ij}}{\text{minimize}} \quad P_{net} \tag{2.15a}$$

$$\text{s.t.} \quad r_{ij} \geq r_i^{\min}, \forall i \in \mathcal{N}, j \in \mathcal{L} \tag{2.15b}$$

$$\sum\nolimits_{i \in \mathcal{N}} B_{ij} \leq b_j B_j^{\max}, \forall j \in \mathcal{L}, \tag{2.15c}$$

$$\sum\nolimits_{i \in \mathcal{N}} g_{ij} \leq g_j^{\max}, \forall j \in \mathcal{L}, \tag{2.15d}$$

$$B_{ij} \geq 0, g_{ij} \geq 0, \forall i \in \mathcal{N}, \forall j \in \mathcal{L}, \tag{2.15e}$$

$$b_j, a_{ki} \in \{0, 1\}, \forall i \in \mathcal{N}, j \in \mathcal{L}, k \in \mathcal{K}, \tag{2.15f}$$

where constraint (2.15b) guarantees the QoS of UEs by keeping the data rate of UE i above or equal the target; and (2.15c) and (2.15d) 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 [62]. Our goal in this chapter 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.

2.5.2 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 (2.15), 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.

2.5.3 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,

$$\mathcal{P}1 : \text{find} \;\; B_{ij}, g_{ij} \tag{2.16a}$$

$$\text{s.t.} \;\; r_{ij} \geq r_i^{\min}, \forall i \in \mathcal{N}_j, j \in \mathcal{L}, \tag{2.16b}$$

$$\sum\nolimits_{i \in \mathcal{N}_j} B_{ij} \leq B_j^{\max}, \forall j \in \mathcal{L}, \tag{2.16c}$$

$$\sum\nolimits_{i \in \mathcal{N}_j} g_{ij} \leq g_j^{\max}, \forall j \in \mathcal{L}, \tag{2.16d}$$

$$B_{ij} \geq 0, g_{ij} \geq 0, \forall i \in \mathcal{N}_j, \forall j \in \mathcal{L}. \tag{2.16e}$$

We claim that all UEs in $\mathcal{N}_j$ can be served by RRH j if a feasible solution to (2.16) 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,

$$\mathcal{P}2 : \underset{B_{ij}, g_{ij}}{\text{minimize}} \;\; \sum\nolimits_{i \in \mathcal{N}_j} g_{ij} \tag{2.17a}$$

$$\text{s.t.} \;\; (2.16\text{b}) \sim (2.16\text{e}). \tag{2.17b}$$

From constraint (2.16b) and the definition of r_{ij} in (2.3), we conclude that,

$$g_{ij} = \frac{N_0 B_{ij}}{h_{ij}} \left(2^{\frac{r_i^{\min}}{B_{ij}}} - 1 \right), \forall i \in \mathcal{N}_j, j \in \mathcal{L}. \tag{2.18}$$

Finally, by substituting (2.18) into (2.17), the optimization subproblem can be recast as,

$$\mathcal{P}3: \underset{B_{ij}}{\text{minimize}} \quad \sum\nolimits_{i\in\mathcal{N}_j} \frac{N_0 B_{ij}}{h_{ij}} \left(2^{\frac{r_i^{\min}}{B_{ij}}} - 1\right), \tag{2.19a}$$

$$\text{s.t.} \quad \sum\nolimits_{i\in\mathcal{N}_j} B_{ij} = B_j^{\max}, \forall j \in \mathcal{L}, \tag{2.19b}$$

$$B_{ij} \geq 0, \forall i \in \mathcal{N}_j, \forall j \in \mathcal{L}. \tag{2.19c}$$

Lemma 1. *Problem* $\mathcal{P}3$ *in* (2.19) *is convex.*

Proof. We rewrite the objective of $\mathcal{P}3$ as,

$$f\left(B_{ij}\right) = \sum\nolimits_{i\in\mathcal{N}_j} \frac{N_0 B_{ij}}{h_{ij}} \left(2^{\frac{r_i^{\min}}{B_{ij}}} - 1\right), \forall j \in \mathcal{L}. \tag{2.20}$$

The objective function of (2.20) is convex *if and only if* its Hessian matrix is positive semi-definite [63]. In our case, the Hessian matrix can be calculated as,

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial(B_1)\partial(B_1)} & \frac{\partial^2 f}{\partial(B_1)\partial(B_2)} & \cdots & \frac{\partial^2 f}{\partial(B_1)\partial(B_N)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f}{\partial(B_L)\partial(B_1)} & \frac{\partial^2 f}{\partial(B_L)\partial(B_2)} & \cdots & \frac{\partial^2 f}{\partial(B_L)\partial(B_N)} \end{bmatrix},$$

$$\frac{\partial^2 f}{\partial(B_j)\partial(B_i)} = \begin{cases} 0, & \forall i \neq j, \\ \frac{N_0 (r_i^{\min})^2 (\ln 2)^2}{h_{ij}(B_{ij})^3} 2^{\frac{r_i^{\min}}{B_{ij}}} > 0, \forall B_{ij} > 0, \forall i = j \end{cases} \tag{2.21}$$

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 (2.19b) and (2.19b) are affine, we can state that problem $\mathcal{P}3$ is convex. The proof is complete.

□

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

$$\begin{aligned}\mathcal{L}(B_{ij},\lambda_i,\mu_i) = &\sum_{i\in\mathcal{N}_j}\frac{N_0 B_{ij}}{h_{ij}}\left(2^{\frac{r_i^{\min}}{B_{ij}}}-1\right)\\ &+\sum_{i\in\mathcal{N}_j}\lambda_i\left(B_{ij}-B_j^{\max}\right)-\sum_{i\in\mathcal{N}_j}\mu_i B_{ij},\end{aligned} \tag{2.22}$$

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 [63]; by applying Karush-Kuhn-Tucker (KKT) conditions [63], the following optimal values can be obtained as,

$$\lambda_i^* = -\frac{N_0}{h_{ij}}\left[\left(1-\frac{r_i^{\min}\ln 2}{B_{ij}^*}\right)2^{\frac{r_i^{\min}}{B_{ij}^*}}-1\right], \tag{2.23}$$

$$\sum\nolimits_{i\in\mathcal{N}_j} B_{ij}^* = B_j^{\max}, \forall j\in\mathcal{L}, \tag{2.24}$$

$$\mu_i^* = 0, B_{ij}^* > 0, \forall i\in\mathcal{N}_j, \forall j\in\mathcal{L}. \tag{2.25}$$

The BPA algorithm is detailed in Algorithm 1, where ϵ and Λ_i are a tolerance and an

Algorithm 1 BPA Algorithm for UE-RRH Clustering

Initialize: $x=0$, $\lambda_i^{(x)}=0$, $\lambda_i^{\min}=0$, and $\lambda_i^{\max}=\Lambda_i$, $\forall i\in\mathcal{N}$
repeat
  $x=x+1$, $\lambda_i^{(x)}=(\lambda_i^{\max}+\lambda_i^{\min})/2$
  for $i\in\mathcal{N}$ **do**
    Determine B_{ij} from $\mathcal{P}3$
  if $\sum_{i\in\mathcal{N}} B_{ij} > B_j^{\max}$ **then**
    $\lambda_i^{\min}=\lambda_i^{(x)}$
  else
    $\lambda_i^{\max}=\lambda_i^{(x)}$
until $|\lambda_i^{(x)}-\lambda_i^{(x-1)}|\le\epsilon$
for $i\in\mathcal{N}$ **do**
  $B_{ij}^*=B_{ij}$
  Determine g_{ij}^* from (2.18)
Output: B_{ij}^*, g_{ij}^*

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,

$$b_j^* = \begin{cases} 1 & \text{RRH } j \text{ is active, if } g_i(\{j\}) \geq g_i^*(\{j\}) \\ & \text{and } B_i(\{j\} \geq B_i^*(\{j\}) \ , \forall j \in \mathcal{L}_j, i \in \mathcal{N}, \\ 0 & \text{RRH } j \text{ is sleep, otherwise} \end{cases} \tag{2.26}$$

2.5.4 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 (2.14), we can cast the EARA problem as,

$$\mathcal{P}4 : \underset{a_{ki}, y_k}{\text{minimize}} \ \sum\nolimits_{k \in \mathcal{K}} \mathcal{E}_k^{bbu} y_k \tag{2.27a}$$

$$\text{s.t.} \ \sum\nolimits_{k \in \mathcal{K}} a_{ki} = 1, \forall i \in \mathcal{N}_j, \tag{2.27b}$$

$$\sum\nolimits_{i \in \mathcal{N}_j} s_i a_{ki} \leq U y_k, \forall k \in \mathcal{K}, \tag{2.27c}$$

$$a_{ki}, y_k \in \{0, 1\}, \forall k \in \mathcal{K}, i \in \mathcal{N}_j, \tag{2.27d}$$

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 (2.27b) ensures that the data from one UE can only be processed by one BBU;

constraint (2.27c) 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 [64, 65]. *In this chapter, 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 (2.8), 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

Initialize: $\mathcal{N}$, $\mathcal{L}$, $\mathcal{K}$, U,G_i, D_i, f_i^{CPS}, $\forall i \in \mathcal{N}, k \in \mathcal{K}$
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 (2.8)
    Select $j^* = \mathrm{argmin}\{\mathcal{E}_{\mathrm{i}}(\mathrm{C_i})\}\ \forall i \in \mathcal{N}_j, j^* \in \mathcal{L}$
    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
        Output: reallocated BBUs set

2.6 Performance Evaluation

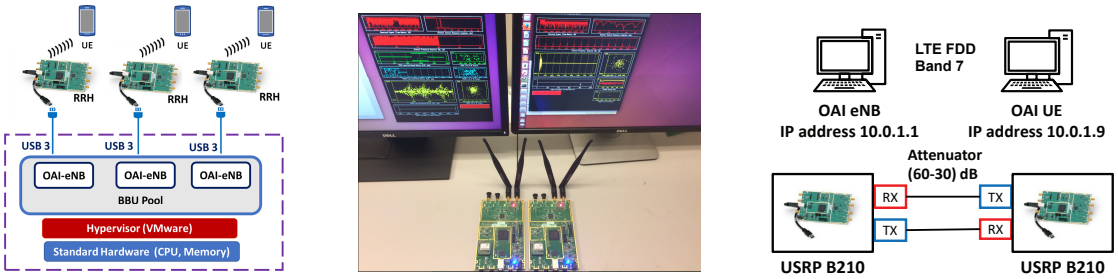
In this section, we first detail the experimental setups and results for the programmable C-RAN testbed. Then, we evaluate the performance of our proposed resource allocation algorithms, BPA and EARA, via numerical simulation results.

2.6.1 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.

Testbed Architecture. Figure 2.4(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 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 3.19.0-91-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 ssse3 and sse4.1 features. These flags must be exposed from the host to the guest, and can be checked by using the command 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) [dBm] and is part of the configuration file, pdsch_referenceSignalPower.



(a) (b) (c)

Figure 2.4: (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.

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 [dBm] is measured over the whole (usable) bandwidth. If the same hardware is used at the UE and at the eNB, the power is $\text{max\_ue\_power} = \text{PDSCH\_EPRE} + 10\log_{10}(12\text{N\_PRB})$.

Monitoring the OAI eNB and the UE. As illustrated in Fig. 2.4(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: (i) L1/L2 Implementation, Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP), GPRS Tunneling Protocol (GTP), and Radio Resource Control (RRC). (ii) Third Generation Partnership Project (3GPP) LTE MAC/PHY implementation. (iii) Frequency Division Duplex (FDD) and Time Division Duplex (TDD) modes supports. (iv) Built-in emulator and simulator.

OAI is supplied with useful monitoring tools such as network protocol analyzers, loggers, performance profilers, timing analyzers, and command line interfaces for performing the intended measurements and monitoring of the network. Specifically, the supported monitoring tools include: (i) OAI Soft Scope, which monitors received-transmitted waveforms and also tracks the channel impulse response. (ii) WireShark Interface and ITTI Analyzer, which analyzes the exchanges between eNB and UE protocols. (iii) OpenAirInterface performance

Table 2.2: 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

Procedure 1 OAI Setup Processing

1: Initialize: OAI installation, Kernel setup, CPU setting, USRP B210 configuration.
2: **repeat**
3: Configure CPU
4: Run eNB
5: Run UE
6: **if** RRC-RECONFIGURED message at UE is observed **then**
7: Observe Ping and Iperf Tests
8: **until** Testbed Stability
9: Output: OAI monitoring tools, OAI Soft Scope, Wireshark/PCAP interface, OAI timing analyzer, OAI message sequence Chart.

profiler, which is used for processing-time measurements.

We summarize the testbed configuration parameters in Table 2.2. 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.

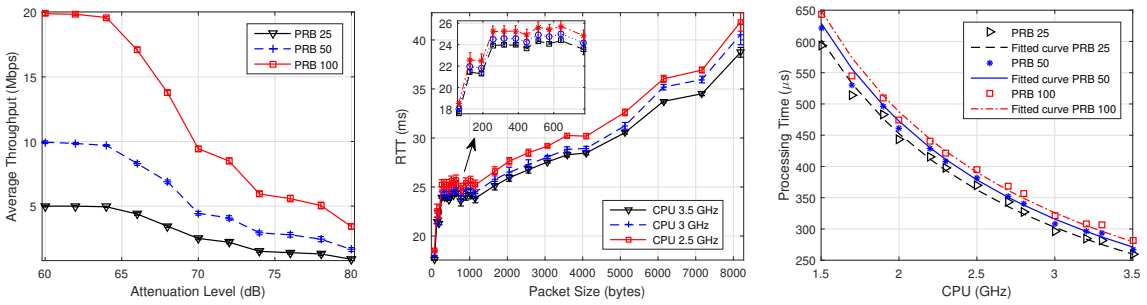
Interference-free Testbed Environment. We set up the experiment environment to emulate a "quiet" transmission between the eNB and UE in which there is no interference from other devices (so to have control of the environment). To accomplish this, we use two configurable attenuators, model name Trilithic 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 2.4(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 2.5(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. 2.5(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.

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 2.5(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 between 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.

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 [58]. The cpupower tool in Linux is used to control the



(a) (b) (c)

Figure 2.5: (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 MCS = 27 and various PRB allocations.

Table 2.3: Values of parameters α_{PRB} and β_{MCS}.

PRB	25	50	100
α_{PRB}	900	940	970

MCS	0	9	10	16	17	24	27
β_{MCS}	0	9.7	11.8	37.5	39.7	64.8	75

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÷3.5 GHz CPU frequency range.

In Fig. 2.5(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. 2.5(c) with different MCS indexes. The subframe processing time $T_{\text{sub}}\,[\mu s]$ can be well fitted as a function of CPU frequency, MCS, and PRB as,

$$T_{\text{sub}}\,[\mu s] = \frac{\alpha_{\text{PRB}}}{f^{CPS}} + \beta_{\text{MCS}} + 2.508, \tag{2.28}$$

where f^{CPS} [GHz] is the CPU frequency, and α_{PRB} and β_{MCS} are two parameters that increase with PRB and MCS values as reported in Table 2.3.

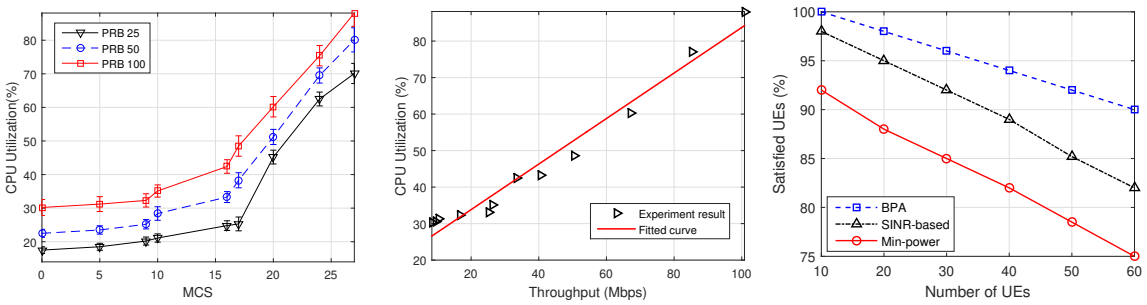


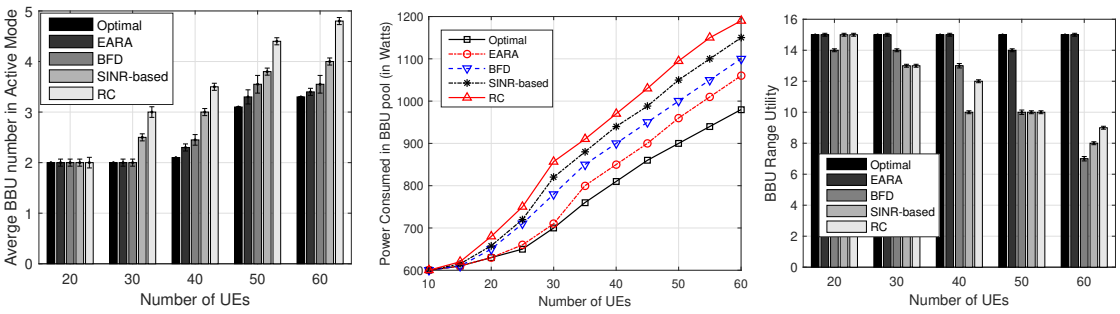
Figure 2.6: (a) CPU utilization of the BBU at different values of MCS and PRB; (b) Percentage of CPU usage versus the downlink throughput; and (c) Percentage of satisfied UEs for the different algorithms.

According to (2.28), we can derive the computation frequency that appears in (2.4) as,

$$f^{\mathrm{CPS}} = \frac{\alpha_{\mathrm{PRB}}}{T_{\mathrm{sub}} - \beta_{\mathrm{MCS}} - 2.508} \tag{2.29}$$

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. 2.6(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.

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



(a) (b) (c)

Figure 2.7: (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.

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) [66]. 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.

Figure 2.6(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,

$$\text{CPU}\,[\%] = 0.6237\phi + 21.3544, \tag{2.30}$$

where ϕ is the throughput measured in Mbps.

2.6.2 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) [67].

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 [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.

Performance of BPA Algorithm. For comparison, we introduce two user-association schemes that have been discussed in the literature: SINR-based scheme [68] and Min-power scheme [69]. 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 2.6(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.

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 [70].

- *RRH-Clustering (RC) Algorithm:* It is proposed in [64] 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 [68], 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 2.7(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 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 2.7(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. 2.7(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.

2.7 Summary

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.

Chapter 3

Energy-efficient Resource Allocation in C-RANs with Capacity-limited Fronthaul

In this chapter, a novel resource allocation scheme that optimizes the network energy efficiency of a C-RAN is designed. First, an energy consumption model that characterizes the computation energy of the BaseBand Unit (BBU) is introduced based on empirical results collected from a programmable C-RAN testbed. Then, an optimization problem is formulated to maximize the energy efficiency of the network, subject to practical constraints including Quality of Service (QoS) requirement, radio remote head transmit power, and fronthaul capacity limits. The formulated Network Energy Efficiency Maximization (NEEM) problem jointly considers the tradeoff among the network accumulated data rate, BBU power consumption, fronthaul cost, and beamforming design. To deal with the non-convexity and mixed-integer nature of the problem, we utilize successive convex approximation methods to transform the original problem into the equivalent Weighted Sum-Rate (WSR) maximization problem. We then propose a provably-convergent iterative method to solve the resulting WSR problem. Extensive simulation results coupled with real-time experiments on a small-scale C-RAN testbed show the effectiveness of our proposed resource allocation scheme and its advantages over existing approaches.

3.1 Introduction

The ever-increasing popularity of mobile devices and their demand for high data rates are presenting serious challenges to the wireless service providers. Cisco [71] envisages that the number of mobile-connected devices might reach 11.6 billion by 2021. In addition, the tremendous traffic growth is caused by the presence of various kinds of mobile devices such as smart phones, ipads, and wearable devices as well as new forms of connectivity such as

Internet of Things (IoT) and Machine-to-Machine (M2M) communications.

Many advanced technologies are being developed such as Heterogeneous Networks (HetNets), which create a complicated structure of different cell-size networks, and massive Multiple-Input Multiple-Output (MIMO), which requires cooperation among the Base Stations (BSs) [72]. However, both technologies come at the cost of an increase in energy consumption because of the additional energy needed to support the higher number of BS sites and substantially rely on expanded fronthaul capacity between the BSs for cooperation. The capacity of the conventional cellular network is generally designed to satisfy the peak traffic demand of the system without considering the temporal-spatial traffic fluctuations in the service area. Therefore, there exist active BSs with light traffic loads that still consume a considerable amount of basic power (e.g., power amplifier and L2/L3 processing power).

Recently, C-RAN has been proposed as a novel architecture for 5G cellular networks to overcome the difficulties in providing fast and reliable real-time communications. Unlike the existing cellular networks, wherein the radio resources for baseband processing are determined at the cell level, C-RAN's radio resources are allocated and coordinated in a centralized and powerful computing platform, a.k.a. the *cloud*. This movement from distributed systems to a centralized one for baseband processing has noticeable gains such as reducing energy usage, capital expenditure (CAPEX), and operating expenditure (OPEX) within the cellular networks [35, 73]. A typical C-RAN consists of: (i) light-weight, distributed Radio Remote Heads (RRHs) plus antennae, which are located at the remote sites and are controlled by a centralized virtual BS pool, (ii) the BaseBand 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 RRHs to the BBU pool. Due to its centralized nature, C-RAN shows significant promise in improving both the Spectral Efficiency (SE) and the Energy Efficiency (EE) of current wireless networks [73].

The centralized BBU structure in C-RAN facilitates cross-cell cooperation, improves SE, and can enhance the QoS for all UEs in the mobile network [16]. The fronthaul links with high-bandwidth and low-latency connect the RRHs with the BBU pool while backhaul links refer to the links between the mobile core network and the BBUs. Although C-RAN

provides strong computing capabilities by sharing computing and storage resources at the BBU pool, it still suffers from performance limitation due to the limited capacity of the fronthaul and backhaul links [58]. To overcome this shortage, traffic through fronthaul links should be taken into account while designing the resource allocation optimization strategy. Furthermore, the energy consumption of a C-RAN can be significantly reduced if we optimize the tradeoff between SE and EE. On the other hand, the relation between SE and EE under realistic and complex network scenarios still requires further investigation because of the many factors that must be jointly considered such as system sum-data rate, computation energy in BBU pool, fronthaul cost, and transmit energy consuming in RRHs In this chapter, our objective is to maximize the network EE of a C-RAN, taking into account the energy consumption at both the BBU pool and the RRHs. Based on the empirical results obtained from our programmable C-RAN testbed, we formulate the Network Energy Efficiency Maximization (NEEM) optimization problem, subject to practical constraints including user QoS requirement, system power, and fronthaul capacity limits. The considered problem is NP-hard and, as such, difficult to solve. Hence, we focus on designing a low-complexity algorithm to allow for practical implementations.

3.2 Related Work

Considerable attention has been paid on cooperative communications techniques for C-RAN under various different objectives. For instance, the trade-off between transmission power and delay performance is investigated in [74] via cross-layer-based approaches. Furthermore, the fronthaul uplink compression problem is addressed in [75]. In parallel, several works have focused on system architectures, feasibility of virtual software BS stacks, performance requirements, and analysis of optical links between the RRHs and the BBU cloud. For example, Kong *et al.* [52] present the architecture and implementation of a BBU cluster testbed to improve EE in a C-RAN. Liu *et al.* [76] implement an OFDMA-based C-RAN testbed with a reconfigurable backhaul architecture. The authors in [51] focus on minimizing computational and networking latency by Virtual Machines (VMs) or containers. 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 [47],

and OpenAirInterface (OAI) platform [49].

Other works have addressed the enhancement of the EE and the resource management of the C-RAN based on maximizing WSR. For instance, the work in [77] presents the problem of downlink beamforming to improve the EE of C-RANs by focusing on two different downlink transmission strategies, namely the data-sharing strategy and the compression strategy. The work in [16] proposes a novel resource-allocation scheme, which is based on bin-packing theory, that minimizes the number of active VMs in the BBU pool to save energy. The authors in [78] propose a user-centric clustering scheme to maximize the network utility based on data joint transmission. A grouping scheme of users and RRHs is proposed in [79] to achieve high network performance in C-RAN. Considering a dynamic radio-cooperation strategy, the authors in [80] address the problem of user-centric radio clustering for a C-RAN system, with a low-complexity and fast-convergence solution.

With a similar focus as ours, in [81] the authors introduce an optimization framework for deciding how to select the functional splitting for each BS, where to place the Mobile Edge computing (MEC) functions, and how to route the data in the shared fronthaul network. The authors in [38] formulate a group sparse beamforming problem to minimize the network power consumption of C-RAN, including the transport network and radio access network power consumption, with a QoS constraint at each user. The work in [82] studies the coordinated multipoint joint transmission design problem for C-RAN that explicitly considers the fronthaul capacity and users' QoS constraints. The authors in [83, 84] jointly optimize the precoding matrices and the set of active RRHs to minimize the network power consumption for a user-centric C-RAN with the consideration of limited fronthaul capacity and the unavailability of full Channel State Information (CSI). However, only weighted sum-rate maximization was considered. Thus, energy-efficient resource allocation optimization in C-RAN with capacity-limited fronthaul needs to be addressed. Along with this line, the authors in [60, 65] theoretically study the problem of jointly optimizing the resource of the BBU pool and beamforming in the coordinated RRH cluster with special attention to the limited fronthaul capacity of C-RAN system. Our proposal is fundamentally different in three aspects. First, we derive an empirical yet realistic model for the processing power

consumption at the BBU pool. Second, we investigate the issues of user' QoS and beamforming for EE maximization in C-RAN systems under capacity-limited fronthaul. Finally, we formulate the NEEM problem as a Mixed Integer Nonlinear Programming (MINLP) optimization problem over the beamforming vectors. Given the difficult non-convex nature of this problem, we utilize successive convex approximation method to transform the original problem into an equivalent WSR optimization problem, which is then solved using a Weighted Minimum Mean Square Error (WMMSE) approach.

Main Contributions: The objective of this chapter is to propose an efficient resource-allocation scheme in a C-RAN that aims at maximizing the EE of the C-RAN system, subject to the practical constraints including QoS, system power, and fronthaul capacity limit. Specifically, the main contributions of this chapter are summarized as follows.

- Using Software-defined Radio (SDR) OAI platform and virtualization environment, we perform real-time experiments on a small-scale C-RAN testbed that establishes transmissions between the eNodeB (eNB) and the User Equipment (UE). The experiments are carried out under various configurations in order to profile the runtime complexity and performance limits of the BBU in terms of processing, throughput, and latency. It is shown that the BBU's CPU utilization can be modeled as a linear increasing function of the maximum downlink data rate. As a side note, our testbed models provide researchers with real-world insights and tools for designing EE algorithms in C-RAN systems.

- Using empirical data collected from our testbed, we model the network power consumption in a C-RAN system consisting of three main parts: the computation power consumed in the BBU pool, fronthaul energy cost, and the Radio Frequency (RF) power transmitted by RRHs. We establish BBU computation model via testbed experiments and novel conic optimization techniques to balance the tradeoff between EE and QoS effectively for the downlink C-RAN.

- The limited fronthaul capacity is explicitly considered in the NEEM optimization problem for the downlink C-RAN. To avoid the difficulty of the feasibility problem caused by relaxing the l_0-norm constraint directly, we reformulate the original NEEM

problem into an equivalent problem by using a cost vector. Then, we utilize the reweighed l_1-norm relaxation and Successive Convex Approximation (SCA) methods to devise a provably-convergent iterative algorithm. Then, we reduce the proposed energy-efficient weighted utility function to a Weighted Sum Rate (WSR) maximization problem and a minimum energy beamforming problem. Then, we utilize the WMMSE approach to solve these problems. Furthermore, we present the Branch-and-Bound (BnB) method to optimally solve the relaxed NEEM problem. Unfortunately, the BnB's complexity scales exponentially with the problem size, making it impractical to use online; hence, it is mainly used as an offline optimal benchmark to make comparisons against.

- We provide formal proofs on the convergence and optimality of our algorithm and evaluate its performance under different network conditions. Numerical results show that the resource management of C-RAN can be optimized in terms of network energy efficiency under practical physical constraints.

Chapter Organization The remainder of this chapter is organized as follows. In Sect. 3.3, we introduce the system and power consumption models considered throughout this work. In Sect. 3.4, we formulate the NEEM optimization problem, discuss practical considerations, and propose an original approach to solve it. Simulation and testbed experiment results are presented in Sect. 3.5. Finally, we draw the main conclusions in Sect. 3.6.

3.3 System Model

We first describe the system models including the network architecture, wireless communications, and computation capacity of each BBU. Then, we mathematically formulate the network power system and present its practical constraints.

3.3.1 System Description

We consider a downlink 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 $M > 1$ antennas. All the RRHs are connected to a BBU pool via low-latency,

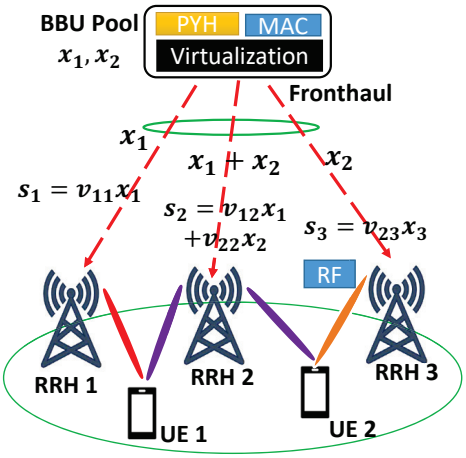


Figure 3.1: Downlink C-RAN with data sharing transmission strategy where the BBU processes the $\{x_1, x_2\}$ data delivered to RRHs. Users 1 and 2 are cooperatively served by RRH clusters (1,2) and (2,3), respectively.

high-bandwidth Common Public Radio Interface (CPRI). The BBU pool is composed of high-speed programmable processors and real-time VMs to carry out PHY/MAC-layer functionalities. All the UE data is assumed to be available at the BBU pool and each UE i receives a signal independent data stream from the RRHs. After processing by the BBU pool, the data is forwarded to the UE via a group of RRHs, denoted as $\mathcal{L}_i \subseteq \mathcal{L}$. We consider the data-sharing transmission strategy for the downlink of C-RAN where each UE's message is shared among a cluster of serving RRHs. Fig. 3.1 illustrates an example where the BBU pool processes the UE data x_1 and x_2, which are forwarded to RRHs 1, 2, and 3 through fronthaul links. UE 1 is cooperatively served by cluster including RRHs 1 and 2, while UE 2 is cooperatively served by the cluster including RRHs 2 and 3 through joint beamforming. Let $v_{ij} \in \mathbb{C}^{M\times 1}$ be the beamforming coefficient vector for RRH j to serve UE i. The value of v_{ij} is set to zero if RRH j is not part of the UE i's serving cluster. The transmit signal s_j at RRH j can be written as $s_j = \sum_{i\in\mathcal{N}} v_{ij}x_i$. We model the user data x_i as independent and identically distributed (i.i.d.) complex Gaussian random variables with zero mean and unit variance. Specifically, the received signal at UE i can be written as,

$$y_i = \sum_{j\in\mathcal{L}_i} h_{ij}^H v_{ij}x_i + \sum_{k\in\mathcal{N}\setminus\{i\}} \sum_{j\in\mathcal{L}_k} h_{ij}^H v_{kj}x_k + z_i, \tag{3.1}$$

where $h_{ij} \in \mathbb{C}^{M\times 1}$, $(.)^H$ represents the conjugate transpose, and z_i is the zero-mean circularly symmetric Gaussian noise denoted as $\mathcal{CN}\left(0, \sigma_i^2\right)$. Hence, the Signal-to-Interference-plus-Noise Ratio (SINR) of user i can be calculated as,

$$\gamma_i = \frac{\left|\sum_{j\in\mathcal{L}_i} h_{ij}^H v_{ij}\right|^2}{\sum_{k\in\mathcal{N}\setminus\{i\}}\left|\sum_{j\in\mathcal{L}_k} h_{ij}^H v_{kj}\right|^2 + \sigma_i^2}, \forall i,k \in \mathcal{N} \tag{3.2}$$

Consequently, the achievable rate for UE i is calculated as,

$$r_i = B\log_2\left(1+\gamma_i\right), \forall i \in \mathcal{N}, \tag{3.3}$$

where B is the channel bandwidth.

3.3.2 Computation Model

In a real-word downlink LTE system, as illustrated in Fig. 2.2, the overall processing is the sum of per User Processing (UP) and Cell Processing (CP). The UP depends only on the Modulation and Coding Scheme (MCS) index and the Physical Resource Block (PRB) allocated to the users as well as on the number of iterations required by the decoder, which is proportional to the SINR and channel conditions. Furthermore, the number of iterations required to successfully encode and decode a codeword for a specific value of γ_i tightly depends on the selected data rate r_i. Specifically, if r_i is chosen close to the channel capacity $\log_2(1+\gamma_i), \forall i \in \mathcal{N}$, a high number of iterations will be required for decoding in the uplink-BBU pool [85]. However, as the allocated r_i decreases for the given γ_i, the number of required iterations also decreases. Hence, the overall computational complexity to process one codeword scales with the number of information bits that are processed and with the number of iterations for coding/decoding. Therefore, the computational capacity can be computed as the product of the number of information bits and the required encoding/decoding iterations divided by the number of channel users, as in [85,86]. However, in our work we aim at establishing a numerical computational capacity model that considers the SINR, channel condition, MCS, and PRB. Therefore, we present in Sect. 3.5.1 the computational characterizations in the BBU pool.

Based on the profiling results, we have observed that the CPU utilization of the BBU increases linearly with the SINR, the PRB resource, and MCS index. Under this premise, we consider the computation capacity C_i [cycles/s], from the BBU pool that is allocated for UE i, as a linearly increasing function of the user downlink data rate, considering the PHY-layer conditions. Specifically, the capacity utilized for processing the data of UE i can be modeled as,

$$C_i = I_i^{snr} + G_i\varphi(r_i) + D_i, \forall i \in \mathcal{N}, \tag{3.4}$$

where $\varphi(r_i)$ is a function of the achievable rate for UE i; while the I^{snr}, G_i, and D_i are positive constants that can be estimated by offline profiling of the C-RAN testbed, as shown in Sect. 3.5.1, Table 3.1. Although the computational capacity in (3.4) is a piece-wise linearization approximation on a "quiet" transmission between the eNB and UE in which there is no interference from other devices, we address the SINR and mobility aspects from different angles: i) we consider the SINR as a key factor that affects the QoS; and ii) we incorporate mobility in our simulations by running multiple channel realizations.

Table 3.1: Values of parameters I_{snr}, G, and D.

MCS	**Modulation**	**SINR (I_{snr})**			**MCS**	
		10	**20**	**30**	***G***	***D***
4	QPSK	0	2.27	7.34	0.3319	5.623
8	16QAM	0	3.13	9.62	0.5194	8.302
27	64QAM	0	14.23	17.4	0.7136	10.71

3.3.3 Power Consumption Model

In this work, we assume that the total power consumption in downlink C-RAN system consists of two main parts: the computation power spent in the BBU pool and the power consumption of RRHs in the downlink transmissions. In practice, the BBU pool can dynamically adjust the VMs' computation capacities to handle the dynamic user traffic and channel states. The power consumption of the BBU pool is closely related to computing workloads for baseband signal processing [87]. Hence, we can model the computation power

consumption for serving UE i as,

$$\mathcal{E}_i^{Pr} = \mathcal{E}_i^S + \mathcal{E}_i(C_i), \tag{3.5}$$

where parameter $\mathcal{E}_i^S$ represents the static part of power consumption of a VM in working mode, which includes the power consumption of fronthaul transmission equipment, while $\mathcal{E}_i(C_i)$ represents the CPU power consumption due to processing of baseband signal x_i of UE i. According to [17], the amount of power consumption for serving UE i can be modeled as,

$$\mathcal{E}_i(C_i) = w_i C_i, \tag{3.6}$$

where $w_i > 0$ is a constant. By substituting (3.4), and (3.6) into (3.5), the power consumption of the entire BBU pool can be calculated as,

$$\mathcal{E}^{Pr} = \sum_{i\in\mathcal{N}} \mathcal{E}_i^{Pr} = \sum_{i\in\mathcal{N}} \mathcal{E}_i^S + \sum_{i\in\mathcal{N}} w_i \left(I_i^{snr} + G_i\varphi(r_i) + D_i\right). \tag{3.7}$$

The fronthaul links that connect the RRHs with the BBU pool can be modeled as a set of communication channels, each with a specific capacity and power dissipation. Accordingly, we formulate the fronthaul power consumption of these links as,

$$\mathcal{E}^{Fh} = \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{L}} P_j \left\| \|v_{ij}\|_2^2 \right\|_0, \forall j \in \mathcal{L}, \tag{3.8}$$

where P_j is a static cost when RRH j is active, the l_0-norm $\left\| \|v_{ij}\|_2^2 \right\|_0$ is an indicator function, which specifies the associations between UEs and RRHs, i.e.,

$$\left\| \|v_{ij}\|_2^2 \right\|_0 = \begin{cases} 1 & \|v_{ij}\|_2^2 \neq 0, \\ 0 & \|v_{ij}\|_2^2 = 0, \end{cases} \forall i \in \mathcal{N}, j \in \mathcal{L}, \tag{3.9}$$

Specifically, $\|v_{ij}\|_2^2 = 0$ indicates that the BBU pool will not deliver data for the i-th UE through the j-th RRH via the corresponding fronthaul link and the j-th RRH does not participate in the joint transmission to the i-th UE; and $\|v_{ij}\|_2^2 \neq 0$ otherwise.

Based on the beamforming vectors of each UE, the total power consumption of the RRHs can be expressed as,

$$\mathcal{E}^{Tr} = \sum\nolimits_{i\in\mathcal{N}} \sum\nolimits_{j\in\mathcal{L}_i} \|v_{ij}\|_2^2, \tag{3.10}$$

where $\sum_{j\in\mathcal{L}_i} \|v_{ij}\|_2^2$ represents the transmit power from all RRHs to UE i. The total network power consumption is calculated as,

$$\begin{aligned} \mathcal{E}(\mathbf{v}) &= \mathcal{E}^{Pr} + \mathcal{E}^{Fh} + \mathcal{E}^{Tr} \\ &= \sum_{i\in\mathcal{N}} \mathcal{E}_i^S + \sum_{i\in\mathcal{N}} w_i \left(I_i^{snr} + G_i\varphi(r_i) + D_i\right) \\ &+ \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{L}} P_j \left\| \|v_{ij}\|_2^2 \right\|_0 + \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{L}_i} \|v_{ij}\|_2^2, \end{aligned} \tag{3.11}$$

where $\mathbf{v} = \{v_{ij} | \forall i \in \mathcal{N}, j \in \mathcal{L}\}$. As we can see from (3.11), there are two approaches to improve the EE of C-RAN: reducing the transmit power and decreasing the BBU pool power. However, these two approaches are tightly coupled and are difficult to realize simultaneously: on the one hand, deceasing RRHs power means reducing the capability of maintaining the QoS for the users; on the other hand, lower computation capacity of BBU pool leads to less user information being shared among RRH j so that the RRHs achieve worse cooperation to mitigate interference. Hence, a joint design is needed to balance the BBU computation capacity and the RRH transmit power.

3.3.4 System Constraints

In addition to the system models described above, we introduce the following three constraints to capture the features of a C-RAN.

1. *QoS constraint:* the QoS can be accounted as the constraint of keeping the data rate of each UE i above or equal to the desired data rate r_i^{min}, i.e.,

$$r_i \geq r_i^{min}, \forall i \in \mathcal{N}. \tag{3.12}$$

2. *System power constraint:* it is assumed that RRH j has a maximum transmit power

of P_j^{max}, i.e.,

$$\sum_{i \in \mathcal{N}} \|v_{ij}\|_2^2 \leq P_j^{max}, \forall j \in \mathcal{L}. \tag{3.13}$$

3. *Fronthaul capacity constraint:* practically, the fronthaul links between the RRHs and the BBU pool in C-RAN system are capacity limited; in other words, the number of UEs accessing to each RRH is limited. In this case, the fronthaul constraint corresponding to RRH j can be expressed as,

$$\sum_{i \in \mathcal{N}} \left\| \|v_{ij}\|_2^2 \right\|_0 \leq B_j, \forall j \in \mathcal{L}, \tag{3.14}$$

where $B_j \in \mathbb{N}$ is defined as the maximum j-th fronthaul capacity, i.e, the maximum number of UEs that can be connected with j-th fronthaul link.

3.4 Energy Efficiency Maximization

We formulate now the NEEM problem ($\mathcal{P}0$) as a MINLP that optimizes the tradeoff among the network accumulated data rate, C-RAN power consumption, fronthaul cost, and beamforming design. Due to the intractability of the problem and the need for a practical online solution, we then present a step-by-step relaxation and reformulation approach to simplify $\mathcal{P}0$ in order to obtain a reasonable sub-optimal solution. Our approach is as follows.

1. In Sect. 3.4.1, we cast the NEEM problem $\mathcal{P}0$ as a MINLP, which is NP-hard and difficult to solve.

2. In Sect. 3.4.2, to tackle the non-convexity in $\mathcal{P}0$, we exploit the properties of fractional programming, in which the original problem in fractional form is transformed into an equivalent optimization subproblem with a subtractive form, $\mathcal{P}1$, and then recast it as $\mathcal{P}2$, as explained in Lemma 2. Then, to handle the convergence of $\mathcal{P}2$, we introduce Theorem 1, which proposes a method that shows the relationship between $\mathcal{P}2$ and suboptimal problem $\mathcal{P}3$. In addition, we present some properties of the cost vector Ψ in $\mathcal{P}3$, and other requirements to satisfy Theorem 1. For a fixed cost vector Ψ, we apply reweighed l_1-norm relaxation on $\mathcal{P}3$ to simplify it to another problem, $\mathcal{P}4$.

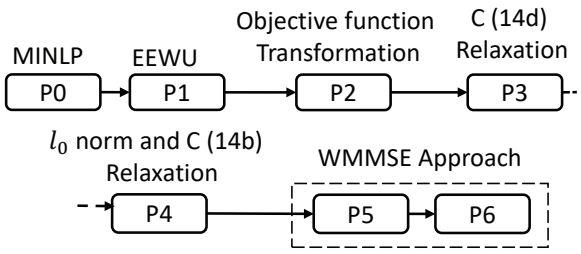


Figure 3.2: The complete process to solve the NEEM optimization problem ($\mathcal{P}0$) via multiple transformations ($\mathcal{P}1 \rightarrow \mathcal{P}6$).

3. Finally, in Sect. 3.4.3, we propose an iterative method to solve $\mathcal{P}4$ based on the WMMSE approach, as described in $\mathcal{P}5$ and $\mathcal{P}6$.

The complete process of transforming the original NEEM problem in such a way as to obtain its suboptimal solution is depicted in Fig. 3.2.

3.4.1 NEEM Problem Formulation

Our objective is to find the optimal beamforming vectors of the RRHs that maximize the EE of the considered C-RAN, subject to the system constraints described in Sect. 3.3.4. The EE of the C-RAN is defined as the ratio between the total throughput, $R(\mathbf{v}) = \sum_{i \in \mathcal{N}} r_i$, and the total power consumption given in (3.11) [88]. Therefore, we formulate the NEEM problem as follows,

$$\mathcal{P}0: \quad \max_{\{v_{ij}\}} \ \eta_{EE} = \frac{R(\mathbf{v})}{\mathcal{E}(\mathbf{v})} \tag{3.15a}$$

$$\text{s.t.} \ \ r_i \geq r_i^{\min}, \forall i \in \mathcal{N}, \tag{3.15b}$$

$$\sum_{i \in \mathcal{N}} \|v_{ij}\|_2^2 \leq P_j^{max}, \forall j \in \mathcal{L}, \tag{3.15c}$$

$$\sum_{i \in \mathcal{N}} \left\| \|v_{ij}\|_2^2 \right\|_0 \leq B_j, \forall j \in \mathcal{L}. \tag{3.15d}$$

In problem $\mathcal{P}0$ above, constraint (3.15b) guarantees the QoS requirement of the UEs by keeping the data rate of each UE i above or equal the target rate; (3.15c) is a maximum transmit power constraint for RRH j; and constraint (3.15d) accounts for the fronthaul capacity limit. In $\mathcal{P}0$, the fractional-form objective function is non-convex, and the l_0-norm in constraint (3.15d) imposes additional difficulties to the problem. It can be seen

that $\mathcal{P}0$ is a MINLP, which is NP-hard [89] and is highly difficult to solve optimally in polynomial time. To overcome these drawbacks, we propose to reformulate problem $\mathcal{P}0$ using relaxation techniques in the following subsections in order to devise a tractable, low-complexity solution.

3.4.2 Problem Transformation

Firstly, in order to deal with the non-convexity of the objective function in $\mathcal{P}0$, which is in fractional form, we convert it into a subtractive form. Specifically, the the EE in $\mathcal{P}0$ can be transformed to the Energy Efficiency Weighted Utility (EEWU) function as in [90]. Therefore, the NEEM problem is now equivalent to,

$$\mathcal{P}1: \max_{\{v_{ij}\}} \; R(\mathbf{v}) - \lambda^* \mathcal{E}(\mathbf{v}) \tag{3.16a}$$

$$\text{s.t.} \quad (3.15b) \sim (3.15d), \tag{3.16b}$$

where λ is a constant representing the system power consumption weight. Without loss of generality, we let $\lambda = \frac{R}{\mathcal{E}(\mathbf{v})}$ and $\lambda^* = \underset{\mathbf{v}}{\text{maximize}} \; \frac{R}{\mathcal{E}(\mathbf{v})} = \frac{R^*}{\mathcal{E}(\mathbf{v}^*)}$, where $\mathbf{v}^*$ is the optimal solution to (3.16). Then, we obtain the following lemma.

Lemma 2. *$\mathcal{P}1$ is equivalent to $\mathcal{P}0$ if and only if $R(\mathbf{v}^*) - \lambda^* \mathcal{E}(\mathbf{v}^*) = 0$.*

Proof. Denote $(\mathbf{v})$ and $(\mathbf{v}^*)$ as a feasible solution and an optimal solution to the $\mathcal{P}1$, respectively. Since $R(\mathbf{v}^*) - \lambda^* \mathcal{E}(\mathbf{v}^*) = 0$ and $R(\mathbf{v}) - \lambda^* \mathcal{E}(\mathbf{v}) \leq 0$, then $\frac{R(\mathbf{v})}{\mathcal{E}(\mathbf{v})} \leq \frac{R(\mathbf{v}^*)}{\mathcal{E}(\mathbf{v}^*)}$. Thus, $\mathbf{v}^*$ maximizes $\frac{R(\mathbf{v})}{\mathcal{E}(\mathbf{v})}$ while satisfying all constraints in $\mathcal{P}0$. Therefore, $\mathbf{v}^*$ is the optimal solution to $\mathcal{P}0$. Then, Lemma 2 is proved. □

According to Lemma 2, $\mathcal{P}0$ can be transformed to $\mathcal{P}1$ if the optimal λ, i.e., λ^*, is obtained. Hence, we employed bi-section method to find λ^* and the corresponding iterative routine is summarized in Routine 1. The objective function in (3.16) can be rewritten as,

Routine 1 Bi-section Search for Finding λ^*

1: Initialize: $\lambda^{min} = 0$, and $\lambda^{max} = \frac{\sum_{i\in\mathcal{N}} r_i^{min}}{\sum_{i\in\mathcal{N}}(\mathcal{E}_i^s + w_i D_i)}$
2: Set a small threshold for convergence check ξ
3: Let $\lambda = \frac{\lambda^{min}+\lambda^{max}}{2}$
4: Solve $\mathcal{P}1$ for a given λ and obtain the feasible solution $\{v_{ij}\}$
5: **if** $R(\mathbf{v}) - \lambda^*\mathcal{E}(\mathbf{v}) \leq \xi$ **then**
6: $\quad\lambda^{max} = \lambda$
7: **else**
8: $\quad\lambda^{min} = \lambda$
9: **if** $|\lambda^{max} - \lambda^{min}| \leq \xi$ **then**
10: $\quad$Convergence and break
11: **else**
12: $\quad$Return to step 3

$$\begin{aligned} f(\mathbf{v}) = &\sum\nolimits_{i\in\mathcal{N}} r_i - \lambda^* \sum\nolimits_{i\in\mathcal{N}} \mathcal{E}_i^S - \lambda^* \sum\nolimits_{k\in\mathcal{K}} w_i \times \\ &(I_i^{snr} + G_i\varphi(r_i) + D_i) - \lambda^* \sum\nolimits_{i\in\mathcal{N}} \sum\nolimits_{j\in\mathcal{L}_i} \|v_{ij}\|_2^2 \\ &- \lambda^* \sum\nolimits_{i\in\mathcal{N}} \sum\nolimits_{j\in\mathcal{L}} P_j \left\| \|v_{ij}\|_2^2 \right\|_0. \end{aligned} \tag{3.17}$$

Hence, the NEEM problem can be recast as,

$$\begin{aligned} \mathcal{P}2: \max_{\{v_{ij}\}} \; &\sum_{i\in\mathcal{N}} q_i r_i - \lambda^* \sum\nolimits_{i\in\mathcal{N}} \sum\nolimits_{j\in\mathcal{L}_i} \|v_{ij}\|_2^2 \\ &- \lambda^* \sum\nolimits_{i\in\mathcal{N}} \sum\nolimits_{j\in\mathcal{L}} P_j \left\| \|v_{ij}\|_2^2 \right\|_0 \end{aligned} \tag{3.18a}$$

$$\text{s.t.} \quad \text{(3.15b)} \sim \text{(3.15d)}, \tag{3.18b}$$

where $q_i = (1 - \lambda^* w_i G_i \delta)$ after assuming $\varphi(r_i) = \delta r_i$, and $\delta > 0$ is a constant. Note that the term $\lambda^* \sum_{k\in\mathcal{N}} \mathcal{E}_i^S + \lambda^* \sum_{i\in\mathcal{N}} w_i(I_i^{snr} + D_i)$ in (3.17) is a constant and has been dropped in the objective function (3.18a).

In problem $\mathcal{P}2$, one of the challenges is the l_0-norm constraint (3.15d). Two commonly used approaches to cope with the l_0-norm are smoothing function approximation [91] and reweighted l_0-norm approximation [78, 92]. However, if the left-hand-side of (3.15d) is a relaxed smooth function or l_0-norm approximations, and the relaxed problem is directly solved, then we cannot guarantee that the obtained solution is also feasible from the problem

$\mathcal{P}2$. To overcome these issues, we reformulate problem $\mathcal{P}2$ as follows.

$$\begin{aligned} \mathcal{P}3: \max_{\{v_{ij}\}} \ & \sum\nolimits_{i\in\mathcal{N}} q_i r_i - \lambda^* \sum\nolimits_{i\in\mathcal{N}} \sum\nolimits_{j\in\mathcal{L}_i} \|v_{ij}\|_2^2 \\ & - \lambda^* \sum\nolimits_{i\in\mathcal{N}} \sum\nolimits_{j\in\mathcal{L}} (P_j + \psi_j) \left\| \|v_{ij}\|_2^2 \right\|_0 \end{aligned} \tag{3.19a}$$

$$\text{s.t.} \quad \text{(3.15b) ~ (3.15c)}, \tag{3.19b}$$

where $\psi_j \geq 0$ is the cost for RRH j. Let $\Psi = [\psi_1, \psi_2, ..., \psi_L]$. We denote $v_{ij}(\Psi)$ as the optimal solution for problem $\mathcal{P}3$ for a given cost vector Ψ. The following theorem establishes the relationship between problem $\mathcal{P}2$ and problem $\mathcal{P}3$,

Theorem 1. *The optimal solution to problem $\mathcal{P}3$ is also optimal to problem $\mathcal{P}2$ if,*

$$I_j(\Psi) = \sum\nolimits_{i\in\mathcal{N}} \left\| \|v_{ij}(\Psi)\|_2^2 \right\|_0 = B_j, \forall j \in \mathcal{L}. \tag{3.20}$$

Proof. For convenience, we define $I_j = \sum_{i\in\mathcal{N}} \left\| \|v_{ij}\|_2^2 \right\|_0$, $\widetilde{R} = \sum_{i\in\mathcal{N}} q_i r_i$, and $Z = \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{L}_i} \|v_{ij}\|_2^2$ to be used in the following proof.

Let $I_j(\Psi) = B_j$, which implies that $v_{ij}(\Psi)$ is also a feasible solution for $\mathcal{P}2$. Since v_{ij}^* and $v_{ij}(\Psi)$ are optimal solution for $\mathcal{P}2$ and $\mathcal{P}3$, respectively, we have,

$$\begin{aligned} & \widetilde{R}(\Psi) - \lambda^* \left(Z(\Psi) + \sum\nolimits_{j\in\mathcal{L}} (P_j + \psi_j) I_j(\Psi) \right) \\ & \leq \widetilde{R}^* - \lambda^* \left(Z^* + \sum\nolimits_{j\in\mathcal{L}} (P_j + \psi_j) I_j^* \right) \\ & \leq \widetilde{R}^* - \lambda^* \left(Z^* + \sum\nolimits_{j\in\mathcal{L}} P_j I_j^* + \sum\nolimits_{j\in\mathcal{L}} \psi_j B_j \right) \\ & \leq \widetilde{R}(\Psi) - \lambda^* \left(Z(\Psi) + \sum\nolimits_{j\in\mathcal{L}} P_j I_j(\Psi) + \sum\nolimits_{j\in\mathcal{L}} \psi_j B_j \right) \end{aligned} \tag{3.21}$$

where the first inequality is due to $v_{ij}(\Psi)$ is the optimal solution for problem $\mathcal{P}3$, the second inequality is based on constraint (3.15d) in problem $\mathcal{P}2$, the third inequality is due to v_{ij}^* is the optimal solution for $\mathcal{P}2$ and $v_{ij}(\Psi)$ is a feasible solution for $\mathcal{P}2$. After substituting B_j with $I_j(\Psi)$ into the right hand side of the third inequality above, we can have,

$$\begin{aligned} & \widetilde{R}(\Psi) - \lambda^* \left(Z(\Psi) + \sum\nolimits_{j\in\mathcal{L}} P_j I_j(\Psi) \right) \\ & = \widetilde{R}^* - \lambda^* \left(Z^* + \sum\nolimits_{j\in\mathcal{L}} P_j I_j^* \right) \end{aligned} \tag{3.22}$$

Hence, the theorem is proved. □

In practice, many factors contribute to the data rate of the fronthaul, which depends on the cell and fronthaul configurations. In our programmable C-RAN testbed, which will be described in more details in Sect. 3.5.1, we can calculate the required fronthaul capacity as,

$$B[\text{bps}] = N_{ant} N_{sec} F_s W_{IQ} O_{cod} K_{com}, \tag{3.23}$$

where N_{ant} is the number of Tx/Rx antennas, N_{sec} is the number of sectors, F_s represents the sampling rate, W_{IQ} is the bit length of a symbol, O_{cod} is the ratio of transport protocol and coding overheads, and K_{com} is the compression factor.

According to Theorem 1, if a cost vector Ψ can be found so that (3.20) holds, then the optimal solution of $\mathcal{P}2$ can be obtained by solving $\mathcal{P}3$. Otherwise, the solution of $\mathcal{P}3$ is suboptimal to $\mathcal{P}2$ if,

$$I_j(\Psi) \leq B_j, \forall j \in \mathcal{L}. \tag{3.24}$$

We present the following proposition that allows us to adjust via iterations the cost vector Ψ in $\mathcal{P}3$ such that the inequality in (3.24) holds.

Proposition 1. *By fixing ψ_k as a constant $\bar{\psi}_k, \forall k \in \mathcal{L}\backslash\{j\}$, and letting $\bar{\Psi}_j = \left[\bar{\psi}_1, .., \bar{\psi}_{j-1}, \psi_j, \bar{\psi}_{j+1}, ..., \bar{\psi}_L\right]$, the following arguments hold: (i) $I_j(\bar{\Psi}_j)$ is a non-increasing function w.r.t. ψ_j, (ii) There is a threshold cost for RRH j, given by,*

$$S_j = \sum_{i \in \mathcal{N}} r_i^{min} - \lambda^* \left\{ \sum_{j \in \mathcal{L}} \left(P_j^{max} + P_j B_j \right) + \sum_{k \in \mathcal{L}\backslash j} \bar{\psi}_k B_k \right\}$$

such that for $\psi_j \geq S_j$, $I_j(\bar{\Psi}_j) \leq B_j$.

Proof. The proof is similar with one provided from [82]. We present it for completeness as follows. For convenience, we define $I_j = \sum_{i \in \mathcal{N}} \left\| \|v_{ij}\|_2^2 \right\|_0$, $\widetilde{R} = \sum_{i \in \mathcal{N}} q_i r_i$, and $Z = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{L}_i} \|v_{ij}\|_2^2$ to be used in the following proof.

Let $\bar{\Psi}' = [\bar{\psi}'_1, ..., \bar{\psi}'_j, ..., \bar{\psi}'_L]$ be a different cost vector from $\bar{\Psi}_j$, such that $\psi'_j \geq \psi_j$ and $\bar{\psi}'_k = \bar{\psi}_k, \forall k \in \mathcal{L}\backslash\{j\}$. We have,

$$\begin{aligned}
&\widetilde{R}(\bar{\Psi}_j) - \lambda^* Z(\bar{\Psi}_j) - \lambda^* \sum\nolimits_{j\in\mathcal{L}} P_j I_j(\bar{\Psi}_j) - \lambda^* \psi_j I_j(\bar{\Psi}_j) \\
&\quad - \lambda^* \sum\nolimits_{k\in\mathcal{L}\setminus\{j\}} \bar{\psi}_k I_k(\bar{\Psi}_k) \\
&\le \widetilde{R}(\bar{\Psi}'_j) - \lambda^* Z(\bar{\Psi}'_j) - \lambda^* \sum_{j\in\mathcal{L}} P_j I_j(\bar{\Psi}'_j) - \lambda^* \psi_j I_j(\bar{\Psi}'_j) \\
&\quad - \lambda^* \sum_{k\in\mathcal{L}\setminus\{j\}} \bar{\psi}_k I_k(\bar{\Psi}'_k) \qquad \text{and,}
\end{aligned} \tag{3.25}$$

$$\begin{aligned}
&\widetilde{R}(\bar{\Psi}'_j) - \lambda^* Z(\bar{\Psi}'_j) - \lambda^* \sum\nolimits_{j\in\mathcal{L}} P_j I_j(\bar{\Psi}'_j) - \lambda^* \psi'_j I_j(\bar{\Psi}'_j) \\
&\quad - \lambda^* \sum\nolimits_{k\in\mathcal{L}\setminus\{j\}} \bar{\psi}'_k I_k(\bar{\Psi}'_k) \\
&\le \widetilde{R}(\bar{\Psi}_j) - \lambda^* Z(\bar{\Psi}_j) - \lambda^* \sum_{j\in\mathcal{L}} P_j I_j(\bar{\Psi}_j) - \lambda^* \psi'_j I_j(\bar{\Psi}_j) \\
&\quad - \lambda^* \sum\nolimits_{k\in\mathcal{L}\setminus\{j\}} \bar{\psi}'_k I_k(\bar{\Psi}_k)
\end{aligned} \tag{3.26}$$

where the inequality in (3.25) is based on the assumption that $v_{ij}(\bar{\Psi}_j)$ is the optimal solution for $\mathcal{P}3$, and the inequality in (3.26) is based on the assumption that $v_{ij}(\bar{\Psi}'_j)$ is the optimal solution for $\mathcal{P}3$. Adding up both sides of the two inequalities above and simplifying it, we have,

$$\left(\bar{\psi}'_j - \bar{\psi}_j\right) I_j(\bar{\Psi}'_j) \le \left(\bar{\psi}'_j - \bar{\psi}_j\right) I_j(\bar{\Psi}_j)$$

Hence, the first statement is proved. We denote $\widehat{v}_{ij}$ as a feasible solution for $\mathcal{P}2$, whose feasible region is nonempty. Then, we have,

$$\begin{aligned}
&\widetilde{R}(\bar{\Psi}_j) - \lambda^* \left(Z(\bar{\Psi}_j) - \sum_{j\in\mathcal{L}} P_j I_j(\bar{\Psi}_j) - \psi_j I_j(\bar{\Psi}_j) - \sum_{k\in\mathcal{L}\setminus\{j\}} \bar{\psi}_k \widehat{I}_k \right) \\
&\le \widehat{\widetilde{R}} - \lambda^* \widehat{Z} - \lambda^* \sum\nolimits_{j\in\mathcal{L}} P_j \widehat{I}_j - \lambda^* \psi_j \widehat{I}_j - \lambda^* \sum\nolimits_{k\in\mathcal{L}\setminus\{j\}} \bar{\psi}_k \widehat{I}_k.
\end{aligned} \tag{3.27}$$

Then, we obtain,

$$\begin{aligned}
&I_j(\bar{\Psi}_j) - \widehat{I}_j \\
&\le \left\{ \widehat{\widetilde{R}} - \lambda^* \left(\widehat{Z} + \sum\nolimits_{j\in\mathcal{L}} P_j \widehat{I}_j + \psi_j \widehat{I}_j + \sum\nolimits_{k\in\mathcal{L}\setminus\{j\}} \bar{\psi}_k \widehat{I}_k \right) \right\} / \psi_j \\
&\le \left\{ \sum\nolimits_{i\in\mathcal{N}} r_i^{min} - \lambda^* \left[\sum\nolimits_{j\in\mathcal{L}} I_j^P + \sum\nolimits_{k\in\mathcal{L}\setminus\{j\}} \bar{\psi}_k B_k \right] \right\} / \psi_j,
\end{aligned} \tag{3.28}$$

where $I_j^P = P_j^{max} + P_j B_j$. Therefore, if $\psi_j \ge S_j$, then $I_j(\bar{\Psi}_j) - \widehat{I}_j \le 1$. Since $I_j(\bar{\Psi}_j)$ and $\widehat{I}_j$ are both integers, then we have $I_j(\bar{\Psi}_j) = \widehat{I}_j \le S_j$. The proof is complete. □

Routine 2 Bisection Search for Finding the Cost Vector Ψ

1: Initialize: $\Psi^{(0)} = [0, ..., 0]$, $l = 1$
2: At iteration l, solve problem $\mathcal{P}3$ given $\Psi^{(l-1)}$ to obtain $I_j\left(\Psi^{(l-1)}\right), \forall j \in \mathcal{L}$
3: **if** $I_j\left(\Psi^{(l-1)}\right) \leq B_j, \forall j \in \mathcal{L}$ **then**
4: break
5: **else**
6: $\forall \widetilde{j} \in \mathcal{A}^{(l)} \triangleq \left\{j : I_j\left(\Psi^{(l-1)}\right) \geq B_j\right\} \forall j \in \mathcal{L}$, set $\psi_{\widetilde{j}}^{(l)} = S_{\widetilde{j}}^{(l)}$
7: Fix $\psi_k^{(l)} = \psi_k^{(l-1)}, \forall k \in \mathcal{L} \backslash \mathcal{A}^{(l)}$
8: $l = l + 1$, go to step 2

Recall that feasible region of $\mathcal{P}3$ is nonempty. Therefore, we can always find $\psi_j \in [0, S_j]$ that satisfies (3.24) by using the bisection method, as elaborated in Routine 2. In each iteration in Routine 2, step 2 involves solving problem $\mathcal{P}3$ which is still NP-hard due to the l_0-norm in the objective function and the non-convex constraint (3.15c). Observe that the phase of v_{ij} does not have impact on the optimality of the solution or the constraints, and thus, we can assume that each term $h_{ij}v_{ij}$ has a zero imaginary part. Then, we can rewrite constraint (3.15b) as an equivalent Second Order Cone (SOC) constraint, expressed as,

$$\sqrt{\sum_{k \in \mathcal{N} \backslash \{i\}} \left| \sum_{j \in \mathcal{L}} h_{ij}^H v_{kj} \right|^2 + {\sigma_i}^2} \leq \tau_i \operatorname{Re}\left(\sum_{j \in \mathcal{L}} h_{ij}^H v_{ij} \right), \tag{3.29}$$

in which $\tau_i = \sqrt{1 + 1/\left(2^{\frac{r_i^{min}}{B}} - 1\right)}, \forall i \in \mathcal{N}$, is the equivalent QoS threshold for UE i. Note that, after such transformation, the NEEM problem is still non-convex because of the l_0-norm in (3.15d). The non-convex l_0-norm function can be relaxed to a convex l_1-norm re-weighted [92] as,

$$\sum\nolimits_{i \in \mathcal{N}} \left\| \|v_{ij}\|_2^2 \right\|_0 \triangleq \sum\nolimits_{i \in \mathcal{N}} \alpha_{ij} \|v_{ij}\|_2^2 \leq B_j, \forall j \in \mathcal{L}, \tag{3.30}$$

where α_{ij} is the weight of the fronthaul link of RRH j and is updated iteratively as,

$$\alpha_{ij} = g\left(\|v_{ij}\|_2^2, \mu\right) = \frac{\eta}{\|v_{ij}\|_2^2 + \mu}, \forall i \in \mathcal{N}, j \in \mathcal{L}. \tag{3.31}$$

In 3.31, $\{v_{ij}\}$ is the beamformer from the previous iteration, μ is a small positive factor to

ensure stability and can be set as $\mu = 10^{-10}$ [78], and η is a constant. The idea behind the heuristic weighted update in (3.31) is that the RRHs having lower transmit power to UE i would have higher weights and would therefore be forced to further reduce the transmit power and encouraged to drop out of the RRH cluster eventually. Finally, problem $\mathcal{P}3$ can be transformed to,

$$\mathcal{P}4: \max_{\{v_{ij}\}} \sum\nolimits_{i\in\mathcal{N}} q_i r_i - \widetilde{\alpha}_{ij} \sum\nolimits_{i\in\mathcal{N}} \sum\nolimits_{j\in\mathcal{L}_i} \|v_{ij}\|_2^2 \tag{3.32a}$$

$$\text{s.t.} \quad (3.29), (3.15c), \tag{3.32b}$$

where $\widetilde{\alpha}_{ij} = \lambda^*(1 + \alpha_{ij}(P_j + \psi_j))$. Problem $\mathcal{P}4$ can be seen as a beamforming design problem for Weighted Sum-Rate (WSR) maximization. Although $\mathcal{P}4$ is still non-convex, we can reformulate it as an equivalent WMMSE problem, which can be solved via an iterative algorithm in the next subsection.

3.4.3 Proposed Iterative Algorithm

Here, we extend the WMMSE approach proposed in [93] to solve problem $\mathcal{P}4$. In particular, according to [93], problem $\mathcal{P}4$ is equivalent to the following problem,

$$\mathcal{P}5: \min_{u_i,\rho_i,v_{ij}} \sum_{i\in\mathcal{N}} q_i \left(\rho_i e_i - \log_2 \rho_i\right) + \widetilde{\alpha}_{ij} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{L}_i} \|v_{ij}\|_2^2 \tag{3.33a}$$

$$\text{s.t.} \quad (3.29), (3.15c), \tag{3.33b}$$

where ρ_i is the Mean Square Error (MSE) weight for UE i, and e_i is the corresponding MSE at UE i under the receiver beamformer $u_i \in \mathbb{C}$, i.e.,

$$\begin{aligned} e_i &= \mathbb{E}\Big\{ \left\| u_i^H y_i - x_i \right\|_2^2 \Big\} \\ &= \sum\nolimits_{i\in\mathcal{N}} u_i^H \left\{ \left| \sum\nolimits_{j\in\mathcal{L}} h_{ij}^H v_{ij} \right|^2 + \sigma_i^2 \right\} u_i \\ &\quad - 2\,\mathrm{Re}\left(u_i^H \sum\nolimits_{j\in\mathcal{L}} h_{ij}^H v_{ij} \right) + 1. \end{aligned} \tag{3.34}$$

The equivalent WMMSE minimization problem $\mathcal{P}5$ is convex with respect to each of the individual optimization variables. We outline the main steps for solving problem $\mathcal{P}5$ as

Algorithm 3 Iterative method for solving $\mathcal{P}4$

1: Initialize: $\alpha_{ij}, v_{ij}, r_i, \lambda^*, P_j, \psi_j \forall i \in \mathcal{N}, j \in \mathcal{L}$
2: **repeat**
3: For fixed v_{ij}, compute the MMSE receiver u_i and the corresponding MSE e_i according to (3.36) and (3.34), respectively
4: Update the MSE weight ρ_i according to (3.35)
5: For fixed u_i, and ρ_i, find the optimal transmit beamformer v_{ij} by solving (3.37)
6: Update α_{ij} as in (3.31) and $\widetilde{\alpha}_{ij} = \lambda^*(1 + \alpha_{ij}(P_j + \psi_j))$. Compute the achievable rate r_i as in (3.3)
7: **until** Convergence

follows.

- The optimal MSE weight ρ_i under fixed u_i and v_{ij} is,

$$\rho_i = e_i^{-1}. \tag{3.35}$$

- By fixing v_{ij} and ρ_i, the optimal receive beamformer u_i can be obtained as,

$$u_i = \frac{\sum_{j \in \mathcal{L}_i} h_{ij}^H v_{ij}}{\sum_{k \neq i} \left| \sum_{j \in \mathcal{L}_k} h_{ij}^H v_{kj} \right|^2 + \sigma_i^2}, \tag{3.36}$$

- Under fixed MMSE receiver u_i and optimal MSE weight ρ_i, the optimal transmit beamforming vector v_{ij} is calculated by solving the following problem,

$$\mathcal{P}6 : \min_{\{v_{ij}\}} \sum_{i \in \mathcal{N}} q_i \rho_i e_i + \widetilde{\alpha}_{ij} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{L}_i} \|v_{ij}\|_2^2 \tag{3.37a}$$

$$\text{s.t.} \quad (3.29), (3.15c). \tag{3.37b}$$

Problem $\mathcal{P}6$ above is a Quadratically Constrained Quadratic Programing (QCQP) problem, which can be solved by using standard convex optimization solvers. Finally, the suboptimal solution of $\mathcal{P}4$ is obtained using Algorithm 3.

Remark 1 (Convergence Analysis): Algorithm 3 relies on the iterative weighted updates in (3.31) to deactivate RRHs and reduce the fronthaul capacity for energy saving. Finding the convergence proof of α_{ij} under arbitrary reweighed function is challenging.

However, we show that, if the reweighing function is selected as,

$$g\left(\|v_{ij}\|_2^2, \mu\right) = \left(\left(\|v_{ij}\|_2^2 + \mu\right) \ln\left(1 + \mu^{-1}\right)\right)^{-1}, \tag{3.38}$$

where $\eta = \frac{1}{\ln(1+\mu^{-1})}$, then the l_0-norm relaxation can be seen as a special case of the Majorization-Minimization (MM) algorithms [94] and is guaranteed to converge.

Remark 2 (Complexity Analysis): The main computational complexity of the proposed solution for $\mathcal{P}4$ lies in Algorithm 3 which can be summarized as follows. The computational complexity of step 3 is approximately $\mathcal{O}(MLN^2)$ due to (3.36) and (3.34). In step 4, the computational complexity of updating the MSE weight ρ_i is approximately $\mathcal{O}(N)$ with the MSE e_i achieved from step 3. For step 5, the computational complexity can be approximately given as $\mathcal{O}\left(N(ML)^3\right)$. With the optimal beamforming from step 5, the computational complexity to achieve step 6 is approximately $\mathcal{O}(MLN^2)$. As it can be observed, the computational complexity of Algorithm 3 per iteration mainly comes from the process of solving the QCQP problem in step 5. Supposed that Routine 2 and Algorithm 3 require T_1 and T_2 total number of iterations to converge. Then, the overall computational complexity of the proposed solution for the NEEM problem is approximately $\mathcal{O}(T_1T_2N(ML)^3)$.

3.4.4 Beamforming Design via Branch-and-Bound

In the previous section, the sub-optimal solution for the $\mathcal{P}0$ problem is obtained by solving the relaxed NEEM optimization problem in $\mathcal{P}4$ using the proposed iterative WMME method in Algorithm 3 and solving the feasibility problem. We present now the Branch-and-Bound (BnB) method to solve the $\mathcal{P}4$ problem to a globally optimal solution. While the BnB method generally has very high computational complexity, which grows exponentially with the problem size, we mainly use the resulting solution to benchmark the suboptimality of our efficient iterative WMME solution. The BnB method presented in the following is an extension of the method in [95] for a Multiple Input Single Output (MISO) network with

non-cooperative BSs. Firstly, let us rewrite problem in $\mathcal{P}4$ in an equivalent form as,

$$\mathcal{P}4': \quad \min_{\{v_{ij}\},\{\gamma_i\}} \Lambda(\boldsymbol{\gamma}) + \Upsilon(\mathbf{v}) \tag{3.39a}$$

$$\text{s.t.} \quad (3.29),\ (3.15c), \tag{3.39b}$$

where $\Lambda(\boldsymbol{\gamma}) = \sum_{i\in\mathcal{N}} -q_i B \log_2(1+\gamma_i)$, $\Upsilon(\mathbf{v}) = \widetilde{\alpha}_{ij} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{L}_i} \|v_{ij}\|_2^2$, and $\boldsymbol{\gamma} = [\gamma_1, ..., \gamma_N]$.

The idea of the BnB algorithm is to generate a sequence of asymptotically tight *upper* and *lower* bounds for the objective function such that they both converge to a global optimal value. Specifically, the BnB algorithm divides the box region into smaller ones, and cuts off boxes that do not contain an optimal solution. The algorithm will converge to the global optimal solution after finite iterations. The algorithm starts with a known N-dimensional rectangle ϖ_{init} that contains the feasible region $\mathcal{G} = \{\gamma|\ (3.29),\ (3.15c)\}$, which can be specified as follows,

$$\varpi_{init} = \left\{\gamma | 0 \le \gamma \le P_j^{max}/\sigma_i^2 \sum\nolimits_{j\in\mathcal{L}} \|h_{ij}\|_2^2\right\}. \tag{3.40}$$

It is easy to verify that $\mathcal{G} \subseteq \varpi_{init}$. In each iteration, the lower and upper bounds are updated by partitioning ϖ_{init} into smaller rectangles. In order for the algorithm to converge, the bounds should be chosen such that they become tighter and tighter as the number of partitions of ϖ_{init} increases. The iterative BnB algorithm terminates when the difference between the upper and lower bounds is within a predefined accuracy level ϵ_t. For any rectangle $\varpi = \{\boldsymbol{\gamma}|\gamma_{i,min} \le \gamma_i \le \gamma_{i,max}, \forall i \in \mathcal{N}\}$ such that $\varpi \subseteq \varpi_{init}$, we define the functions to calculate the lower and upper bounds as $g_{lb}(\varpi)$ and $g_{ub}(\varpi)$, respectively.

For clarity, the BnB algorithm is summarized below (Algorithm 4). In this chapter, we use the bounding functions derived in [95], which can be expressed as,

$$g_{ub}(\varpi) = \begin{cases} \Lambda(\boldsymbol{\gamma}_{min}) & ,\boldsymbol{\gamma}_{min} \in \mathcal{G} \\ 0 & \text{otherwise;} \end{cases} \tag{3.41}$$

Algorithm 4 BnB Algorithm for solving $\mathcal{P}4'$

1: Initialize: ϖ_{init} using (3.40) and optimality tolerance $\epsilon_t > 0$.
2: Set $\bar{\varpi} = \varpi_{init}$, $\mathcal{B} = \{\bar{\varpi}\}$, $\boldsymbol{G}_{ub} = g_{ub}(\bar{\varpi})$, and $\boldsymbol{G}_{lb} = g_{lb}(\bar{\varpi})$.
3: **repeat**
4: Split ϖ along its longest edge into ϖ_I and ϖ_{II} using bisection subdivision.
5: Update $\mathcal{B} = (\mathcal{B}\backslash\{\varpi\}) \bigcup \{\varpi_I, \varpi_{II}\}$.
6: Set $\boldsymbol{G}_{ub} = \min_{\varpi\in\mathcal{B}}\{g_{ub}(\varpi)\}$, $\boldsymbol{G}_{lb} = \min_{\varpi\in\mathcal{B}}\{g_{lb}(\varpi)\}$, $\bar{\varpi} = \arg\min_{\varpi\in\mathcal{B}}\{g_{lb}(\varpi)\}$.
7: **until** $\boldsymbol{G}_{ub} - \boldsymbol{G}_{lb} \leq \epsilon_t$. Return $\bar{\varpi}$.

Routine 3 Bisection Search for Finding $\bar{\boldsymbol{\gamma}}$ for a Given ϖ

1: **for** $i = 1 : N$ **do**
2: Set $\boldsymbol{a} = \boldsymbol{\gamma}_{min}$, $\boldsymbol{a}[i] = \boldsymbol{a}[i] + \boldsymbol{\gamma}_{max}[i] - \boldsymbol{\gamma}_{min}[i]$.
3: **if** $a \in \mathcal{G}$ **then** Set $\bar{\boldsymbol{\gamma}}[i] = \bar{\boldsymbol{\gamma}}_{max}[i]$
4: **else** Set $\boldsymbol{\gamma}' = \boldsymbol{\gamma}_{min}$, $\boldsymbol{\gamma}'' = \boldsymbol{a}$, and tolerance $\epsilon_b > 0$. **end if**
5: **repeat**
6: Set $\boldsymbol{m} = (\boldsymbol{\gamma}' + \boldsymbol{\gamma}'')/2$ **if** $m \in \mathcal{G}$ **then** Set $\boldsymbol{\gamma}'' = \boldsymbol{m}$
7: **else** Set $\boldsymbol{\gamma}'' = \boldsymbol{m}$ **end if**
8: **until** $||\boldsymbol{\gamma}' - \boldsymbol{\gamma}''||_2 \leq \epsilon_b$. **return** $\bar{\boldsymbol{\gamma}}[i] = \boldsymbol{m}[i]$

and

$$g_{lb}(\varpi) = \begin{cases} \Lambda(\bar{\boldsymbol{\gamma}}_{min}) & ,\boldsymbol{\gamma}_{min} \in \mathcal{G} \\ 0 & \text{otherwise.} \end{cases} \tag{3.42}$$

In (3.42), $\bar{\boldsymbol{\gamma}} = [\bar{\gamma}_1, ..., \bar{\gamma}_i]$ can be obtained using bisection search on each edge of the rectangle ϖ as described in Routine 3. Let us define the optimal value of $\gamma \in \mathcal{G}$ for problem $\mathcal{P}4'$ as $\boldsymbol{\gamma}^* = \inf_{\boldsymbol{\gamma}\in\mathcal{G}}$. By using the bounding functions in (3.41) and (3.42), it is shown in [95] that Algorithm 4 will converge in finite number of iterations to a solution arbitrary close to $\boldsymbol{\gamma}^*$. It should be noted that verifying whether $\boldsymbol{\gamma} \in \mathcal{G}$, which is required in (3.41) and (3.42) as well as in steps 6 and 9 of Routine 3, is equivalent to solving a feasibility problem in order to determine if the SINR values specified by $\boldsymbol{\gamma}$ are achievable and, if so, return a set of feasible beamforming vectors $\mathbf{v}$'s.

3.4.5 Timescale and Real-time Scheduling Discussing

Observe that, the NEEM algorithm has been designed to dynamically maximize the network EE by taking into accout the radio resource allocation and capacity-limited fronthaul. One major concern may be the different operation timescales of the radio resource allocation and the fronthaul link. In current mobile networks, e.g., LTE, the Transmission Time

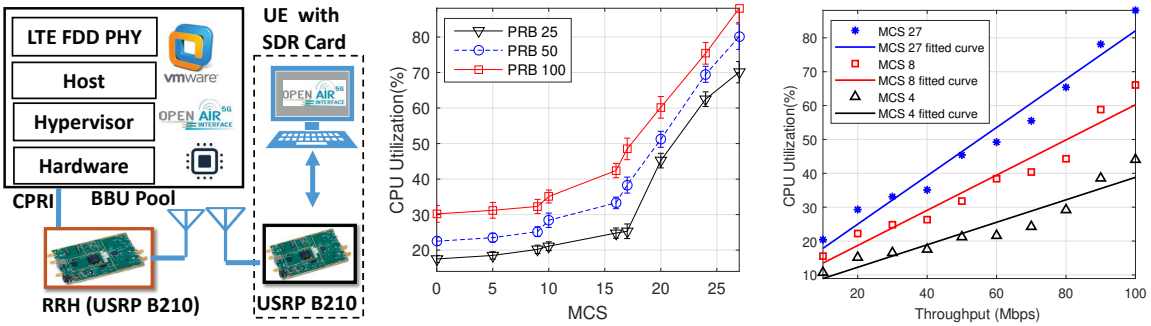
Interval (TTI) is 1ms [96]. Therefore, the resource allocation can be achived on a timescale of 1ms. However, making resource allocation on such a small timescale may lead to excessive computing and communication overheads. Moreover, the channel coherence time may be much larger than one TTI. Therefore, the TTI building technique, which enables the combination and scheduling of multiple consecutive subframes, is supported in the LTE system [96]. As a result, the practical operation timescale of radio resource allocation can be several milliseconds. Regarding the fronthaul, the Passive Optical Network (PON) emerges as a key fronthaul solution for future cloud-based radio access network [97]. In PON, the Optical Line terminal (OLT) is responsible for receiving the upstream data and broadcasting downstream data to optical network Units (ONUs). The ONU can be switched into the idle mode for energy savings when it does not carry any traffic load [98]. The idle mode in ONU operates at a timescale of several milliseconds [98]. Therefore, it is possible to operate the dynamic radio resource allocation and the fronthaul (ONU) idle mode using the same timescale.

3.5 Performance Evaluation

In this section, we present testbed experiments and simulation results to show the effectiveness of our solutions.

3.5.1 C-RAN Experimental Testbed

We have implemented a small-scale, real-world C-RAN testbed to understand the computational requirement in the BBU pool, which provides us the empirical model for (3.4) and the design of the NEEM algorithm. Figure 3.3(a) shows the architecture of our testbed. The RRH front-ends of the C-RAN testbed are implemented using SDR USRP B210s, each supporting 2×2 MIMO with sample rate up to 62 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 via USB 3 connections. Specifically, our C-RAN experimental testbed consists of one unit of UE and one unit of eNB, both implemented



(a) (b) (c)

Figure 3.3: (a) Illustration of C-RAN testbed architecture where the RRH is connected to the virtual BBU pool; (b) CPU utilization of the BBU at different values of MCS and PRB; and (c) Percentage of CPU usage versus downlink throughput.

using the USRP 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.

CPU Utilization: To understand the CPU utilization of the BBU pool, we establish a stable connection between eNB and UE of our testbed. The transmitted and received gain in downlink have been set to 90 and 125 dB respectively. 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. 3.3(b). 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, corresponding 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.

To understand better the BBU computational consumption in C-RAN with respect to the users' traffic demand, we characterize the relationship between the downlink throughput and the percentage of CPU usage at the BBU. We use OAI as a benchmarking implementation for profiling the CPU utilization of the LTE PHY layer given different load scenarios configured through PRBs, MCS, and SINR. The OAI downlink transmission supports 28 different MCSs with index I_{mcs} ranging from 0 to 27 characterized by: QPSK ($0 \leq I_{mcs} \leq 9$),

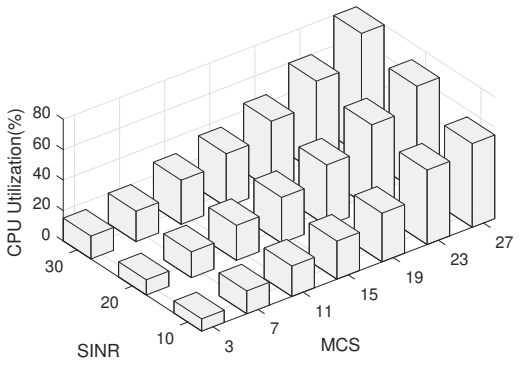


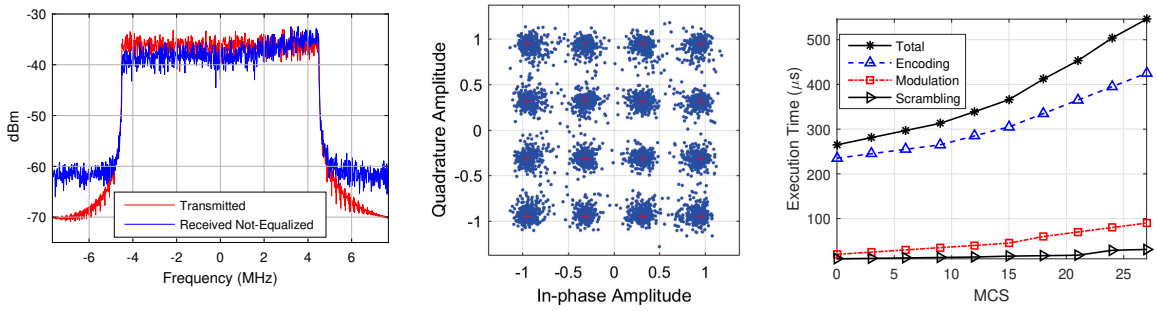
Figure 3.4: CPU utilization versus different MCS and SINR values.

16-QAM ($10 \leq I_{mcs} \leq 16$), and the rest are based on 64-QAM. Specifically, OAI has the *OpenAirLTEPhySimul* tool to run the Physical Downlink Shared Channel (PDSCH) and the Physical Uplink Shared Channel (PUSCH), respectively. Figure 3.3(c) illustrates the CPU utilization percentage at the BBU corresponding to different downlink throughputs and MCS indexes. Figure 3.4 illustrates the relationship between the CPU utilization and SINR under different values of the MCS. We can clearly observe that the CPU utilization increases with the values of SINR and MCS index. Using the measurement from Figs. 3.3(c) and 3.4, the CPU utilization percentage can be well fitted as a function of CPU frequency and MCS as,

$$\text{CPU}\% = I^{snr} + Gr_{th} + D, \tag{3.43}$$

where r_{th} is the achievable throughput, I_{snr} is parameter corresponding with SINR, and G and D are two parameters increasing with MCS values as reported in Table 3.1.

Figure 3.5(a) visualizes the frequency domain of transmitted and received OFDM signal by exploiting LTE System Toolbox 5G Library in OAI. It can be shown that the power spectral density of transmitted and received video (i.e, after channel modeling) signals have a located bandwidth 10 MHz. Figure 3.5(b) illustrates the constellation diagram of the received OFDM signal. Based on profiling results in Fig 3.3(c) and Fig 3.5(c), we note the following observations: we note the following observations: *i) the encoding execution time of downlink process load is dominating the other processing functions such as modulation and scrambling; ii) we can conclude that the total processing time and computing resources were*



(a) (b) (c)

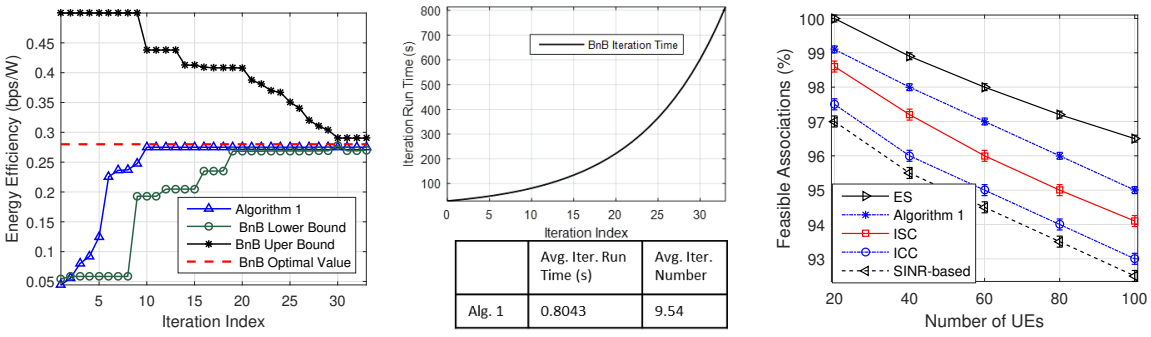
Figure 3.5: SDR over the air transmission/ reception of a data signal with 16 QAM LTE, PRB=50, and SINR=20 dBm; (a) Power spectral density of transmitted/received OFDM signal; (b) Constellations diagram of received OFDM signal; and (c) OFDM downlink processing time for different stage functions.

mainly spent on the encoding, and modulation, these tasks played the bigger roles in terms of complexity and runtime overhead in the BBU pool protocol stack; and iii) the percentage of CPU use at the BBU Pool is well approximated as a linear function of the downlink throughput.

3.5.2 Numerical Simulations

We present now simulation results to evaluate the performance of Algorithm 3. The simulations are carried out using optimization solvers (e.g., MOSEK) [67].

Simulation Setup. We consider a C-RAN system consisting of 6 hexagonal cells with a RRH in the center of each cell. The UEs are randomly located inside each cell so that distance between them and their nearest RRH is d or $d/2$. The distance between two nearest RRHs are $2d$ where $d = 500$ m. We assume that all the wireless channels in the system experience *block fading* such that the channel coefficients stay constant during each scheduling interval but can vary from interval to interval, i.e., the *channel coherence time* is not shorter than the scheduling interval. To reflect the mobility in our simulation, we run 100 channel realizations, and calculate the average performance. We assume that all the RRHs have the same number of transmit antennas $M = 4$ and maximum transmit power $P_j^{max} = P$, $\forall j$, while the maximum j-th fronthaul capacity number $B_j = 5$, $\forall j$. The static cost when RRH j is active is $P_j = 10$, $\forall j \in \mathcal{L}$, and the minimum data rate requirement for the i-th UE is $r_i^{min} = 5$ Mbps. We set the static energy consumption of a VM in



(a) (b) (c)

Figure 3.6: (a) Convergence comparison of of Algorithm 3 with BnB method; (b) Iteration run time comparison; and (c) Feasibility association versus different user numbers.

working mode as $\mathcal{E}_i^s = 20$W, and $G_i = D_i = w_i = 1$, $\forall i$. We adopt the distance-dependent path-loss model, given as L_p [dB] $= 148.1 + 37.6 \log_{10} \mathrm{d}_{[\mathrm{Km}]}$, and the log-normal shadowing variance set to 8 dB. In addition, the bandwidth B is set to 10 MHz and the noise power is -100 dBm.

Convergence of Algorithm 3. Firstly, we evaluate the performance of Algorithm 3 in yielding the WMMSE-based solution, compared to that of the optimal BnB method in Algorithm 4. Since the computational complexity of the BnB method is high, it is difficult to solve the NEEM problem with a large number of variables. Hence, we carry out the compassion in a small network with $N = 4$ users uniformly placed in the area covered by $L = 4$ cells and $P = 10$ dBm. In Fig. 3.6(a), we generate one random channel realization and set the same initial point for Algorithm 3. We observe that the objective value obtained from Algorithm 3 is very close to the optimal value obtained via BnB method. Iteration run time until convergence of the BnB algorithm for one random channel realization is depicted in Fig. 3.6(b). Additionally, we plot here the average iteration number and iteration run time of Algorithm 3 over 100 random channel realizations. It can be seen that the BnB method takes extremely long time to converge and is clearly not a practical solution. The solution of Algorithm 3 converges faster than the BnB algorithm both in terms of average number of iterations and average iteration run time. This fast-convergence performance is very important for the practical feasibility of NEEM since we want to optimize the beamforming design in each iteration.

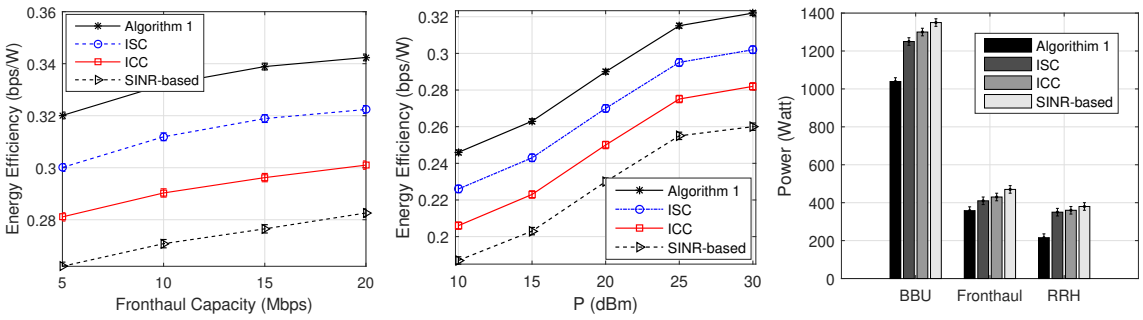


Figure 3.7: Simulations on a C-RAN network with $N = 25$, $L = 4$, and $M = 4$ (a) Network EE versus fronthaul capacity with $P = 10$dBm; (b) Network energy efficiency versus transmit power with optimality tolerance $\epsilon = 10^{-3}$; and (c) Network power consumption with $P = 10$dBm.

Optimality of Algorithm 3. Note that the association relationship between the RRHs and the UEs is obtained when $\|v_{ij}\|_2^2 > 0$. Hence, the beamforming vectors, or RRH-UE associations, can be achieved by our proposed Algorithm 3. To show the optimality of our proposed Algorithm 3, we compare four algorithms: i) *Exhaustive Search (ES)*: This scheme aims at finding the best association by searching all possible RRH-UE associations with extremely high complexity $\mathcal{O}((MNL)^3 2^L)$. ii) *Iterative Static Clustering (ISC)*: This scheme is proposed in [78], which is a heuristic algorithm used to obtain the UE set $\mathcal{N}_j$ that association with RRH j such that $\left\|v_{ij}^*\right\|_2^2 > 0, \forall i \in \mathcal{N}_j$ and $\left\|v_{ij}^*\right\|_2^2 = 0, \forall i \in \mathcal{N}_j^c$, where $\mathcal{N}_j^c$ is the complementary set of $\mathcal{N}_j$. iii) *SINR-based scheme*: This scheme is similar to work in [68], which considers the users associated with the RRH that provides the maximum SINR, i.e., the RRH associates with UE i is determined by $\mathcal{L}_i = \arg\max_j \|h_{ij}\|_2$, $\forall i \in \mathcal{N}$. iv) *Iterative Closest Clustering (ICC)*: This is similar to [69] where UEs are associated with nearest RRHs and the BBU pool allocates to all UEs with same computing power.

Figure 3.6(c) shows a percentage of feasible association, i.e., the number of UEs associated with certain RRH when their QoS requirements, transmitted power, and maximum fronthaul capacity number B_j, are satisfied. It can be seen that our proposed Algorithm 3 shows better performance compared to all the other algorithms, except the optimal search scheme ES. Observe that the percentage of satisfied users decreases while the number of users increases for all five algorithms.

Impact of Fronthaul Capacity. Fig. 3.7(a) illustrates the performance of the network

EE for various values of maximum fronthaul capacity for Algorithm 3 compared with ISC, ICC, and SINR-based algorithms. To quantify the impact of the fronthaul capacity limit, we set the such capacity between 5 and 20 Mbps. We observe that the network EE improves as the fronthaul capacity increases for the all algorithms. The results also show that the network EE reaches its maximum value when the fronthaul capacity is sufficiently large, e.g., 20 Mbps in our simulation setup. By increasing the fronthaul capacity, the corresponding constraint is loosen, resulting in a better EE and, eventually, eliminates the impact of fronthaul capacity. Observe that our algorithm significantly outperforms the others. Also, after reaching a certain value, the fronthaul capacity does not affect the EE anymore.

Energy Efficiency Performance. Figure 3.7(b) illustrates the network EE performance versus different values of transmit power P for different algorithms. It can be seen that the achieved EE significantly increases with the transmit power for the all algorithms. However, our algorithm shows better performance compared to the others. We observe that, when the traffic load in the network is light, the overall power consumption dominates over the data transmission rate. Thus, the network has a low EE. On the other hand, when the number of users increases, the network is saturated with traffic loads. Thus, the data transmission rate dominates over the overall power consumption and the EE improves.

Additionally, Fig. 3.7(c) depicts the power consumption in C-RAN network. Since the BBU pool carries the baseband signal processing, a large number of UEs lead to heavy computing workloads in the BBU pool. Therefore, the power consumption of the cloud platform increases significantly with the number of UEs. This indicates that the energy consumption of the BBU pool is closely related to the traffic load in the network. On the other hand, the energy consumption of the RRHs is much less than that of the BBU pool because the functionality of the RRHs is limited to basic RF signal processing. Hence, our algorithm performs better than the other algorithms.

Generalization to Multi-antenna UEs. Our proposed method in Algorithm 3 can be generalized to the scenario where each UE has multiple antennas. Since we consider a downlink system, we focus particularly on the number of receiving antennas on each UE i, denoted as F_i. In this case, one only needs to replace the channel gain vector and the received signal at UE i in (3.1) with $h_{ij} \in \mathbb{C}^{M \times F}$ and $y_i \in \mathbb{C}^{F \times 1}$, respectively. Accordingly,

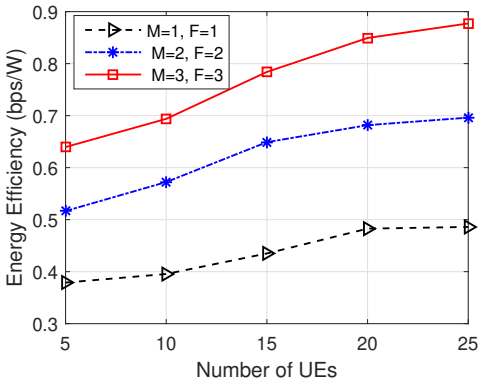


Figure 3.8: Network EE with different number of UEs number of antennas $P = 10$dBm.

the receive beamformer in (3.34) will be replaced with $u_i \in \mathbb{C}^{F\times 1}$. With this increase in dimensionality of the variables, the complexity of Algorithm 3 will also increase. Specifically, the complexities of steps 3 and 6 now become $\mathcal{O}(FMLN^2)$ and $\mathcal{O}(FMLN^2)$, respectively.

To evaluate the effect of having more antennas on UEs, we show the EE performance of Algorithm 3 in Fig. 3.8 with different number of antennas on each UE and on each RRH as well as with different number of UEs. As it can be observed, having multiple antennas on UEs and RRHs can lead to a better EE performance since it helps achieve a proper balance between the capacity of fronthaul and the required QoS. Additionally, it can be seen that the EE increases with the number of UEs. The reason for this behaviour is that, if only few users are in the system, the traffic load in the network is light, thus the static power consumption dominates the overall power consumption of the network. Therefore, the network has a low energy efficiency. When the number of users increase, the network is saturated with the traffic load, which results in higher EE performance.

3.6 Summary

We studied the Network Energy Efficiency Maximization (NEEM) problem in a Cloud Radio Access Network (C-RAN) taking into account practical constraints including QoS requirement, transmit power, and fronthaul capacity limits. Based on real-world data collected from a small-scale, real-time C-RAN testbed, we established an empirical model for the network power consumption. We formulated the NEEM optimization problem as a MINLP

that jointly considers the tradeoff between the network accumulated data rate and network power consumption. To deal with the non-convex nature of the problem, we took advantage of the l_1-norm reweighed method and successive convex approximation techniques. Such approximation allowed us to reformulate the original problem into the equivalent Weighted Sum-Rate (WSR) maximization problem, which can be solved efficiently and with proven convergence via iterations. We evaluated the performance of this iterative algorithm under different network conditions. Extensive simulations coupled with testbed experiments showed that the proposed resource allocation solution optimizes C-RAN energy efficiency under practical physical constraints while significantly outperforming existing approaches.

Chapter 4

On-demand Video-streaming in Mobile-Edge Computing

Mobile-Edge Computing (MEC) has recently emerged as a promising paradigm to enhance the mobile networks by providing cloud-computing capabilities to the edge of the Radio Access Network (RAN) with the deployment of MEC servers right at the Base Stations (BSs). Meanwhile, in-network caching and video transcoding have become important complementary technologies to lower network cost and enhance Quality of Experience (QoE) for video-streaming users. In this chapter, we aim at optimizing the QoE for dynamic adaptive video streaming that takes into account the Distortion Rate (DR) characteristics of videos and the coordination among MEC servers. Specifically, the Video-streaming QoE Maximization (VQM) problem is cast as a Mixed-Integer Nonlinear Program (MINLP) that jointly determines the integer video resolution levels and video transmission data rates. Due to the challenging combinatorial and non-convex nature of this problem, the Dual-Decomposition Method (DDM) is employed to decouple the original problem into two subproblems, which can be solved efficiently using standard optimization solvers. Real-time experiments on a wireless video streaming testbed have been performed on a FDD downlink LTE emulation system to characterize the performance and computing resource consumption of the MEC server under various conditions. Emulation results of the proposed strategy show significant improvement in terms of users' QoE over traditional approaches.

4.1 Introduction

Due to the ever-advancing multimedia processing capabilities on mobile devices and the plethora of Over-The-Top (OTT) video content providers, on-demand video streaming traffic has become the major factor driving the burgeoning traffic demand in mobile networks. According to the prediction of mobile data traffic by Cisco, mobile video streaming will

account for 72% of the overall mobile data traffic by 2019 [99]. The increase in demand for ubiquitous of High-Definition (HD) videos (e.g., 720p, 1080p, and beyond) requesting at least 5–20 Mbps user data rate [100], as well as, future video compression standards such as High Efficiency Video Coding (H.265/HEVC) [101] and the availability of 360-degree video devices may further fuel growth in mobile video traffic and bring great challenges for streaming and network operation. While such demands create immense pressure on mobile network operators, distributed *edge caching* has been recognized as a promising solution to bring video contents closer to the users, reduce data traffic going through the backhaul links and the time required for content delivery, as well as help in smoothing the traffic during peak hours. In wireless edge caching, highly sought-after videos are cached in the cellular Base Stations (BSs) so that that demands from users to the same content can be accommodate easily without duplicate transmissions from original content servers.

Recently, the network quality focus has changed from a network provider's Quality of Service (QoS) perspective to the less easily quantified end user's Quality of Experience (QoE) viewpoint [100]. However, the user diversity in terms of network conditions and device capabilities, which accompanies the worldwide prosperity of various video services, poses challenges for video service providers to achieve this enabler. For example, users with highly capable devices and fast network connection usually prefer high resolution videos while users with low processing capability or low-bandwidth connection may not enjoy high quality videos because the delay is large and the video may not fit within the device's display. By leveraging such behavior, Adaptive Bit Rate (ABR)-streaming techniques [102, 103] have been widely used to improve the quality of delivered video on the Internet as well as wireless networks. In ABR streaming, the quality (bitrate) of the streaming video is adjusted according to the user device's capabilities, network connection, and specific request. From an operator's perspective, it is essential to guarantee the QoE of mobile users and maximize network performance by assisting the users to make the network selection decision for the provided video streaming service. However, optimizing the video quality selection strategy for video streaming over multiple wireless networks—considering the video service's requirements, the wireless channel profiles and the costs of the different links— remains an open issue.

Recently, the Mobile Edge Computing (MEC) concept has emerged as a promising paradigm that enables a capillary distribution of cloud computing capabilities to the edge of the wireless access networks to provide better performance for certain applications as compared to cloud computing. This way, MEC allows for the execution of applications in close proximity to the end users, reducing end-to-end delay and backhaul bandwidth consumption. In this chapter, *we seek to design and analyze a MEC-based efficient mechanism that optimizes the QoE for on-demand video-streaming system with consideration of the Distortion-Rate (DR) properties of video streaming encoder.* We envision a framework to utilize both caching and transcoding processing, in which the MEC servers can perform transcoding of a video to different quality levels to satisfy the user requests. Each quality level is a bitrate version of the video and we consider that a lower bitrate level can be obtained from a higher bitrate variant via transcoding. Furthermore, *we seek to implement real-time over-the-air video streaming testbed that satisfies the computational requirements of the MEC server under the real-word video streaming considerations (e.g, transmission video modulation, encoder, and channel estimation).* Our over-the-air video streaming testbed will also help establising models to provide researchers with real-world insights and tools for designing QoE-aware algorithms in MEC systems.

4.2 Related Work

MEC has attracted a lot of attention from researchers over the last few years. In [104], the authors demonstrate the detailed definition and framework of edge computing, as well as the advantages of edge computing through several case studies. They conclude that edge computing has several potential benefits compared to traditional cloud-based computing paradigm, such as shorter response time and lower energy consumption. Three representative use-cases including mobile-edge orchestration, collaborative caching and processing, and multi-layer interference cancellation are introduced and studied in [105], which demonstrate that edge computing is crucial for enabling low-latency, high bandwidth, and agile mobile services.

The idea of using mobile edge caching to support the mobile communications has been studied in [35,106,107]. In [106], the authors introduce the design aspects and challenges of

mobile edge caching. Further, they reveal that caching at the 5G cellular networks is still an open problem due to the network topology, link interference, and user's mobility. The work in [107] proposes *FemtoCaching* architecture, in which a large number of dedicated *helper nodes* cache popular video files and serve the users' demands through local short-range links. Recently, Tran *et al.* [35] have suggested a mobile network topology combining caching and Cloud Radio Access Network (C-RAN), which comprises the edge-caches distributively deployed at the BSs and the cloud-cache deployed at a Central Processing Unit and all the cache entities are managed by a Central Cache Manager.

Valuable attention has been paid on video transcoding that relies on predicting channel conditions or users' requirements to enhance the system performance [108,109]. The authors in [110] provide a two-step load prediction method to scale video transcoding service through proactive resource allocation under real-time constraints, while the authors in [111] use a Markov prediction model to predict the next video segment requested by users.

As video analytics and caching can be enabled by MEC to further utilize the powerful computing resources of MEC servers at edge nodes, the works in [112–114] proposed to jointly combine the advantages of caching and transcoding to increase the throughput of mobile networks and QoE of users. In [112], a MEC server transcodes a video with higher rate version in the cache to satisfy a request for a lower rate version according to the optimization of the video rate adaptation and the network condition. To improve the users' QoE, the authors in [113] derive a logarithmic QoE model based on empirical results and formulate a cache management problem for adaptive streaming as a convex optimization problem, thereby providing an analytical framework for this engineering problem. Along with this line, the authors in [114] designed a scheme where multiple MEC caching and servers collaborate to provide video caching and transcoding. While the approach of maximizing the users' QoE has previously been considered by those works, our proposal is fundamentally different in two aspects. First, we introduce a novel Video-streaming QoE Maximization (VQM) framework to enhance on-demand video streaming system with proper consideration of the DR properties of versions from different videos. Second, our VQM optimization problem is formulated by considering a numerical model of MEC computation constraints based on carrying out several real-time experiments on a programmable

MEC testbed.

Main Contributions: Our approach aims to find an optimal video quality level for each user considering the limited network resources and the video cache available at the MEC server to maximize the overall QoE. The main objective of this chapter is to utilize the MEC paradigm in mobile network to enhance the video-streaming QoE. Overall, the contributions of this chapter can be summarized as follows.

- We formulate the design optimization problem, referred to as VQM problem, that aims at maximizing the QoE of the system while considering practical constraints such video data rate and computing resource at the MEC servers. The considered problem is cast as a Mixed-Integer Nonlinear Program (MINLP) which is NP-hard, and thus motivates us to design a low-complexity algorithm with practical implementation.

- In addition to video content popularity and network conditions that are commonly considered by existing caching schemes for adaptive video streaming, video content characteristics (i.e, the DR property) are further taken into account to assign different utilities to the representations with same bitrate but with different videos. In this way, the actual performance of the caching system is properly evaluated in terms of the users' viewing quality.

- To deal with the high complexity and non-convexity of the VQM problem, we utilize Lagrangian relaxation and Dual-Decomposition Method (DDM) to decompose the main problem into two sub-problems. The sub-problems are formulated such that they can be solved efficiently using standard optimization solvers (e.g., CVX, MOSEK). The simulations are conducted with different system configurations to show the effectiveness of the proposed scheme.

- Using SDR USRP boards and the Matlab LTE environment, we performe real-time experiments on an over-the-air video streaming testbed. Specifically, we establish transmissions between the eNodeB (eNB) and the User Equipment (UE) under various configurations in order to profile the run-time complexity and performance limits of the MEC server in terms of processing, throughput, and latency. It is shown that the

MEC server's CPU utilization can be modeled as a linear increasing function of the maximum downlink data rate.

Chapter Organization: The remainder of this chapter is organized as follows. Sect. 4.3 introduces the system model considered throughout this work. Sect. 4.4 describes the VQM optimization problem and proposes a distributed approach to solve it. We then present and discuss the architecture of our proposed programmable MEC testbed and simulation results in Sect. 4.5. Finally, the main conclusions are drawn in Sect. 4.6.

4.3 System Model

We first describe the system overview, followed by the setting of considered model. The QoE model is then mathematically formulated based on practical video parameters.

4.3.1 MEC System Architecture

As illustrated in Fig 4.1, the MEC server is geographically closer to the mobile users with high speed localized communication, which is usually assumed to be much faster than the backhaul links connected to the BS [115]. Furthermore, the MEC server mainly consists of Content Request Handler (CRH), computing/cache/transcoding unit, VQM controller, and Radio Network Information Service (RNIS). CRH is responsible for redirecting incoming video content requests to either the remote server (the origin server) or the local server (the MEC server). A VQM controller is built into the MEC server to collaborate with the computing/cache/transcoding unit and adjust users' QoE. The transcoding unit and VQM controller need to consider the channel conditions, so as to find the optimal bitrates to be transcoded. To this end, the transcoding unit and VQM controller need to collaborate with the RNIS, an intrinsic functional component of the edge computing paradigm [116]. Specifically, the RNIS is responsible for capturing up-to-date Channel State Information (CSI) and reporting back to transcoding unit and VQM controller.

In Fig. 4.2, we illustrated the possible events that happen when a user requests for a video. Whenever a mobile user sends a playback request for a specific video, it attempts to download the highest possible quality representation from its adjacent MEC server in

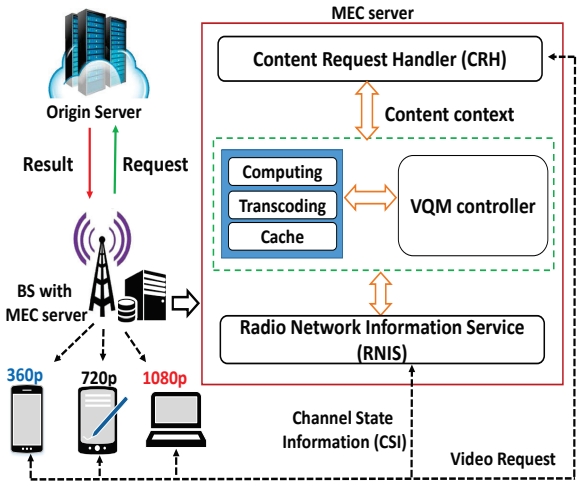


Figure 4.1: Illustration of collaborative video caching and transcoding framework deployed on a MEC network.

accordance with the content bitrate and the available download link capacity. If the same high quality video content is cached in MEC server, the user might want to download it from the MEC server with the highest transmission rate, in order to reduce the initial *startup delay*. That is, the user will first determine whether the downloading of highest resolution level can be supported by the link capacity with an acceptable downloading delay.

If yes, the user could download and play that video version; otherwise, it would make a further selection for the video content with the next lower bitrate. This process will be done by the video transcoder installed at the MEC server. When the requested video is not cached in the MEC server, the user has to turn to the origin server and download the video file with the highest bitrate that could be afforded by the backhaul link connected to the base station. However, retrieving from the origin server will result in a much more expensive transmission cost since the backhaul communication resource is typically very limited compared to the high-speed links offered by the MEC server.

4.3.2 System Setting

We now describe in more details the model considered in this work, and introduce the notations. We consider a MEC network with a set $\mathcal{N} = \{1, 2, ..., N\}$ of N mobile users randomly distributed in a cellular cell and one BS equipped with the MEC server, which has computation and cache capability that may be provided by Internet service providers as

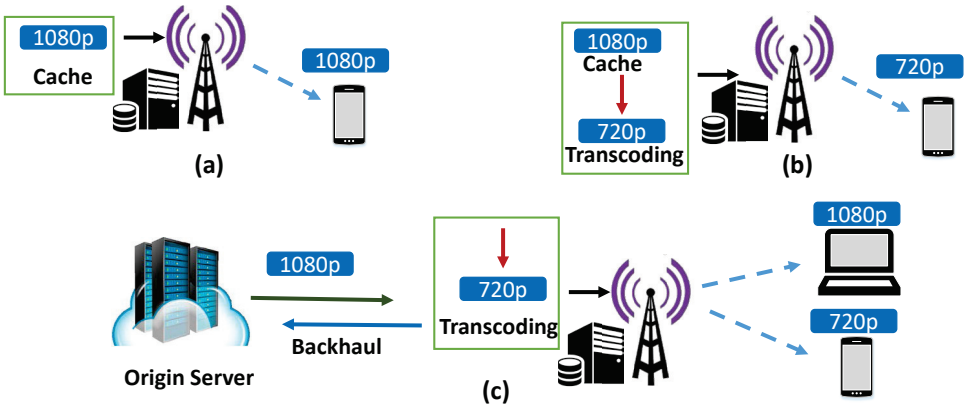


Figure 4.2: Illustration of possible events that happen when a user request for a video. (a) The video is obtained from cache of the MEC server; (b) A higher bitrate version of the video from cache of the MEC server is transcoded to the desired bitrate version and deliver to the user; and (c) The video is retrieved from the origin content server.

a value-added service. The MEC server has ability to store a set $\mathcal{F} = \{1, 2, .., F\}$ of F video files offered to the users. Each video file $f \in \mathcal{F}$ can be encoded into a set $\mathcal{M} = \{1, 2, ..., M\}$ of M different display resolutions having an encoding bitrate set $\mathcal{R} = \{R_{f1}, ..., R_{fm}\}$, $\forall f \in \mathcal{F}$, $m \in \mathcal{M}$. In practical, the MEC cache server caches the highest quality bitrate video file $f \in \mathcal{F}$, which can then be transcoded to lower bitrate versions. For the simplicity, and without loss of generality, similar to [117], we assume that the each video file has the same length T. We define a binary variable $y_{im} \in \{0, 1\}$ as the resolution indicator of user i, where $y_{im} = 1$ if the resolution level $m \in \mathcal{M}$ of the video is selected by user i; otherwise, $y_{im} = 0$. In a real-word video downlink LTE system, which will be described in more details in Sect. 4.5.1, 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 Modulation and Coding Scheme (MCS) index and the Physical Resource Block (PRB) allocated to the users as well as on the number of iterations required by the decoder, which is proportional to the Signal-to-Interference-plus-Noise Ratio (SINR) and channel conditions. Under these considerations, the CPU utilization of the MEC server increases linearly with the PRB resource and MCS index. Under this premise, we can consider the computation capacity C_i [cycles/s] from the MEC server that is allocated for user i, as a linearly increasing function of the user downlink data rate. Specifically, the computation capacity utilized for

processing the data of mobile user i can be modeled as,

$$C_i = G_i r_i + D_i, \forall i \in \mathcal{N}, \tag{4.1}$$

where G_i and D_i are positive constants estimated by offline profiling of the MEC testbed, r_i is the achievable video transmission rate for user i. The lower bound of r_i can be calculated as,

$$r_i \geq R_i^{min} = B \log_2(1 + \gamma_i), \forall i \in \mathcal{N}, \tag{4.2}$$

where R_i^{min} is minimum desired data rate of each user, B is the transmitted bandwidth to each user and γ_i is the received SINR calculated as,

$$\gamma_i = \frac{P^T L(d_i)}{\sigma_i^2 + I}, \forall i \in \mathcal{N}, \tag{4.3}$$

where P^T is the transmit power density at BS, d_i denotes the distance between user i and the BS, $L(d_i)$ is the path loss of the channel between the user i and BS, σ_i^2 is the additive Gaussian noise power density, and I denotes the inter-cell interference when a user is served by BS. With the advanced interference mitigation techniques such as spectrum reuse, power control, and interference alignment, the inter-cell interference can be treated as constant noise [118]. Then, I can be interpreted as the efficiency of interference mitigation. For example, $I = 0$ if the interference is canceled completely, which usually requires perfect channel information. Larger I means less effective interference mitigation, which can happen with incomplete channel information, lower received signal strength, or extensive spatial spectrum use [119]. In this work, we assume that the computing resource required for transcoding a video from the highest resolution to the requested resolution m is q_m. Obviously $q_m < q_{m'}$ if $m > m'$ and specially $q_M = 0$. Thus, the computing capacity constraint at the MEC server can be expressed as,

$$\sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} q_m y_{im} \leq \sum_{i \in \mathcal{N}} \beta_i^{cod} C_i, \tag{4.4}$$

where β_i^{cod} is the maximum video transcoding capacity assigned for user i and defined as the number of encoded bits that can be processed per second at the MEC server. For example, a 400 Mbps computing capacity means up to 400 Mbps videos can be transcoded in one second.

4.3.3 Quality-of-Experience (QoE) Model

The key factors that affect the QoE of video streaming services mainly depend on initial *startup delay* (i.e, the time interval between the user's request and beginning of playback) and the *average video quality* (i.e, the average distortion that may introduce in the video signal during the video processing) [120]. The initial *startup delay* constraint demands that the requested time interval between sending a request and the actual video playback should not exceed the maximum tolerable requesting time of the user denoted by τ_i^{max}. Hence, the initial *startup delay* experienced by the user i to view the the bitrate R_{fm} from the MEC server can be expressed as,

$$\tau_{fm}^{i,e} = \frac{R_{fm}\Delta T}{r_i}, \forall i \in \mathcal{N}, f \in \mathcal{F}, m \in \mathcal{M}, \tag{4.5}$$

where ΔT is the time fraction within a video file required to be buffered by the user before the actual playback starts on the client's screen. Similarly, when the request video is not available in the MEC cache server, the initial *startup delay* experienced by user i to request R_{fm} version from the origin server can be determined by,

$$\tau_{fm}^{i,o} = \frac{R_{fm}\Delta T}{r_i + c_{ifm}^o}, \forall i \in \mathcal{N}, f \in \mathcal{F}, m \in \mathcal{M}, \tag{4.6}$$

where c_{ifm}^o is the downlink transmission rate needed to download the requested video file from the origin server. We utilize the DR model in [121] to formulate the distortion of m-th representation of the video f with the encoding bitrate R_{fm}. Hence, the DR model can be formulated as,

$$\Delta D_f(R_{fm}) = D_{max} - D_0 - \frac{\phi}{R_{fm} - R_0}, \tag{4.7}$$

where D_{max} and $\Delta D_f(R_{fm})$ denote a constant maximal distortion when no video is decoded and the distortion reduction after successfully decoding this representation, respectively. The parameters ϕ, D_0, and R_0 are empirical variables depending on the actual video content, and they can be estimated as fitting parameters from the empirical DR curves of different videos by using regression techniques [121]. The term $D_{max} - \Delta D_f(R_{fm})$ in (4.7) denotes a RD function to model the distortion of the m-th resolutions of the video f with encoding bitrate R_{fm}.

4.4 Video-streaming QoE Maximization

In this section, we firstly formulate the VQM problem as a MINLP. We then propose a low-complexity approach that first relaxes the original VQM problem using DDM to decouple the optimization problem into two subproblems efficiently solved by a standard optimization solver.

4.4.1 System Utility Function

First, we introduce a binary variable a_f, where $a_f = 1$ indicates that the MEC server caches the video file f; and $a_f = 0$ otherwise. Then, we define the following utility function based on the average video distortion reduction experienced by the user i and the cost of the representation downloading either from the MEC server or the origin server.

$$\begin{aligned} U_{im}(R_{fm}) = &\sum_{f\in\mathcal{F}} a_f P_{if} \left\{\Delta D_f(R_{fm}) - \rho_e R_{fm}\right\} \\ &+ \sum_{f\in\mathcal{F}} (1-a_f) P_{if} \left\{\Delta D_f(R_{fm}) - \rho_o R_{fm}\right\}, \end{aligned} \tag{4.8}$$

where P_{if} is used to represent the average probability that the video file f is requested by the user i within this time period. The term $\{\Delta D_f(R_{fm}) - \rho_e R_{fm})\}$ in (4.8) represents the video distortion reduction $\Delta D_f(R_{fm})$ of downloading the bitrate R_{fm} of the video f, and a transmission cost penalty $\rho_e R_{fm}$ where ρ_e is unit price parameter corresponding to the video resolutions download from the MEC server.

Likewise, the second term in (4.8) includes the average distortion reduction plus the

average transmission cost penalty experienced by user i downloading requested video resolutions from the original server. The ρ_o is unit price parameter corresponding to the video representations download from the origin server. Due to the limited bandwidth available in the backhaul channel, the unit price for downloading from the origin server ρ_o is much higher than the unit price for accessing the adjacent edge servers ρ_e.

4.4.2 Problem Formulation

Our target is to find an optimal video quality level for each user with considering the limited network resources and the video cache availability in MEC server to improve the network utility by maximizing the overall QoE. The VQM optimization problem can be formulated as follows,

$$\mathcal{P}0 : \max_{\mathbf{y},\mathbf{r}} \sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}} U_{im}(R_{fm})y_{im} \tag{4.9a}$$

$$\text{s.t.} \sum_{f\in\mathcal{F}}\sum_{m\in\mathcal{M}} a_f R_{fm} T \leq S, \tag{4.9b}$$

$$a_f\tau_{fm}^{i,e} + (1-a_f)\tau_{fm}^{i,o} \leq \tau_i^{max}, \tag{4.9c}$$

$$\sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}} q_m y_{im} \leq \sum_{i\in\mathcal{N}} \beta_i^{cod}\left(G_i r_i + D_i\right), \tag{4.9d}$$

$$\sum_{m\in\mathcal{M}} y_{im} = 1, \forall i \in \mathcal{N}, \tag{4.9e}$$

$$r_i \geq R_i^{min}, \forall i \in \mathcal{N}, m \in \mathcal{M}, \tag{4.9f}$$

$$y_{im}, a_f \in \{0,1\}, \forall f \in \mathcal{F}, m \in \mathcal{M}, \tag{4.9g}$$

where $\mathbf{y}$ and $\mathbf{r}$ are vectors defined as $\mathbf{y} = \{y_{im} | m \in \mathcal{M}, i \in \mathcal{N}\}$, and $\mathbf{r} = \{r_i | i \in \mathcal{N}\}$, respectively. The constraints in $\mathcal{P}0$ can be explained as follows: constraint in (4.9b) represents the cache capacity constraints of the MEC server, where T is the time duration of each video file and S is cache capacity of the MEC server; the *startup delay* constraint in (4.9c) specifies that the initial startup delay experienced by the user i to download the requested video file either from the MEC server or the origin content server should not exceed the maximum tolerant waiting time τ_i^{max}; constraint (4.9d) requires the computing rate to be

smaller than or equal to the computing capacity on the MEC severer; constraint (4.9e) guarantees that only one resolution level is selected for the video requested by each user; finally, constraint (4.9f) ensures that the data rate of each user is above or equal to the desired data rate of each user. It can be seen that $\mathcal{P}_0$ is a MINLP, which is NP-hard [122] and therefore it is highly difficult to solve optimally in polynomial time. Furthermore, the complexity of solving $\mathcal{P}0$ by using greedy or genetic methods will increase significantly with the number of users and BSs. In future cellular networks, the density and number of small cells will rise significantly so that the size of optimization variables will become very large. To overcome these drawbacks, we propose to reformulate problem $\mathcal{P}0$ using relaxation techniques in the following subsections in order to derive a tractable, low-complexity solution.

4.4.3 Distributed-VQM Solution

In our proposed model, the video resolution level is determined by VQM optimization problem according to user playback parameters such as initial *startup delay* and the achievable data rate. Then, the server resources are assigned depending on the video requests. Therefore, the MEC network is required to provide some information to help users and servers to enhance the network performance through caching and transcoding. As shown in $\mathcal{P}0$, our goal is to design an efficient scheme to maximize the average QoE of all requested videos while considering the computing resource limitation. To deal with non-convexity of $\mathcal{P}0$, we adopt the DDM so that the the VQM problem can be decoupled to two subproblems, which can be efficiently solved by standard optimization solver. We define independent local feasible sets Γ_y, and Γ_d for variables $\mathbf{y}$, and $\mathbf{r}$, respectively. These feasible regions only subject to constraints that include one of the variables, which can be described as, $\Gamma_y = \{y_{im} | y_{im} \in \{0,1\}, i \in \mathcal{N}, m \in \mathcal{M}\}$ and $\Gamma_r = \{r_i | r_i \in \mathbb{R}^+, i \in \mathcal{N}\}$. The Lagrangian associated with $\mathcal{P}0$ can be calculated as,

$$L(\mathbf{y}, \mathbf{r}, \mu_i) = \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} U_{im}(R_{fm}) y_{im} + \sum_{i \in \mathcal{N}} \mu_i \Bigg[\sum_{m \in \mathcal{M}} q_m y_{im} - \beta_i^{cod} \left(G_i r_i + D_i \right) \Bigg], \forall f \in \mathcal{F}, \tag{4.10}$$

where μ_i is the Lagrangian multiplier. The dual problem is thus expressed as,

$$\min_{\mu_i \in \mathbb{R}^+} \; g(\mu_i) = g_y(\mu_i) + g_r(\mu_i), \tag{4.11}$$

where $g_y(\mu_i)$, and $g_r(\mu_i)$ are dual functions obtained as the maximum value of the Lagrangians solved as follows,

$$g_y(\mu_i) = \sup_{\mathbf{y} \in \Gamma_y} \left\{ \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} U_{im}(R_{fm}) y_{im} + \mu_i q_m y_{im} \right\} \tag{4.12}$$

$$g_r(\mu_i) = \sup_{\mathbf{r} \in \Gamma_r} \left\{ - \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} \mu_i \beta_i^{cod} (G_i r_i + Di) \right\} \tag{4.13}$$

It is obvious that $g(\mu_i)$ in (4.11) is not a differentiable function due to the binary variable $\mathbf{y}$. Thus, we can employ sub-gradient method to solve dual problem (4.11), which can be described as,

$$z_i^{\mu} = \sum_{m \in \mathcal{M}} q_m y_{im} - \beta_i^{cod} (G_i r_i + D_i), \tag{4.14}$$

where z_i^{μ} is a subgradient of the objective $g(\mu_i)$ for μ_i. According to dual decomposition [123], the parameters μ_i can be updated as,

$$\mu_i^{[t+1]} = \left[\mu_i^{[t]} - \rho_{\mu}^{[t]} z_i^{\mu} \right]^+, \tag{4.15}$$

where $\rho_{\mu}^{[t]}$ is the step length at iteration t, and $[x]^+ = max\{0, x\}$ denotes the projection function to the nonnegative orthant. Thus, if we can solve the inner problems (4.12) and (4.13) in each iteration, the controller at each server can collaborate to update dual variables and transfer them to the BSs and users to assist them to find optimal solutions of their own variables ($\mathbf{y}$ and $\mathbf{r}$).

Video Quality of Experience Selection. The target of this subproblem is to optimally set the video quality level, y_{im}. From (4.10), the the video QoE selection can be

expressed as,

$$\mathcal{P}1: \max_{\mathbf{y}\in\{0,1\}} \sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}} U_{im}(R_{fm})y_{im} + \mu_i q_m y_{im} \tag{4.16a}$$

$$\text{s.t.} \quad \text{(4.9b), (4.9e).} \tag{4.16b}$$

The structure of the objective function for $\mathcal{P}1$ can be described as follows. The parameter μ_i represents the bandwidth cost of user i given by the network, where $\mu_i > 0$ means the network can provide more bandwidth for the user i by using the cost to push the user to select higher resolution. However, if $\mu_i \leq 0$, the network may not have enough resource to support a higher video quality. Problem $\mathcal{P}1$ is still a MINLP due to the binary variable y_{im}. Several standard solvers (e.g., CVX, MOSEK) can solve it optimally by using the Branch-and-Bound (BnB) algorithm [124]. However, the computational complexity of BnB is prohibitive for a large network. For the worst case, 2^M iterations are required, thus the computational complexity is approximated as $\mathcal{O}\left(2^M(MN)^{3.5}\right)$, which grows exponentially with the number of BSs and UEs. Therefore, an efficient algorithm in proposed here below. To make $\mathcal{P}1$ tractable, we relax the video resolution binary variable y_{im} to a real-valued variable, i.e., $y_{mi} \in [0, 1]$. Thus, we have to recover it to a binary value after getting the sub-optimal solution. The basic idea of the rounding method is that we select the highest resolution for users under the current resource allocation solution. The recovering of y_{im} can be achieved as,

$$y_{im} = \begin{cases} 1 & \text{If } m = \arg\max\{D_{im} > 0\}, \\ 0 & \text{otherwise,} \end{cases} \quad \forall m \in \mathcal{M}, \tag{4.17}$$

where D_{im} is the first-order partial derivation of the objective function in $\mathcal{P}1$ with respect to y_{im}.

Algorithm 5 Iterative VQM algorithm for mobile user i

Initialize: $\mathcal{M}$, $\mathcal{F}$, R_{fm}, P_{if}, β_i^{cod}, μ_i, G_i, D_i, S, D_0, ϕ, R_0, τ_i^{max}, T, ΔT, R_i^{max}, c_{ifm}^o, μ_{ik}, $z_\mu^{[t]}$, ρ_e, ρ_o, ρ_μ, infinitesimal number ω and iteration index $t \leftarrow 1$, $\forall i \in \mathcal{N}$, $f \in \mathcal{F}$, $m \in \mathcal{M}$
for $i = 1 : N$ **do**
  while $t \leq I_{max}$, and $|\mu_i(t+1) - \mu_i(t)| > \omega$ **do**
    `//Video QoE Selection//`
    Calculate $D_f(R_{fm})$ from (4.7)
    Compute y_{im} from $\mathcal{P}1$
    `//Video Transmission Rate Scheduling//`
    Calculate $\tau_{fm}^{i,e}$ and $\tau_{fm}^{i,o}$ from (4.5) and (4.6), respectively
    Compute r_i from $\mathcal{P}2$
    `//Lagrangian Multiplier Update//`
    Update Lagrangian multiplier μ_i by (4.15)
    $t = t + 1$

Video Transmission Rate Scheduling. Similarly to the procedure used to generate $\mathcal{P}1$, the problem in (4.13) can be formulated as,

$$\mathcal{P}2 : \max_{\mathbf{r}} \quad -\sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}} \beta_i(G_i r_i + Di) \tag{4.18a}$$

$$\text{s.t.} \quad (4.9c), (4.9f). \tag{4.18b}$$

Notice that $\mathcal{P}2$ is a convex problem because the objective function and its constraint are affine [63]. Thus, it can be solved via standard convex optimization techniques. The proposed VQM algorithm is summarized in Algorithm 5, in which the time complexity of the algorithm for user i is $\mathcal{O}(NMI_{max})$, where I_{max} denotes the maximum number of iterations of the algorithm.

4.5 Performance Evaluation

In this section, we first detail the experimental setups and results for the programmable MEC testbed. Then, we present numerical simulation results to evaluate the performance of our proposed VQM algorithm.

4.5.1 Testbed Experiment

We present here our MEC testbed including the testbed architecture, configuration, and experiment methods. Finally, we analyze the performance of the MEC server, in terms of CPU processing time and latency.

Testbed Architecture. We conducted experiments on a testbed consisting of two main components, i.e.,

- *RAN*: We implement a RAN consisting of one LTE eNB and one User Equipment (UE) using SDR boards—the Ettus USRP B210's supporting 2×2 MIMO with sample rate up to 62 MS/s. The hardware architecture of the SDR platform consists of two-channel USRP devices with continuous RF coverage (70 MHz–6 GHz). Both eNB and UE are defined to Matlab LTE tools, which are deployed on the MEC server.

- *MEC server*: We utilize an Intel Xeon server, a Dell Precision T5810 workstation with Intel Xeon CPU E5-1650, 12-core at 3.5 GHz, and 32 GB RAM, fully configurable with real-time RF transmitting and receiving signals. To realize the LTE Medium Access Control (MAC) layer in the MEC server, we implement a dynamical scheduler to allocate the radio resources based on various scheduling priorities such as a Channel Quality Indicator (CQI), corresponding to a PRB or multiple PRBs in the form of MCS index to the BS. LTE specifications require that all PRBs allocated to the same user in any given Transmission Time Interval (TTI) must use the same MCS (1 ms).

We summarize the testbed configuration parameters in Table 4.1. In particular, the eNB is configured in band 7 (FDD) using a downLink 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 *findsdru* matlab function is used to find and report status for all connected USRP boards.

CPU Utilization. It is of critical importance to understand the CPU utilization of the MEC server in order to design efficient resource provisioning and allocation schemes. Therefore, we profile the physical layer of LTE to understand the relation between the processing load and number of allocated PRBs and MCS. We study this relationship by

Table 4.1: Testbed Configuration Parameters for eNB and UE.

Duplexing Mode	FDD	**Mobility**	Static
Frequency	2.66 GHz	**PRB**	$25, 50, 100$
Transm. power	150 dBm	**Rad. pattern**	Isotropic
MCS	$[0 \div 27]$	**SINR**	20

scheduling a single user for transmission or reception. We prepare a video file which has the length of 10 s and $720p$ resolution to transmit over the air by using MEC testbed. The video file is encoded to H.265 standard and then modulated to 16 QAM with OFDM scheme.

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. Then, based on the MCS index used in each experiment, we can calculate the corresponding downlink throughput by multiplying the bit rate by the number of bits in the modulation scheme [125]. The CPU utilization percentage has been recorded as in Fig. 4.3(a). We can conclude from Fig 4.3(a) that the percentage of CPU use at the MEC server is well approximated as a linear function of the downlink throughput, in which the CPU utilization can be fitted as,

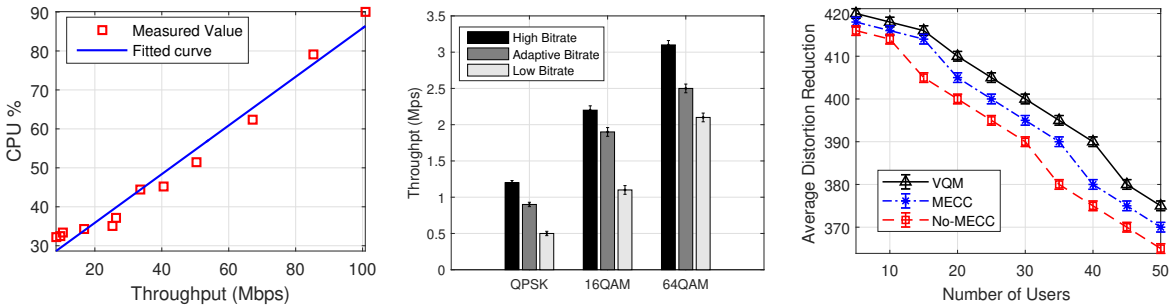
$$\text{CPU}\,[\%] = 0.6247\psi + 23.4, \tag{4.19}$$

where ψ is the throughput measured in Mbps.

Adaptive Video Streaming. In this experiment, we aim to evaluate the performance of the MEC server under the adaptive video streaming, in which the MEC server tries to transmit the video file to user according to its channel condition. To simulate the MEC-based adaptive bitrate streaming, we use two types of bitrate streams provided as: low bitrate at 700 Kbps, and high bitrate at 1600 Kbps. As the CQI is calculated from SINR [126], we use the parameters shown in Table 4.2 to give the transforming process of SINR-to-CQI mapping. The LTE System Toolbox product provides a set of channel models for the test and verification of UE and eNB radio transmission and reception as defined in [127]. In [128], the authors show that there is a linear relation between the CQI

Table 4.2: LTE downlink feedback parameters [1].

Modulation	**CQI**	**SINR** [dB]
QPSK	$1 \sim 6$	$-6.658 \sim 2.424$
16QAM	$7 \sim 9$	$4.487 \sim 8.456$
64QAM	$10 \sim 15$	$10.266 \sim 19.809$



(a) (b) (c)

Figure 4.3: (a) Percentage of CPU usage versus downlink throughput; and (b) Throughput versus modulation scheme for different video versions with 25 PRBs; and (c) Average distortion reduction per user with cache size $S = 120$ Mbps.

index and the actual SINR limits in [dB] as,

$$SINR_u[\text{dB}] = \text{uc}_1 + \text{c}_2, \forall \text{u} \in \mathcal{U} = \{1, ..., 15\}, \tag{4.20}$$

where c_1 and c_2 are constants. It is also shown that the actual range of the SINR limits in [dB] is determined by the following observations: SINR[dB] $= -6$ corresponds to CQI $= 1$, while SINR[dB] $= 20$ corresponds to CQI $= 15$. We then have $-6 = c_1 + c_2$ and $20 = 15c_1 + c_2$, and hence, $c_1 = 13/7$ and $c_2 = -55/7$. Therefore, we select three types of modulation schemes and their corresponding parameters are shown in Table 4.2. Figure 4.3(b) shows the performance of three video bitrate version with 25 PRBs for different modulation schemes. Note that with the increase of CQI index value, it is better to select adaptive bitrate technique to overcome the high throughput requirement at the user side.

4.5.2 Numerical Simulations

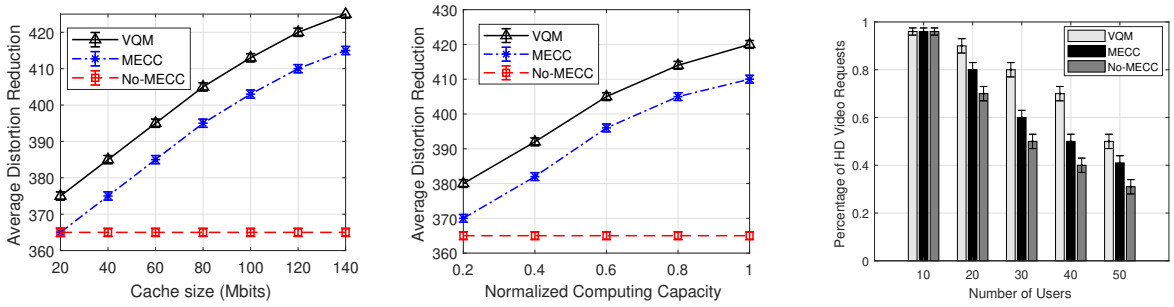
We present now simulation results to evaluate the performance of our proposed Algorithm. The simulations are carried out using a Matlab implementation with MOSEK solver [67].

Simulation Setup. We consider a MEC network with one MEC server deployed on a BS of a cellular RAN. The mobile users are randomly located inside each cell so that distance between them and their nearest BS is d or $d/2$. The distance between two nearest BS are 2d where $d = 500$ m. We assume that the BS has a single antenna and transmit power $P_T =$ 45 dBm, while the minimum data rate requirement for the i-th users is $R_i^{min} = 1.5$ Mbps. We use the distance-dependent path-loss model, given as $L(d)$ [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 is set to 5 MHz, the noise power is -100 dBm, and $\beta_i^{cod} = 200$ Mbps. We assume the MEC testbed parameters are set as $G_i = D_i = 1$. We assume the video library $\mathcal{F}$ that consists of $V = 4$ unique videos, each has the same length of $T = 10$ s and the time fraction with a video file $\Delta T = 1$s that is required to be buffered by the user before the actual playback starts on the user's screen. We assume the constant maximal distortion is set as $D_{max} = 500$ and each video file can be transcoded to $M = 3$ resolution levels with encoding rate being $\{3R, 2R, R\}$ where $R = 2$ Mbps. We assume that the storage capacity for the MEC server is set to $S = 120$ Mbits and them empirical parameters θ, R_0 and D_0 are set to be 1, $R/2$, and 1, respectively. The popularity of the videos being requested at the BS follows a Zipf distribution with the skew parameter $\alpha = 0.8$, i.e., the probability that an incoming request is for the i-th most popular video is given as, $P_{if} = \frac{1/i^{\alpha}}{\sum_{f \in \mathcal{F}} 1/f^{\alpha}}$. To show the effects of MEC resources, we consider three scenarios, as follows:

- VQM refers to our algorithm, which considers the video cache and transcoding (i.e., cached video can be utilized only if the exact resolution level is requested) at a MEC server.
- MECC refers to the scenario where the MEC server only performs caching without transcoding.
- No-MECC refers to the scenario where all requested videos must be downloaded from the original server.

Impact of system parameters. In this subsection, we evaluate and compare the algorithm performance of different schemes under various simulation settings, in order to gain a further insight into the impact of different system parameters.

1) Number of users: To evaluate our proposed scheme, we increase the number of users to represent the increasing network load. Figure 4.3(c) illustrates the average distortion reduction with cache size of all users with different network settings. It can be observed that the average distortion value decreases when the number of users increase for three scenarios. The reason is that the BS and its MEC server allocate their transmission resources fairly to all connected users. when more users join the network and connect to the BS and MEC server, they will compete for the shared transmission resources, resulting in a higher probability of communication link interference and a lower average user throughput. However, VQM algorithm has better performance than that of MECC and No-MECC schemes. *2) Cache size*: Figure 4.4(a) illustrates the the average distortion reduction per user under varying the cache size of the MEC server S. The general observation for all schemes is that the average distortion reduction per user increases as the cache size gradually increases. The reason is that, the MEC server can pre-fetch more video files in its local cache with the increment of the cache size, which in turn can create more opportunities for the MEC server to serve more user requests without the need to communicate with the origin server. In comparison, the VQM has better performance than that of MECC and No-MEC schemes. *3) Computing capacity and percentage of HD video*: Figure 4.4(b) illustrates that increasing the capability of MEC servers can further improve the performance. Specifically, it is shown that the proposed VQM scheme becomes stable after a certain level of the capability of MEC servers, because other parameters, bandwidth of wireless links, and data rate of user have limited value. As pointed out in [129], the video data rate more than or equal 5 Mbps can be considered as a minimum requirement for the next generation mobile network. Percentage of HD video requests refers to the proportion of users that have achievable data rate higher than 5 Mbps (i.e., resolution higher than 1080p). Figure 4.4(c) shows that the ratio of HD video (1080p and higher resolutions) decreases with the load of the network, and that it increases with the available network resources. The proposed scheme gets a much higher ratio, up to around 14%, on the 1080p and higher resolutions compared to the MECC and No-MECC schemes. In addition, as shown in Fig. 4.4(c), when the network load is high and network resource is very limited, the proposed VQM scheme still can provide around 20% and 13% of users with resolutions of 1080p and higher.



(a) (b) (c)

Figure 4.4: (a) Average distortion reduction with different value of cache sizes; and (b) Average distortion reduction with different value of computing capacities; and (c) Percentage of HD video requests with different number of users.

4.6 Summary

We utilized the Distortion-Rate (DR) properties to model the end-user Quality of Experience (QoE) and formulated an optimization problem that aims at maximizing the average QoE over all users subject to practical constraints on video data rate and computing resources. Specifically, the considered VQM problem is cast as a MINLP that jointly determines the integer video resolution levels and data rate variables. Due to the challenging combinatorial and non-convex nature of the VQM problem, we utilized Lagrangian relaxation and Dual-Decomposition Method (DDM) to decompose the main optimization problem into two sub-problems. The sub-problems were formulated such that they can be solved efficiently using standard optimization solvers. Then, we presented the main MEC testbed implementation, discussed its challenges, and proposed a technique to stream video adaptively based on CQI index. Our experimental results showed that the frame processing time and CPU utilization of the MEC sever increase with MCS index and with downlink throughput. Additionally, simulation results were presented to demonstrate the superior performance of our VQM algorithm over competing schemes.

Chapter 5

Latency and Quality-Aware Task Offloading in Multi-Node Next Generation RANs

Next-Generation Radio Access Network (NG-RAN) is an emerging paradigm that provides flexible distribution of cloud computing and radio capabilities at the edge of the wireless Radio Access Points (RAPs). Computation at the edge bridges the gap for roaming end users, enabling access to rich services and applications. In this chapter, we propose a multi-edge node task offloading system, i.e., QLRan, a novel optimization solution for latency and quality tradeoff task allocation in NG-RANs. Considering constraints on service latency, quality loss, edge capacity, and task assignment, the problem of joint task offloading, latency, and Quality Loss of Result (QLR) is formulated in order to minimize the User Equipment (UEs) task offloading utility, which is measured by a weighted sum of reductions in task completion time and QLR cost. The QLRan optimization problem is proved as a Mixed Integer Nonlinear Program (MINLP) problem, which is a NP-hard problem. To efficiently solve the QLRan optimization problem, we utilize Linear Programming (LP)-based approach that can be later solved by using convex optimization techniques. Additionally, a programmable NG-RAN testbed is presented where the Central Unit (CU), Distributed Unit (DU), and UE are realized by USRP boards and fully container-based virtualization approaches. Specifically, we use OpenAirInterface (OAI) and Docker software platforms to deploy and perform the NG-RAN testbed for different functional split options. Then, we characterize the performance in terms of data input, memory usage, and average processing time with respect to QLR levels. Simulation results show that our algorithm performs significantly improves the network latency over different configurations.

5.1 Introduction

Motivation: Mobile platforms (e.g., smartphones, tablets, IoT mobile devices) are becoming the predominant medium of access to Internet services due to a tremendous increase in their computation and communication capabilities. However, enabling applications that require real-time, in-the-field data collection and mobile platform processing is still challenging due to (i) the insufficient computing capabilities and unavailable aggregated/global data on individual mobile devices and (ii) the prohibitive communication cost and response time involved in offloading data to remote computing resources such as cloud datacenters for centralized computation. In light of these limitations, the *edge computing* term was introduced to unite telco, IT, and cloud computing and provide cloud services directly from the network edge. In general, the edge cloud servers or nodes are usually deployed directly at the mobile Base Stations (BSs) of a Radio Access Network (RAN), or at the local wireless Access Points (APs) using a generic-computing platform. Hence, the edge cloud node has ability to execute the offloading applications in close proximity to end users. In this way, the network end-to-end (e2e) latency and the back/mid/fronth-haul cost will be reduced. Recently, Cloud Radio Access Network (C-RAN) [20] has been emerged as a clean-slate redesign of the mobile network architecture in which parts of physical-layer communication functionalities are decoupled from distributed, possibly heterogeneous, Radio Access Points (RAPs), i.e., BSs or WiFi hotspots, and are then consolidated into a baseband unit pool for centralized processing. However, the centralized C-RAN design follows a *"one size fits all"* architectural approach, which makes it difficult to address the wide range of Quality of Service (QoS) requirements and support different types of traffic [130]. Also, a fully centralized architecture imposes high capacity requirements on fronthaul links [19]. Therefore, Next Generation RANs (NG-RAN) [131] has been introduced as a resource-efficient solution to address the above issues and reduce deployment costs. *It is worthy of note that, due to the flexibility of NG-RAN architecture, mobile network operators will have high degree of freedom to move from a "full centralization" in C-RAN to a "partial centralization" in NG-RAN with a specific functional splitting option to a "distributed approach" in edge cloud [21]—enabling rich services and applications in close proximity to the end users.*

Task offloading can enhance the performance of mobile devices because servers in the edge cloud have higher computation capabilities than mobile devices. Therefore, enabling task offloading in NG-RAN is proposed to address the limitations (e.g., storage and computing resources) in the existing RANs. Meanwhile, in some cases, processing the entire input data in edge cloud servers would require more than the available computing resources to meet the desired latency/throughput guarantees. In the context of NG-RAN applications (e.g., IoT, AR/VR), transferring, managing, and analyzing large amounts of data in an edge cloud would be prohibitively expensive. Hence, the tradeoff between service latency and the tolerance of quality loss can improve key network performance metrics like the user's QoS [132,133]. In this chapter, we define the Quality Loss of Results (QLR) term as the level of relaxing/approximating in data processing while the user's QoS is still at an acceptable level. Accordingly, *our key idea is motivated by the observation that in several NG-RAN applications such as media processing, image processing, and data mining, a high-accuracy result is not always necessary and desirable; instead, obtaining a suboptimal result with low latency cost is more acceptable by vendors or end users.* Consequently, relaxing QLR in such applications alleviates the required computation workload and enables a significant reduction of latency and computing cost in NG-RAN.

Our Vision: Our objective is to design a holistic decision-maker for an optimal joint task offloading scheme with quality and latency awareness in a multi-edge NG-RAN to minimize the UEs' overall offloading cost. Specifically, we consider a multi-edge node network where each RAP is equipped with an edge node to provide computation offloading services to UEs. In this way, several key benefits could be brought up to NG-RAN system over the multi-node servers; (i) preventing the resource-limited edge node/servers from becoming the bottleneck. Usually, the cloud servers overload when serving a large number of UEs with high processing priority. By directing many UEs to nearby edge nodes, the overloaded can be alleviated; (ii) reducing the energy consumption and network latency. Each UE has the capability to offload its task to the RAP with a more favorable uplink channel condition; (iii) getting better network collaboration. The NG-RAN with multi-RAP set could coordinate with each other to manage and balance the computation resources between the

edge servers. *In this work, a Latency and Quality tradeoffs task offloading problem, QLRan, is formulated to trade off between the service latency and the acceptable level of QLR under specific application requirements (e.g., QoS, computing, and transmitting demands). Additionally, the process of task allocation across edge nodes is formulated as an objective optimization problem. The optimization objectives include both minimizing the average service latency and reducing the overall quality loss.*

Our Contributions: The main objective of this chapter is to design the QLRan algorithm, optimizing the trade-off between the application completion time and QLR cost. The main contributions of this chapter are summarized as follows.

- Subject to transmission and processing delays, quality loss, and computing capacity constraints, we formulate and analyze mathematically the QLRan optimization problem in NG-RAN as a Mixed Integer Nonlinear Program (MINLP) that jointly optimizes the computational task allocation and QLR levels. The problem formulation and analysis trade off optimizing the service latency and the overall quality loss.

- The QLRan optimization problem is proved as a non-deterministic polynomial-time hard (NP-hard) problem. To solve the problem efficiently, we first relax the binary computation offloading decision variable and QLR level to real numbers. Then, we utilize the Linear Programming (LP)-based method to solve the relaxed QLRan problem by using convex optimization techniques.

- We provide a set of tools to deploy the NG-RAN mobile network. To explore the virtualization in the 5G system, we assign several OpenAirInterface (OAI) [49] containers composing of a RAN and the core of the 5G system. Specifically, we implement a programmable testbed to demonstrate a connection between UE, RAN, and Evolved Packet Core (EPC) implemented in the NG-RAN virtualization environment. The real-time experiments are carried out under various configurations in order to profile functional splitting, the data input, memory usage, and average processing time with respect to QLR levels.

- We provide formal proofs on the convergence and optimality of our algorithm and

evaluate its performance under different network conditions. In terms of computing capacity and number of tasks, the numerical results show that latency can be reduced while decreasing the QLR level under practical physical constraints.

Chapter Organization. The remainder of this article is organized as follows. The related work is introduced in Sect. 5.2. We present the system model in Sect. 5.3. The QLRan problem is formulated in Sect. 5.4, followed by presenting a linear programming-based solution for QLRan optimization problem. The performance evaluation is discussed in Sect. 5.5; finally, we conclude the chapter in Sect. 5.6.

5.2 Related Work

In this section, we introduce the key concepts and papers from both industry and academia over the past several years.

5.2.1 Related Concepts and Technologies

Several cloud-based task offloading frameworks have been proposed in recent years. For example, Mobile Cloud Computing (MCC) has been proposed as a cloud-based network that can provide mobile devices with significant capabilities such as storage, computation, and task offloading to a centralized cloud [134]. However, MCC has faced several noticeable challenges to address the mobile next generation in terms of end-to-end network latency, coverage, and security. To tackle these challenges, Multi-access Edge Computing (MEC) has been introduced by European Telecommunications Standards Institute (ETSI) as an integration of the edge cloud computing systems and wireless mobile networks [135]. One of the key-value features of MEC is to enable rich services and applications in close proximity to end users. with the MEC paradigm, mobile devices have options to offload their computation-intensive tasks to a MEC server to meet the demanding Key Performance Indicators (KPIs) of 5G and beyond, especially in terms of low latency and energy efficiency. Similar to MEC systems, fog computing networks are proposed by CISCO systems to bring cloud services to the edge of an enterprise network [136]. In fog networks, the computation processing is mainly executed in the local area networks and in IoT gateways or fog nodes.

Recently, the concept of NG-RAN has been defined by 3GPP as a promising approach to merge edge cloud features and RAN functionaries. In industry, many RAN organizations have made significant progress in implementing open source-software that supports NG-RAN technology. For instance, EURECOM has implemented the OpenAirInterface (OAI) platform [49], which provides an open, full software implementation of 5G and beyond systems compliant with 3GPP standards under real-time algorithms and protocols. Plus, ORAN [137], founded by AT&T, aims to drive the mobile industry towards an ecosystem of innovative, multi-vendor, interoperable, and autonomous NG-RAN with reduced cost, improved performance, and greater agility. In general, these open RAN-software projects have a high degree of flexibility, such as being able to run CU and DU entities over a fully virtual environment such as VMs or Linux containers, as well as enabling promising next-generation features (e.g., network slicing and functional splitting). Such NG-RAN software will undoubtedly speed up the transition from monolithic and inflexible networks to agile, distributed elements depending on *virtualization*, *softwarization*, openness, and intelligence-fully interoperable RAN components.

5.2.2 Task Offloading in Cloud-based RANs

As part of task offloading in cloud-based RAN, several papers have focused on enhancing overall system performance in network energy, system latency, and energy efficiency. For instance, the work in [138] formulates a joint task offloading and resource allocation to maximize the users' task offloading gains in MEC. Then the main optimization problem has been decomposed into several sub-optimal problems that are solved using convex and quasi-convex optimization techniques. The authors in [139] study the energy-latency tradeoff problem for IoT partial task offloading in the MEC network by jointly optimizing the local computing frequency, task splitting, and transmit power. Then, the optimization is solved by an alternate convex search-based algorithm. In [140], by considering a cloud-fog computing network, the authors design a computation offloading algorithm to minimize total cost with respect to the energy consumption and offloading latency. To maximize the energy efficiency of task offloading, Vu *et al.* propose an approach based on the interior point method and bound algorithm. Exploiting machine learning methods in task offloading has

Table 5.1: Summary of Key Notations

Symbol	Description
$\mathcal{U}$	set of UEs
$\mathcal{S}$	set of edge nodes
$\mathcal{K}$	set of computational tasks
a_{uk}	indicator to show whether the task k is generated by UE u
a_{us}	indicator to show whether edge node s is available for UE u
a_{sk}	indicator to show whether task k is assigned to the edge node s
q_k	QLR level assigned to task k
$D_u(q_k)$	input data transfer the computing task k from UE u to the edge
$C_u(q_k)$	workload of computation to accomplish the task k
R_{us}	transmission data rate of the link between edge node s and UE u
τ_k^{up}	uplink transmission time
τ_k^{exe}	execution time of task k at the edge
f_{us}	assigned CPU-clock frequency on edge s of UE u
$B(q_k)$	computing demanded from task k with QLR level
δ^t	weight of latency consumption time for task k
δ^q	weight of QLR level for task k

also attracted several types of research in cloud-based RAN systems. Using reinforcement learning, the work in [141] introduces a MEC-based blockchain network where multi-mobile users act as miners to offload their data processing and mining tasks to a nearby MEC server via wireless channels. Although the focus of our article is in the line direction of mentioned works, applying different offloading schemes and constraints within the joint optimization NG-RAN framework could open up new, interdisciplinary avenues for researchers in the context of the 5G and beyond systems. Previously mentioned works consider a single remote server as the offloading destination. In contrast, with considering constraints on service latency, quality loss, and edge capacity, our work proposes an algorithmic approach for latency and quality tradeoff task offloading in multi-node NG-RANs. Furthermore, our work is based on real-world NG-RAN testbed experiments that allow us to characterize the performance in terms of data input, memory usage, and average processing time with respect to QLR levels.

5.3 System Model

In this section, we describe the task offloading process, network setting, quality loss of result tradeoff, and task uploading model. Table 5.1 summarizes the key notations used.

5.3.1 Task Allocation Process

The main process of the task allocation in our proposed NG-RAN system can be summarized as follows:

1. *Edge cloud nodes:* Initially, a UE searches its communication area for the best edge cloud node to connect to. Hence, the UE will send a pilot signal and collects response from edge cloud nodes. Any edge cloud node that responds will be considered to be a potential candidate. For instance, in Fig. 5.1, the edge cloud candidate of the UE is the edge node within the coverage area of LTE eNB DU.

2. *Task classification:* After the edge cloud node assignment, the UE starts uploading the task information to the edge node. Some key information include; i) the unique ID of the uploading task; ii) the application's layers and requirements; iii) the task profile, which include the task constraints (e.g., tolerable latency, QLR level, workload).

3. *Task executing:* After task classification, the RAP will run a resource allocation algorithm to determine: i) the service time required for task accomplishment; ii) computing capacity that is available for the task executing; iii) the compare these estimates to the tasks' tolerated latency requirement.

5.3.2 Network Description

For the NG-RAN system model, we consider a multi-cell, multi-node edge system as illustrated in Fig. 5.1, in which each RAP (e.g., BS, eNodeB (eNB), gNodeB (gNB), etc.) engages with a set $\mathcal{S} = \{1, 2, .., S\}$ of S edge nodes (e.g., neighboring DU servers) to supply computation offloading services to the limited-resource mobile devices such as smartphones, tablets, and IoT devices. Specifically, each edge cloud node can be realized either by a physical server, or by Virtual Machine (VM)/container, which can communicate with the UE

through wireless channels provided by the corresponding RAP. Plus, each UE can select to offload its computation task to an edge node from the candidate nearby severs. Accordingly, we denote the set of UEs in the mobile system and the set of computation tasks as $\mathcal{U} = \{1, 2, ..., U\}$ and $\mathcal{K} = \{1, 2, ..., K\}$, respectively. To define the association between UEs and RAPs, we define two binary indicators as follows, $a_{uk} \in \{0, 1\}$ is presented to indicate whether the task k is generated by UE u, while $b_{us} \in \{0, 1\}$ is presented to indicate whether edge node s is available for UE u (i.e., the edge node s has an acceptable channel state condition to be in the list of edge candidate). Hence,

$$a_{uk} = \begin{cases} 1, & k \in \mathcal{K}_u \\ 0, & \text{Otherwise} \end{cases}, \; a_{us} = \begin{cases} 1, & s \in \mathcal{S}_u \\ 0, & \text{otherwise} \end{cases}, \tag{5.1}$$

where $\mathcal{S}_u \subseteq \mathcal{S}$ is denoted as the set of edge candidates for UE u, and $\mathcal{K}_u \subseteq \mathcal{K}$ is defined as the set of tasks generated by UE u. Thus, from (5.1), we can denote a_{sk} as a binary variable to indicate whether task k is assigned to edge node s or not. If the edge node s is available for UE u, the task k will be successfully assigned to the edge node s. Hence, a_{sk} will be satisfy the following requirement,

$$a_{sk} \leq \min\{a_{uk}, a_{us}\}, \forall u \in \mathcal{U}, k \in \mathcal{K}, s \in \mathcal{S}. \tag{5.2}$$

The modeling of user computation tasks, task uploading transmissions, edge computation resources, and offloading utility are presented here below.

5.3.3 Quality Loss of Result Tradeoff

Many emerging applications in cloud-based computing networks (e.g., online recommender, video streaming, object recognition, and image processing) exhibit variant optional parameters that authorize end-users to take advantage of the tradeoff between QLR and service latency. For instance, many object recognition algorithms basically demand specific extraction methods of several numbers of layers with given wavelengths and orientations from image datasets for advanced analysis [142]. Hence, the achieved QLR managing the processing time in the object recognition can be relaxed if the number of extracted layers are

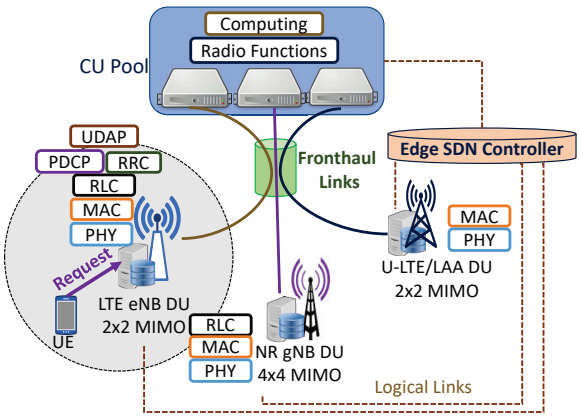


Figure 5.1: System overview of QLRan, in which the gray circle represents the communication range of the RAP.

properly adapted. Another example is multi-bitrate video streaming, in which the Over-The-Top (OTT) video content providers (e.g., YouTube, Amazon Prime, Netflix, ...) offer to end-users different video quality levels to fit within the device's display and network connection [21,143]. Adjusting the video quality levels can save extra computational energy and time for OTT video providers at the same time make users experience good video watching without interruption. In this chapter, we denote q_k as QLR level assigned to task k. Hence, we allow each UE u to select different q_k values to exploit the trade-off between processing cost and latency. We define QLR as five levels in which level 1 refers to the strictest demand for quality, while level 5 represents the highest tolerance for quality loss. In practice, QLR levels are determined at an application-specific level.

5.3.4 Task Uploading

The computation task uploading in NG-RAN system can be described as a tuple of two parameters, $\langle D_u(q_k), C_u(q_k)\rangle$, where $Du(q_k)$ [bits] represents the amount of input data required to transfer the application execution (including system settings, application codes, and input parameters) from the local device to the edge node, and $C_u(q_k)$ [cycles] denotes the workload, i.e., the amount of computation to accomplish the task. Each UE $u \in \mathcal{U}$ has one computation task at a time that is atomic and cannot be divided into subtasks. The values of $D_u(q_k)$ and $C_u(q_k)$ can be obtained through carefully profiling of the task execution [138].

In Sect. 5.5, we will provide more details about the modeling of these metrics. Besides, the computation task associated with UE can be executed locally or offloaded to an edge cloud node. The Mobile device would save battery bower by offloading part of its task application to the remote edge; however, a considered cost, time and energy, from uploading the input data would be added in the task offloading scenario. Therefore, similar to [138], we consider several time parameters in the case of the UE u offloads its task k to one of the edge nodes, the overall uploading time delay consists of the follows: (i) the time τ_k^{up} [s] to transmit the input to the edge node on the uplink, (ii) the time τ_k^{exe} to perform the computation task at the edge node, and (iii) the time to bring the output data from the edge node back to UE on the downlink. In general, the size of the output data is much smaller than the input data, and the downlink data rate is much higher than that of the uplink. Therefore, similar to [138,144,145], we neglect the delay of sending the output in our computation model. Note that when the delay of the downlink transmission of output data is non-negligible, our proposed approach can still be directly applied for a given downlink rate allocation scheme and known output data size. The transmission time of UE u, that is required to send its task data input $D(q_k)$ in the uplink, can be determined as,

$$\tau_k^{up} = \frac{D_u(q_k)}{R_{us}}, \forall u \in \mathcal{U}, k \in \mathcal{K}, s \in \mathcal{S}, \tag{5.3}$$

where R_{us} is the transmission data rate of the link between the selected edge node s and UE u. Given the computing resource assignment, the execution time of task k at edge node s is,

$$\tau_k^{exe} = \frac{C_u(q_k)}{f_{us}}, \forall u \in \mathcal{U}, k \in \mathcal{K}, s \in \mathcal{S}, \tag{5.4}$$

where f_{us} denotes the assigned CPU-clock frequency on edge s to UE u of task k.

5.3.5 System Constraints

We now introduce the following four constraints to capture the features of a task offloading multi-node NG-RAN system.

1. *QLR constraint:* As we will describe in 5.3.2, q_k can be modeled based on a specific

key metric in an application. In our scenario, we adopt the video resolution level as q_K in the video streaming application. Under these considerations, which will be described in more details in Sect. 5.5 the QLR constraint for the task k is defined as, $q_k = \{1, 2, 3, 4, 5\}, \forall k \in \mathcal{K}$

2. *Task association constraint:* We assume each computation task of the UE must be assigned to one edge cloud node. Hence, the offloading policy would satisfy the task association constraint, expressed as,

$$\sum\nolimits_{s \in \mathcal{S}} a_{sk} = 1, \forall k \in \mathcal{K}. \tag{5.5}$$

3. *Service latency constraint:* In many graphic applications with multiple tasks, the reduction of computation workload at the edge node considerably affects the task execution latency. For instance, real-time gaming applications have a preferred response time around 50 ms latency to enjoy a higher Quality of Experience (QoE) [146]. Achieving appropriate latency for a graphic video application demands tradeoffs processing time, uploading time, and quality. In this chapter, we denote parameter $\tau_k^{\max}$ to define the maximum tolerable system latency for the task k. To guarantee that the task is accomplished in the allowed threshold time, the service latency constraint is expressed as,

$$\tau_k^{up} + \tau_k^{exe} \leq \tau_k^{\max}, \forall k \in \mathcal{K}. \tag{5.6}$$

4. *Resource constraint:* In multi-node NG-RAN with intensive workloads, the computation capacity should be taken into account while designing a latency-quality optimization algorithm. The computation capacity could refer to several hardware metrics such as GPU, CPU, and memory. Adjusting these parameters is directly affected by the service latency and the required quality. However, the computational processing capacity at the edge cloud node cannot exceed its limited capacity. Therefore, we present the parameter, $B_s^{\max}$, as the maximum computation capacity of edge node s, while $B(q_k)$ is defined as the require computation capacity generated from processing

task k at QLR q_k. Hence, the capacity constraint is model as,

$$\sum\nolimits_{k\in\mathcal{K}} B(q_k)a_{sk} \leq B_s^{\max}, \forall s \in \mathcal{S}. \tag{5.7}$$

5.4 Problem Formation

In this section, we mathematically formulate the QLRan optimization problem, which optimizes the trade-off between the service latency and quality loss while offloading tasks in NG-RAN edge nodes. Due to the intractability of the problem and the need for a practical solution, we then present a step-by-step solution based on a linear programming-based solution, which is employed to transform the QLRan problem into a convex optimization problem.

5.4.1 Latency and Quality Tradeoffs Problem

For a given $\mathcal{A} = \{a_{sk}|s \in \mathcal{S}, k \in \mathcal{K}\}$,the the set of selected edge nodes, and $\mathcal{Q} = \{qk|k \in \mathcal{K}\}$, the set of selected QLR levels, we define the system utility as the weighted-sum of all the UEs' offloading utilities,

$$J_k(\mathcal{A}, \mathcal{Q}) = \delta^t \tau_k + \delta^q q_k \sum\nolimits_{s\in\mathcal{S}} a_{sk}, \forall s \in \mathcal{S}, k \in \mathcal{K}, \tag{5.8}$$

where $\tau_k = \left(\tau_k^{up} + \tau_k^{exe}\right)$, $0 \leq \delta^t \leq 1$ and $0 \leq \delta^q \leq 1$ denote the weights of latency consumption time and QLR levels for task k, respectively. Note that we define the latency and quality tradeoffs utility, $J_k(\mathcal{A}, \mathcal{Q})$ of task k as a linear combination of the two metrics because both of them can concurrently reflect the latency-quality tradeoff of executing a task, i.e., both higher longer computation completion time and high accuracy of result lead to higher computational cost. To meet task-specific demands, we allow different UEs to select different weights, which are denoted by δ^t and δ^q, in decision making. For example, a UE with low accuracy application demand would like to choose a larger δ^q to save more computational cost. On the other hand, when a UE is running some delay-sensitive applications (e.g., online movies), it may prefer to set a larger δ^t to reduce the latency. We now formulate the Latency and Quality Tradeoffs (QLRan) problem as a system utility

minimization problem, i.e.,

$$\mathcal{P}1 : \min_{\mathcal{A},\mathcal{Q}} \sum\nolimits_{k\in\mathcal{K}} J_k(\mathcal{A},\mathcal{Q}) \tag{5.9a}$$

s.t. :

$$a_{sk} \in \{0,1\}, q \in \{1,2,3,4,5\}, \forall s \in \mathcal{S}, k \in \mathcal{K}, \tag{5.9b}$$

$$\sum\nolimits_{s\in\mathcal{S}} (\tau_k^{up} + \tau_k^{exe}) a_{sk} \leq \tau_k^{\max}, \forall k \in \mathcal{K}, \tag{5.9c}$$

$$\sum\nolimits_{k\in\mathcal{K}} B(q_k) a_{sk} \leq B_s^{\max}, \forall s \in \mathcal{S}, \tag{5.9d}$$

$$\sum\nolimits_{s\in\mathcal{S}} a_{sk} = 1, \forall k \in \mathcal{K}. \tag{5.9e}$$

The constraints in the formulation above can be explained as follows: constraint (5.9b) secure that the computation task can be accomplished in the time that cannot exceed than the demanded maximum threshold time,$\tau_k^{\max}$; constraint (5.9c) implies that the demand for computation capacity must not exceed its edge node capacity; finally, constraint (5.9d) indicates that each task must be assigned as a whole to one edge node.

Proposition 2. $\mathcal{P}1$ *is an NP-hard problem.*

Proof. To demonstrate that $\mathcal{P}1$ is an NP-hard, let first consider the case, where $\delta^t = 0$, $\delta^q = 1$. That means the time spent for uploading and executing a task is neglected for this case and the focusing is done only on the second part of $J_k(\mathcal{A},\mathcal{Q})$, where the QLR term is important. We assume that $\hat{q}_k$ represents the opposite value of q_k and denotes as the quality level in the result of task k. Accordingly, we can reformulate the $\mathcal{P}1$ as $\hat{\mathcal{P}}1$, in which the the new objective function $\hat{J}(\mathcal{A},\mathcal{Q})$ will be maximized. Plus, constraint (5.9c) can be omitted for simplicity. Besides, Constraint (5.9d) is rewritten to imply that the resource requirement of task k is exactly equal to its quality value $\hat{q}_k$. Each edge cloud node in the NG-RAN system can only handle one task generated from the UE in the RAP coverage area. Let $\hat{a}_k$ is defended as a binary indicator to show whether the task k is assigned to the edge node, and B to denote the resource capacity of the edge node. With these considerations,

the optimization problem in (5.9) can be relaxed as,

$$\widehat{\mathcal{P}}1 : \max \ \sum\nolimits_{s\in\mathcal{S}} \hat{q}_k \hat{a}_k \tag{5.10a}$$

$$\text{s.t.} : \ \sum\nolimits_{k\in\mathcal{K}} \hat{q}_k \hat{a}_k \leq B, \tag{5.10b}$$

$$\hat{a}_k \in \{0, 1\}. \tag{5.10c}$$

It is obvious that problem $\widehat{\mathcal{P}}1$ is a standard weighted-sum problem that is an NP-complete problem [147]. Therefore, $\mathcal{P}1$ also can be characterized as an NP-hard problem. The proof is completed. □

Next, we will propose an iterative approach to solve $\mathcal{P}1$ based on Linear Programming-based (LP) optimization. By utilizing the standard optimization solver (e.g., MOSEK [67]), the proposed system can generate an efficient task allocation decision with an acceptable latency tolerance constraint.

5.4.2 Linear Programming-based Solution

The key challenge in solving the optimization problem in $\mathcal{P}1$ is that the integer constraints $a_{sk} \in \{0, 1\}$ and $q \in [1, 5]$ make $\mathcal{P}1$ a MIP problem, which is in general non-convex and NP complete [148]. Thus, similar to works in [133, 149], we first relax the binary computation offloading decision variable, a_{sk}, and QLR level, q_k, to real numbers, i.e., $0 \leq a_{sk} \leq 1$. Then we will discuss the convexity of $\mathcal{P}1$ with the relaxed optimization variables a_{sk} and q_k. Then, we consider the following; $D(q_k) = y_d q_k + z_d$, $C(q_k) = y_t q_k + z_t$, $B(q_k) = y_b q_k + z_b$, and $x_{sk} = q_k a_{sk}$. The parameters y_d, z_d, y_t, z_t, y_b, and z_b can be estimated by offline profiling of the NG-RAN testbed, as detailed in Sect. 5.5. The LP problem for the primal problem

is given by,

$$\mathcal{P}2: \min_{\mathcal{A},\mathcal{Q},\mathcal{X},t} \delta^t t + \delta^q \sum\nolimits_{s\in\mathcal{S}} x_{sk} \tag{5.11a}$$

s.t. :

$$0 \leq a_{sk} \leq 1, 1 \leq q_k \leq 5, t \leq \tau_k^{max}, \forall s \in \mathcal{S}, k \in \mathcal{K}, \tag{5.11b}$$

$$0 \leq x_{sk} \leq 5a_{sk}, \forall s \in \mathcal{S}, k \in \mathcal{K}, \tag{5.11c}$$

$$q_k - 5(1 - a_{sk}) \leq x_{sk} \leq q_k \forall s \in \mathcal{S}, k \in \mathcal{K}, \tag{5.11d}$$

$$\sum\nolimits_{s\in\mathcal{S}} \left(\frac{y_d}{R_{us}} + \frac{y_t}{f_{us}}\right) x_{sk} + \left(\frac{z_d}{R_{us}} + \frac{z_t}{f_{us}}\right) a_{sk} \leq \tau_k^{\max}, \tag{5.11e}$$

$$\sum\nolimits_{s\in\mathcal{S}} a_{sk} = 1, \forall k \in \mathcal{K}. \tag{5.11f}$$

Proposition 3. *Constraints* (5.11c) *and* (5.11d) *can be relaxed to the constraint* $x_{sk} = a_{sk}q_k$.

Proof. **Case 1:** ($a_{sk} = 0$**, and** $q_k \in [1, 5]$). Form constraints (5.11c) and (5.11d), we can conclude the follows,

$$x_{sk} \leq 0, x_{sk} \geq 0, \text{and } x_{sk} \leq q_k, x_{sk} \geq q_k - 5, \tag{5.12}$$

After solving (5.12), we can get $x_{sk} = 0$.

Case 2: ($a_{sk} = 1$**, and** $q_k \in [1, 5]$).

$$x_{sk} \leq 5, x_{sk} \geq q_k, \text{and } x_{sk} \geq q_k, x_{sk} \geq 0, \tag{5.13}$$

From (5.13), we can conclude $x_{sk}q_k = q_k$. From **Case 1** and **Case 2**, we demonstrate that the constraints (5.11c) and (5.11d) are equivalent to the constraint $x_{sk} = a_{sk}q_k$. The proof is complete. □

5.5 Performance Evaluation

In this section, we describe the testbed experiments and simulation results to provide more details about the QLR level model in terms of memory and CPU usage, as well as to test

the effectiveness of the QLRan algorithm.

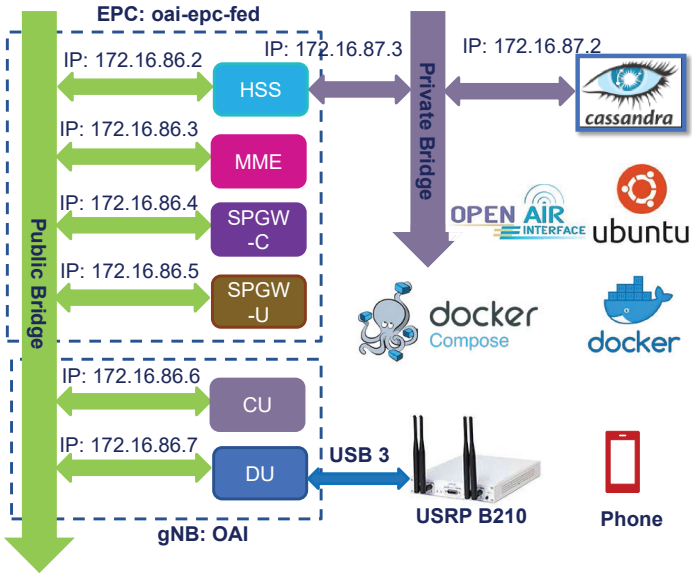


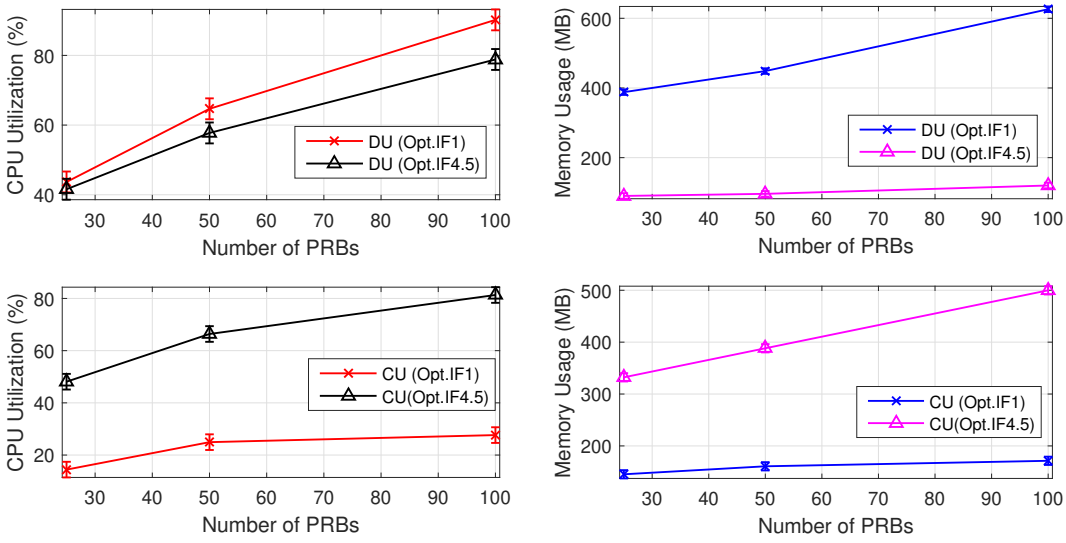
Figure 5.2: Logical illustration of the fully containerized-based NG-RAN testbed.

5.5.1 Testbed Experiment

We present here our QLRan testbed, including the architecture, configuration, and experiment methods. Then, we analyze the performance of QLRan in terms of CPU processing time and latency.

Architecture. We conducted experiments on a testbed consisting of various components, i.e.,

- *End users:* For our experiment we use a Samsung Galaxy S9 running on Android 10 that acts as the UE.
- *Edge nodes:* To simulate the edge node, we use Asus Laptop equipped with an Intel Pentium III processor running Ubuntu 18.04. The cloud is represented by the more powerful desktop PC Intel Xeon E5-1650, 12-core at 3.5 GHz and 32 GB RAM.
- *Network:* The structure of OAI consists of two components: one, called *oai*, is used for building and running gNB units; the other, called *openair-cn*, is responsible for building and running the Evolved Packet Core (EPC) networks, as shown in Fig. 5.2. The Openair-cn component provides a programmable environment to implement and



(a) (b)

Figure 5.3: (a) CPU utilization vs. number of PRBs for DU and CU in Options IF1 and IF4.5; (b) Memory usage vs. number of PRBs for DU and CU in Options IF1 and IF4.5.

manage the following network elements: Mobility Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (SPGW-C), and PDN Gateway (PGW-U). We use WiFi as well as LTE to act as our physical link between the UE and the edge. The edge is connected to the cloud through Ethernet. As illustrated in Fig. 5.2, all the EPC and gNB components are implemented by as container image by using Docker and docker compose [150]. The UE and RF RAN are implemented in hardware, conventional cell phone and USRP 210, respectively.

NG-RAN Testbed for Different Functional Options. We endowed our testbed with several functional split options so as to realize the CU and DU in gNB. All containers in Fig. 5.2 are hosted by the desktop PC Intel Xeon E5-1650, 12-core at 3.5 GHz and 32 GB RAM. For the UE, we use a Samsung Galaxy S9 running on Android 10. For network configuration, we run our NG-RAN prototype for three functional splits; Option F1, (PDCP/RLC, Option 2 in 3GPP TR 38.801 standard), Option IF4.5 (Lower PHY/Higher PHY, a.k.a Option 7.x in 3GPP TR 38.801 standard), and Option LTE eNB. We summarize the testbed configuration parameters in Table 5.2.

Figure 5.3(a) shows the CPU utilization percentage at DU and CU containers. The CPU utilization percentage is measured by the *docker stats command* in Ubuntu, which

provides a live data stream for running containers. The downlink UDP traffic repeatedly is sent from the SPGW-U container to the UE with various PRB settings in two functional split Options, F1 and IF4.5. It can be observed that the CPU consumption for DU and CU is continuing to increase linearly as the number of PRBs is increased in the two functional split options. However, Option IF1 consumes more CPU percentage at DU than at the CU. For example, the CPU utilization percentage is 43.67 % in DU while it is 14.42 % in CU. That is because the higher PHY operations such as RLC/MAC, L1/high, tx precode, rxcombine, and L1/low operations reside in DU for split Option IF1 [151]. In Option IF4.5, the pattern is reversed. We can see from Fig. 5.3(a) that the CPU usage at CU is higher than at DU. Figure 5.3(c) shows the memory usage of DU and CU containers when the NG-RAN testbed performs in Options IF1 and IF4.5 at different values of PRBs. Similar to the CPU consumption pattern, the memory usage at DU is higher than at DU in Option IF1. For example, the memory usage is 388 MB in DU while it is 145.3 MB in CU at Option IF1 and 25 PRB.

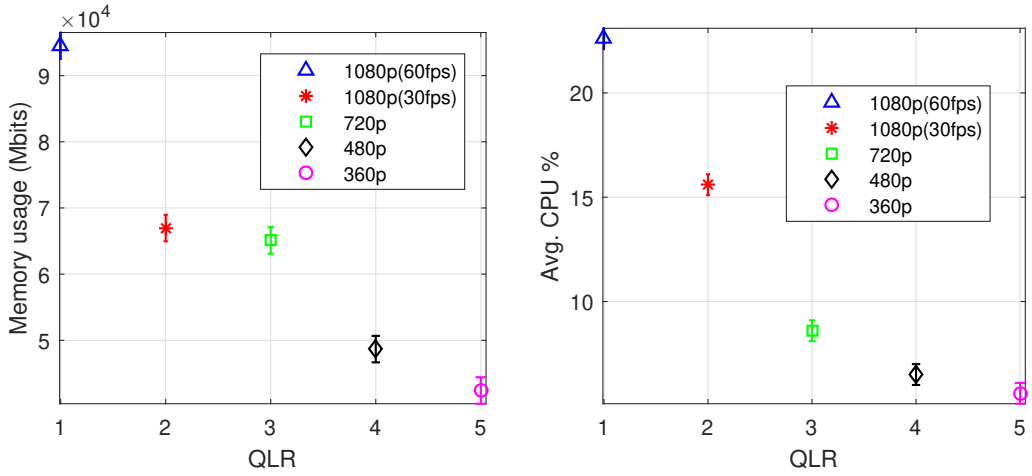
Table 5.2: Testbed Configuration Parameters for gNB.

Mode	FDD	**Options**	IF1, IF4.5, eNB
Frequency	2.68 GHz	**PRB**	$25, 50, 100$
TX Power	150 dBm	**Env.**	Multi-container
MCS	28	**SINR**	$15 - 20$

5.5.2 Application Profiling

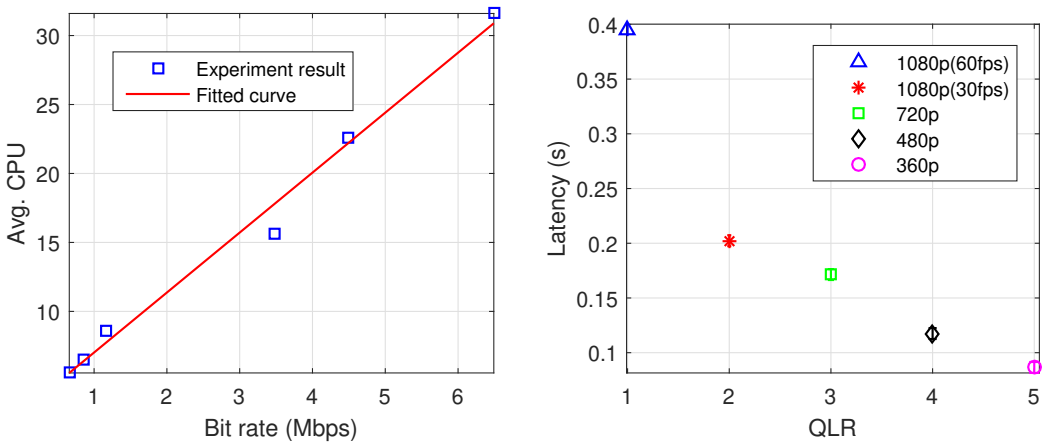
To test QLRan, we consider two applications: video streaming and facial detection in smart surveillance cameras. These two tasks are both video-based tasks that require varying degrees of quality.

Video streaming application: Video streaming is run on two Dell Workstations, each with two Xeon E6-1650 processors. Each workstation is equipped with 32GB of RAM running Ubuntu 18.04. In our experiments, a prerendered movie of one minute is streamed between these two computers using *ffmpeg*, a video transcribing and streaming application. On the other end, a *ffplay* is used to receive and render the video stream. Four different



(a) (b)

Figure 5.4: (a) Memory usage for various QLR levels in video streaming; (b) CPU usage for various QLR levels in video streaming.



(a) (b)

Figure 5.5: (a) Relation between a video's bitrate and CPU consumption in video streaming; and (b) Latency in facial recognition.

video resolutions are used: 360×240, 480×360, 960×720, and 1920×1080. Additionally for the highest resolution of 1920×1080, 30 fps as well as 60 fps is used as well as a stereographic stream for 60 fps for potential 3D reconstruction applications.

Facial recognition application: In addition to network streaming, a basic facial detection and recognition application is tested against the very same resolutions in the network stream. The facial recognition algorithm is based on the popular and simple *dLib library* available for python [152].

For both applications, we have chosen QLR 1 to represent the best networking conditions

while a QLR of 5 represents the worst network conditions. Using the `top` utility, we were able to log in 1 second intervals the CPU consumption as well as the memory consumption of the process on the server streaming the video. Note that in both Figs. 5.4(a) and 5.4(b) we witness a linear increase in both memory and CPU consumption which can be expressed in the following equations,

$$B(q_k) = -10.4q_k + 95.9,\ C(q_k) = -5.2q_k + 33.3, \forall k \in \mathcal{K}. \tag{5.14}$$

In Fig. 5.5(a), since we downsampled the video resolutions ourselves, we are able to extract the exact average bitrate for various stream profiles to arrive at an equation,

$$D(q_k) = 4.30x + 2.75, \forall k \in \mathcal{K}, \tag{5.15}$$

where x represents the achievable bit rate in Mbps. Similarly—as shown in Fig. 5.5(b)—as video resolutions increase in facial recognition application, so does processing time. Hence, the QLR processing time can be modeled as,

$$T^{proc} = -0.08q_k + 0.51, \forall k \in \mathcal{K}. \tag{5.16}$$

5.5.3 Numerical Result

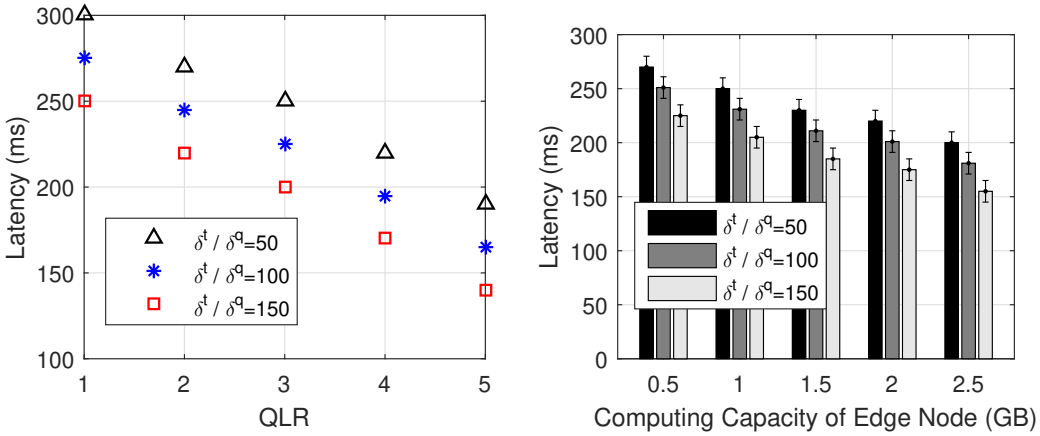
We consider a NG-RAN system consisting of 100 m $\times$ 100 m cell with a RAP in the center. The mobile devices, $N = 25$, are randomly located inside the cell. The channel gains are generated using a distance-dependent path-loss model given as $L[\text{dB}] = 140.7 + 36.7 \log_{10} \text{d}_{[\text{km}]}$, where d is the distance between the mobile device and the BS, and the log-normal shadowing variance is set to 8 dB. The other network parameter values are listed in Table 5.3.

In general, the computational tasks can be classified into two different categories: (i) approximatable, tasks that can be approximated to achieve significant savings in execution time, with however a potential loss of accuracy in the result; and (ii) non-approximatable,

Table 5.3: Configuration Parameters for Simulation.

y_d,z_d	4.3, 2.75	Capacity [GB]	1.5
y_t, z_t	-5.24, 3.31	δ^t/δ^q	$[50, 100, 150]$
y_b,z_b	-10.41, 95.9	Data Rate [Mbps]	2
U, K, S	$10, 10, 20$	Delay Tolerance[ms]	300
$B_s^{\max}$ [GB]	3	QLR	$[1, 2, 3, 4, 5]$

tasks whose execution without any approximation is necessary for the success of the application, i.e., if any approximation technique were applied on these tasks, the application would not generate meaningful results. We refer the interested readers to the work in [153], which introduces a lightweight online algorithm that selects between these tasks to enable real-time distributed applications on resource-limited devices. Accordingly, we consider video streaming and facial recognition applications, which can be considered as approximatable tasks, for profiling. The reason for choosing these task applications is that they can highly benefit from the collaboration between mobile devices and edge platforms. In experimental results, we study the impact of the difference of service quality level, which can be considered as the resolution level of video streaming and facial applications, on the system latency and edge node computing capacity.



(a) (b)

Figure 5.6: System latency performance versus: (a) QLR levels; and (b) Computing capacities.

Impact of Control Parameters δ^t and δ^q. We discussed the definitions of the scalar weights δ^t and δ^q in Sect. 5.4. In general, these parameters are used to make a tradeoff between the service latency and quality. Specifically, when δ^t/δ^q is increased, the QLRan

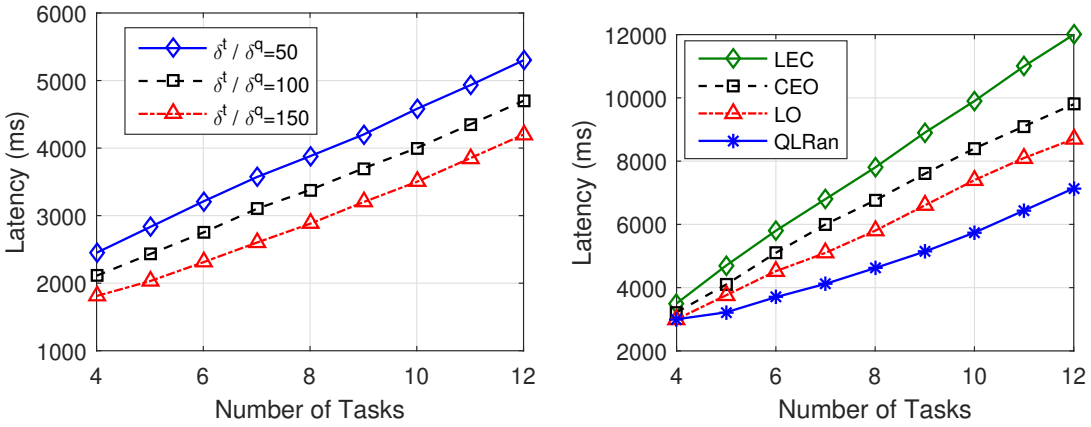


Figure 5.7: (a) System latency performance versus number of computational tasks; and (b) System latency versus number of computation tasks under different execution schemes.

algorithm will be more sensitive to system latency; otherwise, it will be the quality of result sensitive. Fig. 5.6(a) shows that the latency cost decreases with a larger QLR parameter for different values of δ^t/δ^q, which are 50, 100, and 150. Specifically, the average system latency value at the δ^t/δ^q ratio is 50 and the QLR level is 1 is around 300 ms, while the average system latency values are 275 ms and 250 ms for QLR level is 1 and δ^t/δ^q ratio are 100 and 150, respectively. That is because when QLR level is 1, which refers to the best accuracy that can be obtained from processing the computational task in the edge cloud node, the computational complexity at QLRan will increase as well as the system latency. The system latency decreases with the tolerance of quality becoming low. Plus, QLRan shows good performance when the algorithm is acting towards the latency performance. For instance, when the QLR level increases to 4, the average system latency of QLRan turns down to 220 ms, 200 ms, and 180 ms, for $\delta^t/\delta^q = 50$, 100, and 150, respectively.

Impact of Computing Capacity of Edge Cloud Node. To evaluate the offloading performance in term of memory usage, $B(q_k)$, We run the QLRan algorithm for different values of computing capacity $B(q_k)$ at δ^t/δ^q ration of 50, 100, and 150. We observe that as long as the memory requirements are sufficient, the computing capacity (CPU/GPU) requirements can be satisfied. Hence, the performance can be evaluated with several memory sizes. As mentioned in Table 5.3, We set the memory size of a edge node node to $B_s^{\max} = 1.5$ GB by default, while the ratio of $\delta^t/\delta^q = 50$, 100, and 150 are tuned to measure the system latency and QLR with several memory capacity values. Also, the memory size

of each edge node is tuned from 0.5 to 2.5 GB. As illustrated in Fig. 5.6(b), the system latency decreases when the memory capacity of the QLRan algorithm increases. Specifically, the service latency decreases by around 12% from tuning the δ^t/δ^q ratio from 50 to 100 at computing capacity is 0.5 GB, while the overall pattern continues to decrease as the computing capacity value is increased.

Impact of Increasing Number of Tasks. For the computation task, we use the face detection and recognition application for airport security and surveillance [154], which can highly benefit from the collaboration between mobile devices and edge platforms. The setting value of 12 computational tasks are selected to be in the range of 90 and 250 KB for the data size and between 890 and 1150Megacycles for the CPU cycles. Fig. 5.7(a) shows the performance of different schemes versus the number of tasks. In this figure, the parameter of task data input is a random variable following linearity increasing with QLR levels. It can be seen that the case $\delta^t/\delta^q = 150$ has less latency cost compared to the other.

5.5.4 Comparing QLRan with Other Baseline approaches

We compare the QRan algorithm with the following existing benchmarks:

- Cloud Edge Executing Only (CEO): Each UE $u \in \mathcal{U}$ has only one option: to offload its task to cloud edge node within its communication coverage without considering the tradeoff between latency and approximate computing;
- Local Executing Only (LEO): Each UE $u \in \mathcal{U}$ has only one option: to execute its task locally within its communication coverage without considering the tradeoff between latency and approximate computing;
- Latency-aware ask Offloading (LO): Each UE $u \in \mathcal{U}$ can offload its task to edge cloud within its communication coverage. Here, only the latency is considered in the objective function, while approximate computing is ignored.

As illustrated in Fig. 5.7(b), we evaluate the running performance of 12 computational tasks under different offloading schemes. Our joint latency and quality-aware offloading scheme outperforms other schemes. Specifically, the performance gap between QLRan and other

schemes increases when the number of task increases. That is because the QLRan algorithm is designed to trade off between the latency and QLR level, while the other schemes only focus on the offloading and executing scenarios.

5.6 Summery

We presented latency-quality tradeoffs and task offloading in multi-node next-generation RANs. We designed our algorithm, QLRan, to reduce system service latency while adjusting the overall quality level. Practical NG-RAN system constraints have been considered to formulate the proposed task offloading problem. The constraints depend on network latency, quality loss, and edge node computing capacity, while the objective function is the weighted sum of all the UEs' offloading utilities. The QLRan is cast as an NP-hard problem; therefore, we propose a Linear Programming (LP)-based approach that can be solved via convex optimization techniques. Simulation results are generated from running several real-time applications on the NG-RAN testbed, which is completely implemented under container-based virtualization and functional-split option technologies. We considered video-streaming and facial-recognition applications as building blocks of many cloud-based applications. We evaluated our solution and thorough simulation results showed that the performance of the QLRan algorithm significantly improves the network latency over different configurations.

Chapter 6

Deep Reinforcement Learning-based Resource Allocation for Next Generation Radio Access Networks

Next-Generation Radio Access Network (NG-RAN) will leverage a novel architecture that accelerates the transition from inflexible and monolithic networks to agile and disaggregated components. In this chapter, we introduce a novel Deep Reinforcement Learning Based Resource Allocation (ReLAX) framework to deal with the joint optimization of UE association and power allocation in NG-RAN systems. Considering the dynamic nature of the NG-RAN environment, ReLAx problem has been formulated to maximize the network Energy Efficiency (EE) under the constraints of Quality of Service (QoS), fronthaul link, functional split configuration and transmit power budget. The optimization problem is cast via a Mixed-Integer Non-Linear Programming (MINLP), which is in general non-convex and NP-complete. A multi-task Deep Deterministic Policy Gradient (DDPG) method is proposed to solve the NG-RAN resource allocation optimization problem, in which two actors are trained to generate UE association and power allocation, respectively. However, using two separate models for two closely-related variables could be a waste of training time and resources. As such, we introduce the soft multi-task learning as a constraint during training so that one model would not drift too far away from the other one. Our real-time experiments on a fully containerized NG-RAN testbed show the effect of functional splits on CPU utilization and system latency. Besides, simulation results show that the proposed resource allocation solution outperforms competing traditional algorithms, such as ordinary DDPG and Weighted Minimum Mean Square Error (WMMSE).

6.1 Introduction

Motivation. In the near future, it is expected that mobile data traffic continues to surge due to the proliferation of smart portable devices and enormous demands for emerging technologies, such as IoT, video streaming, and Augmented/Virtual Reality (AR/VR). According to a recent report from Cisco, there will be 5.3 billion total Internet users, while the average 5G connection speed will reach 575Mbps by 2023 [155]. The increase in the traffic pattern for Beyond 5G (B5G) services imposes significant challenges in meeting the specific requirements, including Quality of Service (QoS), channel condition, and service latency, from the existing mobile network architectures. Radio and computation resources can be considered as a real bottleneck in satisfying the increasing trend in B5G's demands. On the other hand, adding more radio and computing resources at the network sites could bring another critical issue in the energy consumption of the future mobile communication systems, especially the direct impact on increasing the Operating Cost (OPEX) of network operators. Considering the limited communication radio resources and the prohibitive signaling energy costs, it is an essential need for studying novel practical RAN systems, in which the resource allocation algorithms can be efficiently applied.

Recently, Next Generation Radio Access Networks (NG-RAN) has been presented as an emerging framework to enable the *virtualization* and *softwarization* technologies [156]. The key feature of NG-RAN design is to flexibly move the main signal processing functions performed by the digital baseband (PHY/MAC) processing to the Central Unite (CU) while maintaining the radio access and low levels of communication functionalities at the cell sites in the Distributed Unites (DUs). Cooperation between two main units in an efficient way will open a path to enhance the overall network significant metrics, including architecture planning, network operation, resource utilization, and back/mid/front-haul management. Consequently, multiple wireless B5G services, such as massive Machine-Type Communication (mMTC), enhanced Mobile Broadband (eMBB), and ultra-Reliable Low-Latency communication (uRLLC), can dynamically deploy and manage to satisfy the emerging requirements of a variety of B5G applications.

Our Approach. Recently, Machine Learning (ML)—specifically Deep Reinforcement

Learning (DRL)—has proposed as an effective technique for tackling key wireless challenges, especially for resource management and power control in wireless communication networks [157]. However, how to enable DRL to assist wireless networks and User Equipments (UEs) in an intelligent and decision-making procedure is still wide open research area in cloud communication systems [158]. In this context, Deep Neural Network (DNN) has been integrated into cloud-based RANs as the representative technology of ML. The non-linear approximation performance feature of the full-connected DNN enables the DRL scheme to solve several problems in resource management, such as beamforming, channel association and power control. Reinforcement Learning (RL), due to its nice property of not requiring much training data, can be adopted as a feasible option for dealing with real-time decision-making problems, especially dynamic resource allocation, since the requirements of the system model and prior data in RL are less restricted. Hence, DRL provides a fast convergence rate and a high accuracy degree in wireless communication systems with large state and action spaces (e.g., multi-user systems). A well-known approach of DRL is Deep Deterministic Policy Gradient (DDPG). Using DDPG as a controller to optimize variables is a promising direction because DDPG is good at generating continuous actions given the system state. However, in this problem, two variables need to be optimized, the UE association, which is discrete, and the power allocation, which is continuous. Jointly optimizing these two variables together could cause a great increase in the amount of parameters needed and a bad performance because the action spaces contain both discrete and continuous spaces. Thus, we introduce a twin-actor approach where two actors are adopted with each one focus on one variable and a centralized critic to jointly criticize the actors. Furthermore, we argue that UE association and power allocation are closely-related, which indicates some level of overlap in the actors' parameters spaces. Therefore, we introduce the multi-task learning technique, which greatly reduces the amount of parameters and can increase the training efficiency. Compared with standard convex optimization approaches, DRL-based resource allocation algorithm can make real-time decisions given the description, i.e., the state of the network. This kind of intelligent decision-making is critical for most B5G services, especially those that demand real-time, low latency requirements, such as AR/VR applications and unmanned aerial vehicle control [159]. In this chapter, we

provide the system model for the NG-RAN system and formulate our DRL-based resource allocation, ReLAX, aiming at maximizing overall EE in NG-RAN under the constraints of QoS requirement, transmitted power budget, and limited fronthaul capacity.

Related Work. The cloud-based RAN paradigm has received significant attention in academia and industry over the past few years. In 2013, the Centralized-RAN (C-RAN) was presented as a step forward towards realizing the virtual-based RAN concept [160]. Besides, there is industry consensus within the 3rd Generation Partnership Project (3GPP) and IEEE to re-think existing cloud-based RAN architecture and evolve it towards the needs of the NG-RANs by splitting different parts of the radio stack between different network elements (CU, DU). Thus, recently IEEE has formed Next Generation Fronthaul Interface (NGFI) working group [161] to standardize transport fronthaul interface for future cellular networks. With the main target of integrating a full radio stack platforms and open door towards virtual-cloud-based RAN ecosystems, the architecture of NG-RAN will become intelligent and agile. However, how to properly manage various radio-computation resources in NG-RANs has become a key challenge and research focus in wireless communication field. For instance, energy allocation problems have been studied in [162, 163], while Fang *et al.* [164] have considered the user fairness in Multi-Carrier Non-Orthogonal Multiple Access (MC-NOMA) system. The joint user assignment and power allocation problems have been included in [65, 165]. Besides, the authors in [16] have proposed a resource allocation solution as a bin-packing problem aiming at minimizing the number of active Virtual Machines (VMs) in the cloud center.

In the meanwhile, DRL has become a new research trend in the B5G applications and been a feasible tool to address dynamic resource allocation problems for cloud-based RAN systems [166–168]. In [166], the authors have proposed a Deep Q Network (DQN) method to allocate power in the wireless network. The model has been firstly trained in the simulator following deep Q learning rule and, then, it has been deployed in the real environment for fine-tuning. However, the DQN showed a good performance for discrete action spaces while adopting it in continuous power models might lead to undesired performance. The work in [167] study a resource allocation method by designing a novel DNN-based optimization approach solution comprising of a series of alternating direction method of multipliers

iterative schemes that assign the CSI values as the learned weights. Besides, the authors in [168] present a three deep-reinforcement-learning-based scheme, which can solve the joint sub-channel assignment and power allocation problem in an uplink multi-user NOMA system to maximize the network EE. While DNN-based methods brought a significant gain in solving the resource management in the cloud-based wireless system, these studies often overlook the system challenges and mostly depend on simplified assumptions when modeling the radio-computation resources of the CUs and the DUs. Although these studies included the resource allocation problems from different perspectives separately, they did not take into account the dynamic of resource allocation problem in the NG-RAN scenario, in which the functional splits are supported. In this chapter, we formulate a DRL-based resource allocation for NG-RAN systems under the realistic circumstances of required QoS and limited fronthaul capacity and transmitted power. Besides, we support our model by real-time experiments carried out on the fully containerized-based NG-RAN testbed. The main contributions of this work are listed as follows.

- We study and investigate the resource allocation for NG-RAN system. Then, a comprehensive network system model, including wireless link model, fronthaul model, and network power consumption model, is presented to simulate the real features of PHY/MAC layers in the NG-RAN architecture.
- Subjecting to QoS, fronthaul capacity, functional splitting, and transmission power budget for DU, we mathematically formulate and analyze the resource allocation optimization problem in NG-RAN as a Mixed-Integer Non-Linear Programming (MINLP) problem jointly optimizing the UE association, and DU transmit power. The objective function in the proposed problem is modeled as the network EE, which is defined as the ratio between achievable data rate to total power conception under several functional splitting scenarios.
- To deal with the complexity of the formulated optimization problem, we develop a deep learning-based framework—the modified from dual DDPG method and named ReLAx—which can dynamically find the optimal values of UE association, and transmitted power in the downlink NG-RAN systems.
- Using *OpenAirInterface* OAI software platform [49] and container-based virtualization

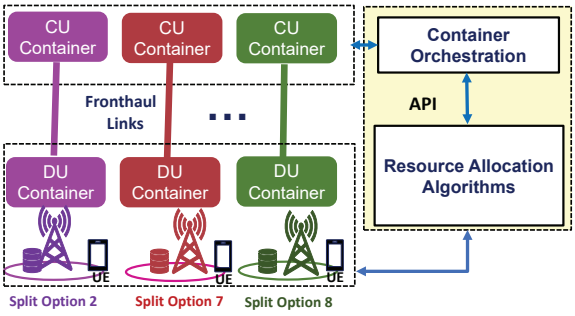


Figure 6.1: Proposed NG-RAN architecture with resource allocation algorithms.

technique, we integrate a real-time programmable NG-RAN testbed that can establish uplink/downlink wireless connection between the CU, DU, and the Commercial Off-The-Shelf (COTS) UE. The experimental results from the testbed show that the CU-DU CPU utilization depends on several network parameters (e.g., Physical Resource Block (PRB), and functional split options).

- Extensive simulations reveal that by giving the NG-RAN system model, the proposed ReLAx framework can optimize EE and show outperform competing algorithms, such as Deep Deterministic Policy Gradient (DDPG) and Weighted Minimum Mean Square Error (WMMSE).

Chapter Organization. The remainder of this chapter is organized as follows. In Sect. 6.2, we present the system model. Then, in Sect. 6.3, we formulate the EE maximization problem, followed by our proposed ML solution. In Sect. 6.4, we discuss experimental results and numerical simulations. Finally, we conclude the chapter in Sect. 6.5.

6.2 System Model

This section presents the network description, functional split model, wireless link model models, and the network power model.

6.2.1 Network Description

We consider that a Next Generation Node B (gNB) consists of multiple CUs connected to multiple DUs via a fronthaul interface—high-speed optical fiber. Figure 6.1 shows the logical diagram of the NG-RAN downlink transmission, which comprises L DUs, and U

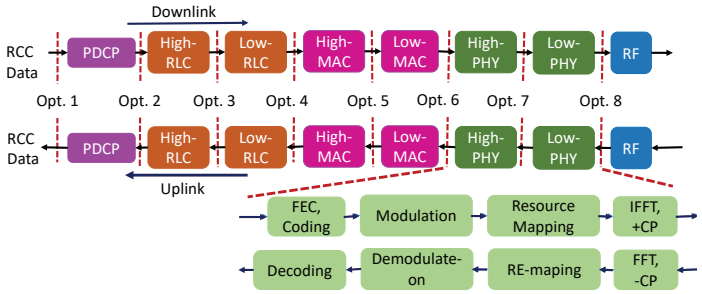


Figure 6.2: Split options as specified by 3GPP [2].

UEs. We denote $\mathcal{L} = \{1, 2, ..., L\}$ and $\mathcal{U} = \{1, 2, ..., U\}$ as the sets of the DUs and UEs, respectively. The Orthogonal Frequency Division Multiple Access (OFDMA) technique has been adopted to provide communication services in the downlink scenario. As a part of NG-RAN, 3GPP has proposed eight different functional split options, defined in 3GPP TR 38.801, between DU and CU, as shown in Fig. 6.2. Hence, we assume that the functional split technique is integrated into gNB while formulating the NG-RAN system's essential modes. In general, there are crucial benefits of enabling a flexible, functional split orchestration in NG-RAN. Some of them include cost-reducing, traffic load balancing, latency-fronthaul cost minimizing. It is worth noting that, in real NG-RAN testbed implementation, CU and DU can be deployed by virtualization technique. For example, in the OAI platform, each CU can be realized by one container image and associated with one DU container image. To connect our model with a real experimental NG-RAN testbed, we adopt the DU-to-CU model.

Functional Split Model. Placing all RAN functions in the CU pool can lead to the maximization the energy saving, however, adopting full centralized RAN architecture is not always realizable. For instance, physical layer processes, such as FTT, parallel/serial, Cyclic Prefix (see Fig. 6.2), have stringent latency requirements as well as incur high traffic capacity on the limited back/mid-haul interface if installed at CU. Hence, these processes are typically implemented on DU (e.g., 7.2 functional split option in O-RAN [169]). High PHY layer and MAC/RLC processes also have strict constraints, such as in LTE where the round-trip latency tolerance of MAC layer, synchronous HARQ technique, is $3ms$ [170]. However, in the 5G MAC layer adopting the fully asynchronous HARQ technique, strict latency requirement is no longer challenging and the round-trip time is mainly affected

Table 6.1: Practical functional split options for NG-RAN.

Split s	Split type	DU $\leftrightarrow$ CU
z_{j1}	No split, all at CU	$\leftrightarrow f_1, f_2, f_3$
z_{j2}	F1 split	$f_1 \leftrightarrow f_2, f_3$
z_{j3}	IF 4.5 split	$f_1, f_2 \leftrightarrow f_3$
z_{j4}	No split, all at DU	$f_1, f_2, f_3 \leftrightarrow$

Table 6.2: CPU load for RAN functions in NG-RAN at PRB=50.

Split function	CPU load (ω_s)
f_1	63%
f_2	21%
f_3	15%

by the service mode being provided. In the light of the above, it can be concluded that different RAN processes can be served by different 5G services, such as uRLLC, eMBB, and mMTC. Therefore, we consider the four practical CU/DU configurations based on the functional split selection described in Table 6.1. In our model, we denote the set of NG-RAN functions and the set of functional split options as $\mathcal{F}$ and $\mathcal{Z}$, respectively. A functional split $z_{js}, \forall j \in \mathcal{L}, s \in \{1, 2, 3, 4\}$, is performed at gNB j if all RAN functions above and including f_s are run at CU while RAN functions below f_s are run at DU. Hence, at functional split $z_{js} \in \mathcal{Z}$, CPU utilization at CU (ω) is equivalent to the summation of processing load of all RAN functions above and including f_s, i.e., $\omega_s = \sum_{j \geq s} \varrho_j$, where ϱ_j represents the CPU requirement at functional split s. Based on experimental results in Sect. 6.4.1, we have generated Table 6.2 to define the CPU processing load for downlink traffic. Besides, we define the CU-DU functional split indicator as, $z_{js} = 1$ indicates that gNB j runs at functional split s; otherwise z_{js} set to 0.

6.2.2 Wireless Link Model

We assume that each UE can establish a wireless connection with the DU thought up/down cellular links. Plus, the UE is considered in static and the cellular channels are invariant during each decision-resource allocation algorithm procedure. In this chapter, we adopt the downlink OFDMA system as a scheme for the NG-RAN proposed model. Consequently, the operational frequency band B is divided into N equal sub-bands of size $W = B/N$[Hz]. To meet the orthogonality property in our model, we consider that each UE is assigned

to one sub-band of the downlink transmission. Thus, each DU can serve at most N UEs at the same interval time. Furthermore, we consider large scale and small scale fading, where we assume that large scale fading is the same for all sub-bands and small scale fading is frequency-selective and flat. Define $g^n_{j,u}$ as the channel gain from DU j to UE u using sub-band n and it is calculated as,

$$g^n_{j,u} = \varpi_{j,u}|h^n_{j,u}|^2, \forall j \in \mathcal{L}, n \in \mathcal{N}, u \in \mathcal{U}, \tag{6.1}$$

where $\varpi_{j,u}$ is the large scale fading including pass loss and shadowing, and $h^n_{j,u}$ is the small-scale Rayleigh fading. To model the Rayleigh fading, we adopt Jake's model [171] and the small-scale fading is modeled as a first-order complex Gauss-Markov process and the update rule is,

$$h^n_{j,u} = \rho h^n_{j,u} + \sqrt{1-\rho^2} e^n_{j,u}, \forall j \in \mathcal{L}, n \in \mathcal{N}, u \in \mathcal{U}, \tag{6.2}$$

where $\rho = J_0(2\pi f_d T)$ is the correlation between two adjacent fading blocks, J_0 is the zero-order Bessel function of the first kind and f_d is the maximum Doppler frequency. T is the time separation that we re-estimate the channel gain. $e^n_{j,u}$ is the channel innovation process and they follow circularly symmetric complex Gaussian distribution. A greater value of ρ means that the channel has changed significantly since the last channel estimation, which could be caused by large T or a rapidly changing f_d. Let $\mathcal{N} = \{1, ..., N\}$ be the set of available sub-band at each DU. We denote the sub-channel association between UE u and sub-channel n of DU j as,

$$x^n_{ju} = \begin{cases} 1, & \text{UE } u \text{ associated with DU } j \text{ on sub-channel } n \\ 0, & \text{otherwise,} \end{cases} \tag{6.3}$$

Denote h^n_{ju} as the downlink channel gain between DU j on sub-band n and UE u, including the effect of path-loss, shadowing, and antenna gain. The DU-UE association usually occurs in a large period of time that is much larger than the time scale of small-scale fading Hence, similar to [172], the effect of fast-fading is assumed to be averaged out during the association.

Let p_{ju}^{n} denote the transmission power from DU j on sub-band j to UE u. Hence, the Signal-to-Interference-plus-Noise Ratio (SINR) from DU j on sub-band j to UE u is defined by,

$$\gamma_{ju}^{n} = \frac{p_{ju}^{n} h_{ju}^{n}}{\sum\limits_{k \in \mathcal{K} \setminus \{j\}} \sum\limits_{r \in U} x_{kr}^{n} p_{kr}^{n} h_{kr}^{n} + \sigma^{2}}, \tag{6.4}$$

where σ^2 is the variance of Additive White Gaussian Noise (AWGN). Then, the maximum achievable data rate of UE u using the sub-channel n in DU j can be calculated as,

$$R_{ju}^{n}(\mathcal{X}, \mathcal{P}) = W \log_2 \left(1 + \gamma_{ju}\right), \forall j \in \mathcal{L}, n \in \mathcal{N}, u \in \mathcal{U}, \tag{6.5}$$

where $\gamma_{ju} = \sum_{n \in N} \gamma_{ju}^{n}$ is the total SINR, $\mathcal{X} = \{x_{ju}^{n} | j \in \mathcal{L}, n \in \mathcal{N}, u \in \mathcal{U}\}$ and $\mathcal{P} = \{p_{ju}^{n} | j \in \mathcal{L}, n \in \mathcal{N}, u \in \mathcal{U}\}$ are to represent the UE assignment and power allocation, respectively. Hence, the sum-rate of the network R^T can be written as,

$$R^{T}(\mathcal{X}, \mathcal{P}) = \sum\nolimits_{j \in \mathcal{L}} \sum\nolimits_{n \in \mathcal{N}} \sum\nolimits_{u \in \mathcal{U}} R_{ju}^{n}(\mathcal{X}, \mathcal{P}) \tag{6.6}$$

Fronthaul Model. In NG-RAN, the baseband signal processing between the CUs and DUs is transmitted via a fronthaul interface, standardized as the *F1 interface* in 3GPP [156]. This kind of fronthaul transmission requires high speed data rate— $10\times$ more data rate than of the original one [173]. For this reason, the fronthaul link is considered the bottleneck of cloud-based RANs. To that end, 3GPP proposed a novel functional splitting technique to flexibly mange and control the data rate transmission between the CUs and the DUs in NG-RAN. Specifically, the functional split can significantly reduced the transmission cost by shifting part of the baseband signal processing operations from the CU to DUs [174]. In this chapter, we define the fronthaul capacity constraint by,

$$\sum\nolimits_{u \in \mathcal{U}} \sum\nolimits_{n \in \mathcal{N}} \sum\nolimits_{s \in \mathcal{Z}} x_{ju}^{n} z_{js} R_{ju}^{n}(\mathcal{X}, \mathcal{P}) \leq C_j, \forall j \in \mathcal{L}, \tag{6.7}$$

where C_j represents the fronthaul capacity of DU j. while ϵ is the ratio of the bandwidth that are demanded from the baseband transmission between CUs and DUs. Let C_j^{max} be considered as the fronthaul capacity of DU j. Hence, C_j can be expressed as, $C_j = C_j^{max}/\epsilon$,

where the value of C_j mainly relays on the fronthaul transmission technologies (e.g, optical fiber technology).

6.2.3 Computational Power Model

The computational power consumption for the downlink NG-RAN is modeled to be two main parts, the power consumption of the CUs and the power consumption of the DUs.

CU-power consumption. In general, the CUs can usually be realized virtually by VMs or containers. In this way, the CU containers' capacities can be dynamically modified to deal with variable traffic load and channel states. Thereby, the power consumption of the CUs depends on computing workload size while processing the baseband signals from DUs [87]. Hence, we can model the CU power computation to handle the baseband traffic from DU j as,

$$P_j^{\mathrm{CU}} = P_j^{\mathrm{C}} + \alpha_j \sum\nolimits_{u\in\mathcal{U}} \sum\nolimits_{n\in\mathcal{N}} \sum\nolimits_{s\in\mathcal{Z}} x_{ju}^n z_{js} \omega_s, \forall j \in \mathcal{L}, \tag{6.8}$$

where P_j^{C} represents the static power of CU j corresponding container of DU j. α_j represents the container power consumption factor determined by the architecture, traffic size, functional splitting mode, and hardware equipment of the CU pool.

DU-power consumption. Similarly, we can suppose that the DU power consummation consists of two main parts: static and dynamic power consumption. The static power consumption is needed to run the DU container while dynamic power consumption is usually proportional to the DU transmitted power, traffic workload, and network configurations. Hence, the power consumption of DU j can be modeled as,

$$P_j^{\mathrm{DU}} = P_j^{\mathrm{D}} + \beta_j \sum\nolimits_{u\in\mathcal{U}} \sum\nolimits_{n\in\mathcal{N}} \sum\nolimits_{s\in\mathcal{Z}} x_{ju}^n z_{js} (\omega_1 - \omega_s) p_{ju}^n, \tag{6.9}$$

where P_j^{D} models the static power consumption of DU j, and β_j is the power factor of DU j characterizing the link between the dynamic power consumption and the traffic load. The value of β_j is determined based on the architecture of the DU, traffic load, and the type of functional split mode. Hence, the power factor parameters β_j and α_j are detailed in Sect. 6.4.1. Based on the above considerations, the total network power consumption

model of NG-RAN can be expressed as, $P(\mathcal{X}, \mathcal{Z}, \mathcal{P}) = P_j^{\mathrm{CU}} + P_j^{\mathrm{DU}}$.

6.3 Energy Efficiency Maximisation

In this section, we formulate the EE maximization problem, followed by the proposed solution.

6.3.1 Problem Formulation and Relaxation

To efficiently use the radio resources (e.g., radio spectrum, transmit power) and computation capacity (e.g., fronthaul capacity, and CU-DU computation capacity) as well as meet the UEs' QoS requirement, we define the network EE of total system as a more effective objective for downlink NG-RAN systems. Hence, the network EE of NG-RAN is defined as,

$$\zeta_{\mathrm{EE}}(\mathcal{X}, \mathcal{Z}, \mathcal{P}) = \frac{R^T(\mathcal{X}, \mathcal{P})}{P(\mathcal{X}, \mathcal{Z}, \mathcal{P})} \tag{6.10}$$

The adaptive EE function in (6.10) quantitatively describes the effect of system performance brought by network achievable data rate and total power consumption. The main resource allocation problem can be formulated as,

$$\underset{\mathcal{X},\mathcal{Z},\mathcal{P}}{\mathrm{Max}} \ \zeta_{\mathrm{EE}}(\mathcal{X}, \mathcal{Z}, \mathcal{P}) \tag{6.11a}$$

$$\text{s.t.} \ \sum\nolimits_{u\in\mathcal{U}} \sum\nolimits_{n\in\mathcal{N}} \sum\nolimits_{s\in\mathcal{Z}} x_{ju}^n z_{js} R_{ju}^n \leq C_j, \forall j \in \mathcal{L}, \tag{6.11b}$$

$$\sum\nolimits_{u\in\mathcal{U}} \sum\nolimits_{n\in\mathcal{N}} p_{ju}^n \leq P_j^{\max}, \forall j \in \mathcal{L}, \tag{6.11c}$$

$$\sum\nolimits_{j\in\mathcal{L}} \sum\nolimits_{n\in\mathcal{N}} x_{ju}^n R_{ju}^n \geq R_u^{\min}, \forall j \in \mathcal{L}, \tag{6.11d}$$

$$\sum\nolimits_{u\in\mathcal{U}} x_{jun}^n \leq 1, \forall j \in \mathcal{L}, n \in \mathcal{N}, \tag{6.11e}$$

$$\sum\nolimits_{s\in\mathcal{Z}} z_{js} = 1, \forall j \in \mathcal{L}, \tag{6.11f}$$

$$x_{ju}^n = \{0, 1\}, \forall j \in \mathcal{L}, u \in \mathcal{U}, n \in \mathcal{N}, \tag{6.11g}$$

The constraints in (6.11) can be described as follows; the fronthaul capacity is modeled as the maximum tolerated data rate transmitted on the fronthaul link [175, 176]. Therefore, constraint (6.11b) implies that the fronthaul capacity of DU j must not exceed the jth

maximum fronthaul capacity system, C_j; constraint (6.11c) specifies the transmission power budget of each DU; constraint (6.11d) is to ensure the data rate requirement of each UE must accede the minimum data rate of each UE, $R_u^{\min}$; constraint (6.11e) restricts that each UE can serve on sub-band in each allocation decision; constraint (6.11f) restricts that each gNB j can be performed on one functional split option at each iteration; finally, constraint (6.11g) imposes the binary resource allocation variable in NG-RAN.

The fractional-form objective function in Problem (6.11) is non-convex. Besides, with the binary variables $\mathcal{X}$, and $\mathcal{Z}$, the optimization problem in (6.11) is a mixed integer non-linear programming (MINLP) problem which is NP-hard and difficult to be solved [177]. Similar to [65, 178], the primary problem in (6.11) is reformulated as,

$$\underset{\mathcal{X},\mathcal{Z},\mathcal{P}}{\text{Max}} \;\; R^T(\mathcal{X},\mathcal{P}) - \psi P(\mathcal{X},\mathcal{Z},\mathcal{P}) \tag{6.12a}$$

$$\text{s.t. } (6.11\text{b}) - (6.11\text{f}), \tag{6.12b}$$

where ψ denotes the network power consumption weight. The Lagrangian of problem in (6.12) is described as,

$$\begin{aligned}
\mathcal{L}(\mathcal{X},\mathcal{Z},\mathcal{P},\boldsymbol{\mu},\boldsymbol{\gamma},\boldsymbol{\upsilon}) = & \sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{n\in\mathcal{N}} x_{ju}^n R_{ju}^n(\mathcal{X},\mathcal{P}) \\
& - \psi[P_j^C + P_j^{\text{D}} + \alpha_j \sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{n\in\mathcal{N}}\sum\nolimits_{s\in\mathcal{Z}} x_{ju}^n z_{js}\omega_s \\
& + \beta_j \sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{n\in\mathcal{N}}\sum\nolimits_{s\in\mathcal{Z}} x_{ju}^n z_{js}(\omega_1-\omega_s)p_{ju}^n] \\
& - \sum\nolimits_{j\in\mathcal{L}} \mu_j \left(\sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{n\in\mathcal{N}}\sum\nolimits_{s\in\mathcal{Z}} x_{ju}^n z_{js} R_{ju}^n - C_j\right) \\
& - \sum\nolimits_{j\in\mathcal{L}} \gamma_j \left(\sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{n\in\mathcal{N}} p_{ju}^n - P_j^{\max}\right) \\
& + \sum\nolimits_{u\in\mathcal{U}} \upsilon_u \left(\sum\nolimits_{j\in\mathcal{L}}\sum\nolimits_{n\in\mathcal{N}} x_{ju}^n R_{ju}^n - R_u^{\min}\right)
\end{aligned} \tag{6.13}$$

where $\boldsymbol{\mu} = \{\mu_j | j \in \mathcal{L}\}$, $\boldsymbol{\gamma} = \{\gamma_j | j \in \mathcal{L}\}$, and $\boldsymbol{\upsilon} = \{\upsilon_u | u \in \mathcal{U}\}$ are the Lagrange multipliers. Hence, the Lagrange dual problem is,

$$\underset{\boldsymbol{\mu},\boldsymbol{\gamma},\boldsymbol{\upsilon}}{\text{Min}}\ \underset{\mathcal{X},\mathcal{Z},\mathcal{P}}{\text{Max}}\ \mathcal{L}(\mathcal{X},\mathcal{Z},\mathcal{P},\boldsymbol{\mu},\boldsymbol{\gamma},\boldsymbol{\upsilon}) \tag{6.14a}$$

$$\text{s.t.}\ \ \boldsymbol{\mu}\geq 0,\boldsymbol{\gamma}\geq 0,\boldsymbol{\upsilon}\geq 0 \tag{6.14b}$$

6.3.2 ML-based Proposed Solution

The major challenge in solving the resource allocation problem in (6.11) is that the integer variable x_{ju}^{n} makes the optimization problem a MIP problem that is in general non-convex and NP complete [148]. Besides, in real wireless network environments, the QoS, fronthaul link, and transmit power requirements update dynamically. Therefore, it is in general infeasible to adopt the traditional optimization solutions (e.g., standard convex solutions) to handle a resource management complexities. Hence, an deep reinforcement method is modified to deal with these challenges. Specifically, we will first provide a general background of traditional ML approaches to solve optimization problems and, then, we will present our proposed solution to solve the problem in (6.11).

Background on Reinforcement Learning. Problem that can be modeled as a Markov Decision Process (MDP) can be solved by RL algorithms. A MDP consists of a set of states $\mathcal{S}$, which characterizes the properties of the system, and a set of actions $\mathcal{A}$. Plus, the RL agent is deployed with a policy $\pi:\mathcal{S}\rightarrow\mathcal{A}$ parameterized by θ, which provide decisions given the state. After taking an action, the environment will change according to a state transition function $\mathcal{T}:\mathcal{S}\times\mathcal{A}\rightarrow\mathcal{S}$. The agent will receive a reward from the environment as a function of state and action $r:\mathcal{S}\times\mathcal{A}\rightarrow\mathbb{R}$. The goal of RL algorithms is to maximize the total expected return $R=\sum_{t=0}^{T}\gamma^{t}r_{t}$, where T is the maximum steps, and γ is a discount factor.

Deep Q Learning. A popular algorithm to solve MDP is Q Learning, which learns a Q table, $Q^{\pi}(s_t,a_t)$, recording the expected reward of state s_t and action a_t at time t. Q learning has a widely-used deep learning version DQN [179]. DQN replaces the Q table with a neural network as the function estimator. The goal of DQN is to learn an optimal Q-function, denoted as $Q^{*}(s_t,a_t)$, by iteratively minimizing the Temporal Difference (TD)

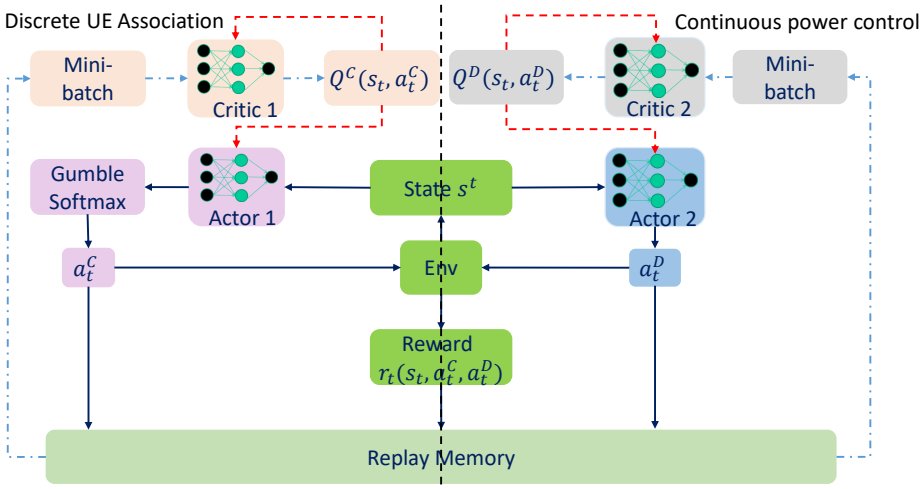


Figure 6.3: The framework and workflow of ReLAx. Solid lines indicate data flow, red dashed lines and blue dash-dotted lines represent forward and backward gradient propagation.

error [180]. Typically, the target Q network, $\bar{Q}$, is a copy of the Q network and it is updated in a predefined frequency while the Q network is updated every step. The purpose of using the target Q network is to stabilize the training process and reduce the model variance [179].

Policy Gradient (PG) Algorithms. Another popular algorithm that is different from DQN for solving MDP is the PG algorithm. Instead of optimizing the Q network, PG directly calculates the gradient of the policy function and optimize it towards the optimal direction. PG algorithms adjusts its parameters θ iteratively to maximize the expected long term reward. To approximate the state-action value, many works has been proposed. One popular framework is the actor-critic, which uses a neural network to approximate $Q(s_t, a_t)$. Actor functions as the policy which generates actions/decisions and critic will give scores to these actions. The critic, i.e., the Q network, is updated following (6.17).

Deterministic Policy Gradient (DPG). The algorithms introduced above treat the policy as a distribution over the action space, which makes it not capable to deal with continuous action spaces. To solve this, DPG is proposed, in which the policy μ_θ is regarded as a deterministic function. With the help of deep learning, Deep DPG (DDPG) algorithm is proposed, which introduces a neural network to estimate the Q function. The Q network, i.e., the critic, is trained by minimizing the estimation error between it and the real rewards.

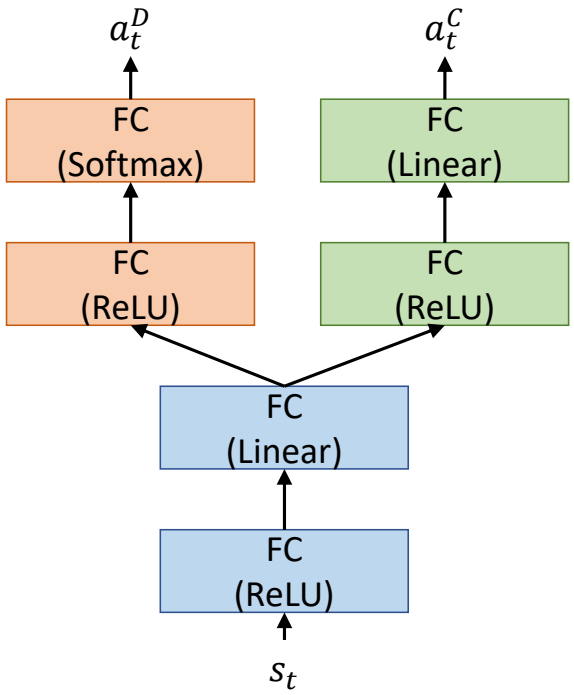


Figure 6.4: The structure of the multi-task actor structure. Each block/box represents a neural network layer. FC stands for fully-connected layer.

Algorithm 6 ReLAx algorithm for solving (6.11)

Initialize continuous actor θ^C, discrete actor θ^D, critic θ^Q and replay buffer $\mathcal{D}$
Initialize target networks, $\theta^{C'}$, $\theta^{D'}$ and $\theta^{Q'}$
for episode = 1, 2, ... **do**
  Initialize Ornstein-Uhlenbeck noise, $\mathcal{N}$
  Observe initial state s
  for step = 1, 2, ... **do**
    Sample discrete action, $a^C \sim \mu_{\theta^C}(s_t) + \mathcal{N}$
    Sample continuous action, $a^D \sim \mu_{\theta^D}(s_t) + \mathcal{N}$
    Execute actions and receive r_t and s_{t+1}
    Add new item $< s_t, a_t^C, a_t^D, r_t, s_{t+1} >$ to $\mathcal{D}$
    Sample a batch $(s_i, a_i^C, a_i^D, r_i, s_{i+1})$ from $\mathcal{D}$
    Update θ^C following Eq. (6.15)
    Update θ^D following Eq. (6.16)
    Update θ^Q following Eq. (6.17)
    if reaches target model update frequency **then**
      $\theta^{C'} \leftarrow \theta^C$
      $\theta^{D'} \leftarrow \theta^D$
      $\theta^{Q'} \leftarrow \theta^Q$

6.3.3 ReLAx Design in NG-RAN System

In the optimization problem (6.11), we face two types of challenges; (i) we are optimizing a discrete and a continuous variable simultaneously and the classic RL algorithm could

lead to a slow convergence rate and bad performance; (ii) as the number of DUs and UEs increase (i.e. the dimension of $\mathcal{X}$ and $\mathcal{P}$ increase), the required amount of parameters grows significantly (a.k.a. the curse of dimensionality [181]). To solve the challenges, we propose a dual DDPG framework, ReLAx. The proposed algorithm consists of a pair of actors and a centralized critic where one actor handles the sub-band allocation problem and the other one deals with the power allocation problem. However, the two variables are not completely independent to each other and there could be some level of overlap in the models' parameter spaces. Thus, in ReLAx, we adopt the multi-task training concept so that the actors can share parameters to reduce the number of parameters needed to be trained. To illustrate, in classic multi-task learning [182], the tasks will share a common model but with different output layers. In this way, the model can learn the correlation between the variables and improve its performance. Figure 6.3 shows the computational diagram of the proposed framework where the red dashed lines indicate backpropagation direction and blue dashed line represents the feed forward process in the update of the agent. We can see that the state for two actors are the same and they are supposed to make decisions on different aspects given the state. Figure 6.4 exhibits the structure of the two actors. They share the some parameters at the lower level and then have their own different layers at the end. The update rules for the actors follow the DDPG update rule. Nonetheless, variable $\mathcal{X}$ is discrete and DDPG can only handle continuous outputs. As such, we apply the Gumble Softmax trick [183]. This trick can transform the continuous output from DDPG to discrete outputs in a differetiable way.

Let θ^C and θ^D denote the continuous and discrete actors, respectively. The gradient for the actors can be written as,

$$\nabla_\theta J(\theta^C) = \mathbb{E}_{s\sim\mathcal{D}}\left[\nabla_\theta \mu_{\theta^C}(s_t)\nabla_a Q^\mu(s_t, a_t^C)|_{a_t=\mu_{\theta^C}(s_t)}\right], \tag{6.15}$$

$$\nabla_\theta J(\theta^D) = \mathbb{E}_{s\sim\mathcal{D}}\left[\nabla_\theta \mu_{\theta^D}(s_t)\nabla_a Q^\mu(s, g(a_t^D))|_{a_t=\mu_{\theta^D}(s_t)}\right] \tag{6.16}$$

where a_t^C and a_t^D stand for the continuous and discrete actions, i.e., UE association and power allocation, and $g(\cdot)$ represents the Gumbel Softmax. To optimize the critic, we need the information from both actors at the same time. The intuition behind this is

that, if we regard these two actors as separate agents, this problem becomes a Multi-Agent Reinforcement Learning (MARL). If the critic can only observe one agent once at a time, the whole environment will become dynamic and hard to optimize and a centralized critic can solve this problem [184]. In this problem, the loss function for the critic can be written as,

$$\mathcal{L}(\theta^Q) = \mathbb{E}_{s_t, a_t^C, a_t^D, r_t, s_{t+1}} \left[\left(Q^*(s_t, a_t^C, a_t^D | \theta^Q) - y \right)^2 \right] \tag{6.17}$$

$$y = r(s_t, a_t^C, a_t^D) + \gamma \max_{a_{t+1}^C, a_{t+1}^D} \bar{Q}^*(s_{t+1}, a_{t+1}^C, a_{t+1}^D), \tag{6.18}$$

The detailed workflow for ReLAx is shown in Algorithm 6. Besides the structure of the proposed framework, we need to formulate the MDP that ReLAx is trying to solve. We describe state, action, reward as follows,

State: We encode every helpful information describing the system and help the actors make decisions as the state. In this problem, the state is defined as a tuple consisting of the current subband allocation, power allocation and channel gain, $s_t = \{\mathcal{X}_{t-1}, \mathcal{Z}_{t-1}, \mathcal{P}_{t-1}, H_{t-1}\}$, where $\mathcal{X}_{t-1}$, $\mathcal{Z}_{t-1}$, $\mathcal{P}_{t-1}$, and H_{t-1} are the values of $\mathcal{X}$, $\mathcal{Z}$, $\mathcal{P}$, and channel gain from previous iterations. The actor is supposed to give good decisions based on these information.

Action: The action is the set of variables to be optimized. The action for variable $\mathcal{X}$ is defined as $a_t^D \in \{0, 1\}$ where each entry is an integer indicating the selected sub-band. The continuous action, $a_t^C \in \mathbb{R}$, represents variable $\mathcal{P}$. To meet the maximum power constraint described in (6.11c), we normalize the powers on each DU so that they do not exceed P_j^{max}.

Reward: Reward is the criterion of the action given the state and the greater the reward is, the better the agent performs. Thus, we defined the reward as the EE that we are trying to maximize

$$r_t = \zeta_{\text{EE}}(\mathcal{X}, \mathcal{Z}, \mathcal{P}) = \frac{R^T(\mathcal{X}_t, \mathcal{P}_t)}{P(\mathcal{X}_t, \mathcal{Z}_t \mathcal{P}_t)}. \tag{6.19}$$

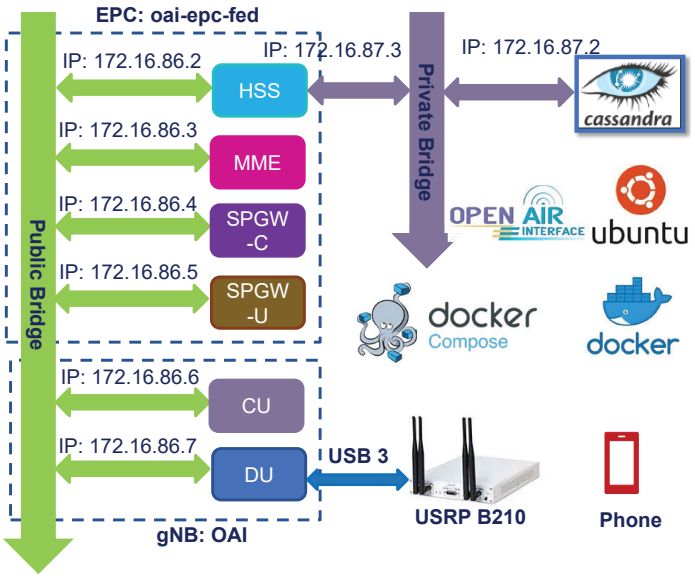


Figure 6.5: Logical illustration of the fully containerized-based NG-RAN testbed.

6.4 Performance Evaluation

We first provide the settings for experiments and results we obtained for the programmable NG-RAN testbed. Then, we evaluate ReLAx on resource allocation tasks via numerical simulation.

6.4.1 Testbed Experiment

We have implemented NG-RAN testbed as a non-standalone 5G architecture, including DU/CU and Core implementation. Then, we have performed our testbed in terms of packet delay, CPU utilization under various PRB, and functional split options. **NG-RAN Testbed Architecture.** We have utilized an open-source project, *OpenAirInterface* (OAI) [49] to build our experimental prototype. OAI has been fully tested and validated to compatible with the 5G protocol stack for gNB and UE allowing for end-to-end deployment of a 5G network. As illustrated in Fig. 6.5, We have implemented a RAN consisting of two containers; CU and DU, deployed in *Docker* [150]. Plus, we have used *oai-epc-fed* [185], an implementation of the 3GPP specifications concerning the Evolved Packet Core (EPC) networks, to implement the core network. The *oai-epc-fed* consists of the following network elements: Mobility Management Entity (MME), Home Subscription Server (HSS), and Packet Gateway and Service Gateway (SPGW-C-U). All *oai-epc-fed* components have

Table 6.3: Testbed Configuration Parameters for gNB.

Mode	FDD	**Options**	F1, IF4.5, eNB
Frequency	2.68 GHz	**PRB**	$25, 50, 100$
TX Power	150 dBm	**Env.**	Multi-container
MCS	28	**SINR**	$15 - 20$

been deployed in containers. Besides, we have used Software Defined Radio (SDR) boards—the brand-new Ettus USRP B210, each covering from 70 MHz–6 GHz and supporting 2×2 MIMO with sample rate up to 62MS/s. All containers are run by *Docker Compose* [186], in which we develop a *YAML* file to configure the containers. All containers are hosted by the desktop PC Intel Xeon E5-1650, 12-core at 3.5 GHz and 32 GB RAM. For UE, we use a Samsung Galaxy S9 running on Android 10. For network configuration, we run our NG-RAN prototype for three functional splits; Option F1, (PDCP/RLC, Option 2 in 3GPP TR 38.801 standard), Option IF4.5 (Lower PHY/Higher PHY, a.k.a Option 7.x in 3GPP TR 38.801 standard), and Option LTE eNB. We include the key testbed parameters in Table 6.3.

Impact Functional Splits on CPU Utilization. To understand better the CU-DU CPU power consumption, mentioned in Sect. 6.2.3, in NG-RAN with respect to the UEs' traffic request, we plan to set the related link between the functional split options and the percentage of CPU usage at the CU and DU. In this experiment, we record the CPU utilization percentage by using the *docker stats* command in Ubuntu, which provides a live data stream for running containers. We start repeatedly sending downlink UDP traffic from the SPGWU to the UE with different PRB values in two functional split settings, F1, and IF4.5. The percentage of the CPU usage has been measured for functional split Options F1, and IF4.5 in Fig. 6.6(b), and Fig. 6.6(c), respectively. One of the key notes from Fig. 6.6(b) is the CPU utilization of DU is reduced by 25.5% when we move from PRB 100 to PRB 50. However, lower CPU reduction, which is 2.7%, when moving from PRB 100 to PRB 50 in CU. The reason CPU is consumed higher in DU than in CU, in Option F1, is that the higher PHY operations such as RLC/MAC, L1/high, tx precode, rx combine, and L1/low operations reside in DU for split Option F1 [151], while CU has only PDCP and RCC operations. However, in Fig. 6.6(c), the trend of CPU consumption is different from Option F1. It can be observed that the highest CPU consumption occurred

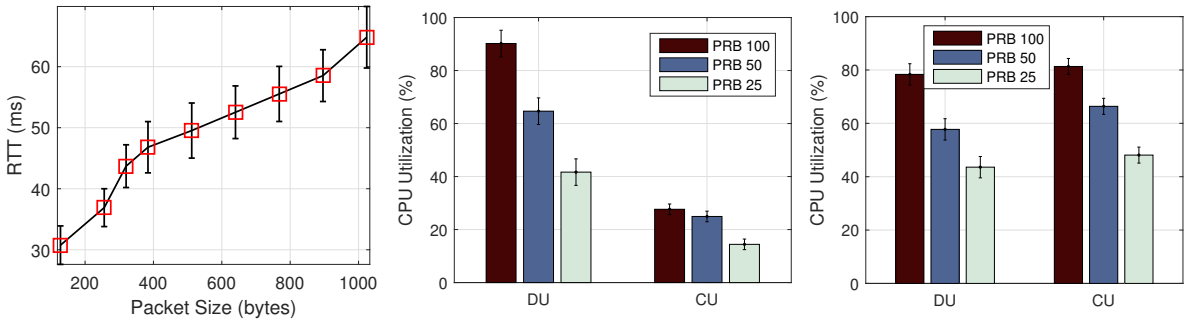
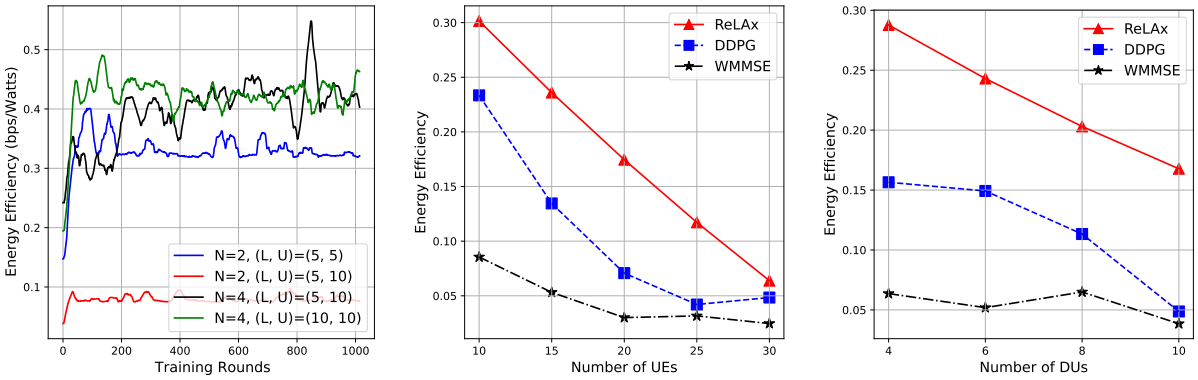


Figure 6.6: Fully containerized NG-RAN testbed experimental results for different configurations; (a) RTT measurement for different packet sizes; (b) CPU utilization of functional split Option F1 for downlink traffic; (c) CPU utilization of functional split Option IF4.5 for downlink traffic.

at CU, while CPU consumption of CU is reduced by 14.9% when we move from PRB 100 to PRB 50. In general, the power usage in the NG-RAN system can be remarkably minimized if we adapt the computational CU/DU resources such as the CPU cycles per second [16]. Based on experimental results in Figs. 6.6(b), and 6.6(c), *we can conclude that the power factor parameters α_j and β_j, in* (6.8) *and* (6.9)*, respectively, mainly depend on NG-RAN configurations (e.g., number of PRBs, and functional split options). The value of α_j is increased while moving from Option 1 to Option 8. However, the value of β_j is decreased while moving from Option 1 to Option 8. Specifically, the case of $\alpha_j > \beta_j$ is obtained, when CPU consumption in CU is higher than in DU such as in Fig. 6.6(c). Otherwise, the case of $\alpha_j \leq \beta_j$ will be estimated.*

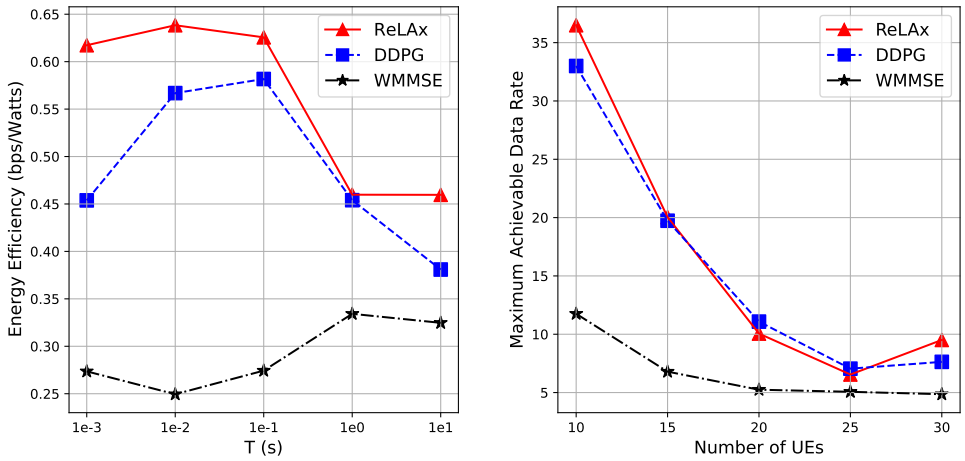
6.4.2 Numerical Simulations

Simulated results are mentioned here to evaluate the performance of our proposed algorithm. The simulations are conducted using Python with the assist of Pytorch toolkit [187]. We implemented the ReLAx framework along with other two methods—DDPG and WMMSE as comparison. ReLAx has an actor with three in-between Fully-Connected (FC) layers of size 64, 128 and 128, and two different output FC layers corresponding to the discrete and continuous actions. There are two critics in ReLAx and they are of the same shape where the state is processed by a FC layer of size 64 and the output is concatenated with the action. Then the concatenated will be processed by a FC layers of size 128. The nonlinear



(a) (b) (c)

Figure 6.7: The network EE versus (a) Training rounds; (b) The number of UEs; and (c) The number of DUs.



(a) (b)

Figure 6.8: (a) The network EE versus the time duration between two successive channel estimations; and (b) The maximum achievable data rate against the number of UEs.

function used in the network is Rectified Linear Unit (ReLU). A learning rate of 0.01 and 0.05 are used for actor and critic, respectively. Table 6.4 lists the common parameters used during our simulations. Our proposed solution for joint UE association and transmit power allocation problem is evaluated against the two popular approaches; i) DDPG, which is used the Deterministic Policy Gradient (DDPG) approach to solve the joint UE association and transmit power allocation problem; and ii) WMMSE, the sub optimal solution is generated by using so-called iterative Weighted Minimum Mean Square Error (WMMSE) [78, 162].

Convergence of the ReLAx Algorithm. To show that the proposed framework is able to achieve a good performance in the resource allocation scenario, in Fig. 6.7(a), the

Table 6.4: Simulation Parameters

Number of Sub-bands W	4
Maximum transmission power $P_j^{\max}$	100W
Static power P_j^C	10W
Static power P_j^D	10W
Maximum Doppler Frequency f_d	10Hz

Table 6.5: Value of ρ for different interval time T.

T (ms)	1	10	100	1000	10000
ρ	1.00	0.90	0.22	0.07	0.02

reward (i.e. the EE defined in (6.10)) against training rounds is exhibited. We observe three comparisons; (i) for the first two curves, we observe that, with the same number of sub-bands and DUs, the one with 5 UEs performs better than the one with 10 UEs. The result indicates that, under the same condition, the more UEs there are, the worse the performance of the model; (ii) for the second and third scenarios, we can see that, with the same number of DUs and UEs, the model with 4 sub-bands outperforms the one with 2 sub-bands. The result shows that more sub-bands can improve the model's performance; (iii) for the last two scenarios, we can observe that, with the same amount of UEs and sub-bands, the two models achieve similar performance, which is because that the amount of DUs are sufficient in both scenarios and, therefore, the two models converges to similar level.

Impact of the Number of UEs. Figure 6.7(b) shows the ReLAx approach solution comparing to different other methods versus the number of UEs. We can observe that when the amount of UEs increases, the performance of all three models degrades because the dimension of action spaces has increased. Furthermore, WMMSE and DDPG have similar bad performances. For WMMSE, the reason of this bad performance is that it only optimizes the set of the power variable $\mathcal{P}$ and, thus, when the number of UEs increases, it cannot assign links to good sub-bands. As for DDPG, it the action space becomes too large for it to learn a good representation and therefore its performance is much worse than ReLAx. 5 DUs and 5 subbands are used in this experiments with $T = 0.01$s.

Impact of the Number of DUs Figure 6.7(c) shows ReLAx's performance when varying the number of DUs in comparison with the other two methods. We can observe that the energy efficiency is not significantly impacted by the number of DUs except for

a small drop when increasing the number of DUs. The energy efficiency drop could be caused by the static DU and CU power consumption, because the improvement of data rate cannot compensate the increase of the power usage. 3 subbands and 20 UEs are used in this experiment with $T = 0.01$s.

Time processing of the ReLAx algorithm. T is the time interval between two channel estimation and the greater it is, the the more the channels change. From Fig. 6.8(a), we observe that while increasing T, the performance of all three models deteriorates, which is expected because, from a RL standpoint, the environment changes more significantly between two steps with a higher T and, thus, RL/WMMSE agent cannot depend on the knowledge it learned in former steps. The value of the channel change factor ρ is shown in Table 6.5. We can see that when T changes from 0.001s to 0.01s, the channel is relatively steady and when T reaches 0.1s, the value of ρ becomes 0.22 indicating that the channel becomes very hard to be estimated. Due to this phenomenon, ReLAx drops fast after $T = 1 \times 10^{-2}$s because the decision it makes is to maximize the reward in the environment from the last step. Conversely, other two models' performances are flat comparing to ReLAx's, and this is because they do not learn much from interacting with the environment and thus their actions are somehow random. Due to this randomness, their performance tends to be flat no matter how the environment changes. 5 DUs, 20 UEs and 3 subbands are used here.

Impact of Number of UEs on Maximum Achievable Data Rate. In Fig. 6.8(b), we show the results of the maximum achievable data rate against the number of UEs. As the number of UEs increase, ReLAx, DDPG and WMMSE show a performance drop, which is expected because the network is getting more and more crowded. Besides, ReLAx and DDPG have achieved similar maximum achievable data rate but has a significant performance difference in terms of energy efficiency (see Figure 6.7(b)). This proves that ReLAx can do a better job at power allocation than DDPG according to Eq. (6.10). 5 DUs and 3 subbands are used in this experiment.

6.5 Summery

We studied the network Energy Efficiency (EE) maximization problem in NG-RAN taking into account practical constraints including Quality of Service (QoS) requirement, transmit power, and fronthaul capacity. Based on real-world data collected from a programmable, realtime NG-RAN testbed, we established a conclusion to better understand the network power consumption model in the NG-RAN system. The proposed optimization problem is classified as a Mixed-Integer Non-Linear Programming (MINLP) problem, which is in general non-convex and NP complete. Therefore, we proposed a Deep Reinforcement Learning (DRL)-based algorithm—the modified dual DDPG method, named ReLAx. Simulation results coupled with real-time experiments on a fully containerized NG-RAN testbed show that the proposed approach solution, outperforms competing algorithms, such as DDPG and WMMSE.

Chapter 7

Conclusion and Future Directions

This chapter summarizes the main contributions of this dissertation and discusses future research directions.

7.1 Summary of Dissertation Contributions

This dissertation describes novel cooperative frameworks that exploit the synergies between resource allocation, video streaming, and task offloading computing in cloud-assisted wireless networks. The proposed innovations leverage the emerging 5G and beyound paradigms—C-RAN, MEC, and NG-RAN—to design optimized control policies aimed at making best use of the resources available to satisfy the data- and computation-service requests from mobile users. Firstly, given the high degree of cooperation provided by the centralized nature of C-RAN, we proposed a novel resource-allocation solution that aims at optimizing the energy consumption of a C-RAN. The proposed algorithm, with reasonably low-complexity, was demonstrated to significantly improve the system performance over traditional approaches. Secondly, we proposed a novel resource-allocation scheme that aims at maximizing the network energy efficiency of a C-RAN subject to practical constraints including QoS requirement, transmission power, and fronthaul capacity. Extensive simulation results coupled with real-time experiments on a small-scale C-RAN testbed showed the effectiveness of our proposed resource allocation scheme and its advantages over existing approaches. Thirdly, we focused on designing a dynamic video-streaming QoE maximization that takes into account the distortion rate characteristics of videos and the coordination among MEC server to enhance adaptive bitrate-video streaming in a MEC network. Real-time experiments on a wireless video streaming testbed performed on a FDD downlink LTE emulation system to characterize the performance and computing resource consumption of the MEC server

under various conditions. Emulation results of the proposed strategy showed significant improvement in terms of users' QoE over traditional approaches. Fourthly, task offloading can enhance the performance of mobile devices because servers in the edge cloud have higher computation capabilities than mobile devices. Therefore, enabling task offloading in NG-RAN is proposed to address the limitations (e.g., storage and computing resources) in the existing RANs. Meanwhile, in some cases, processing the entire input data in edge cloud servers would require more than the available computing resources to meet the desired latency/throughput guarantees. In the context of NG-RAN applications, transferring, managing, and analyzing large amounts of data in an edge cloud would be prohibitively expensive. Therefore, we proposed we proposed a multi-edge node task offloading system, i.e., QLRan, a novel optimization solution for latency and quality tradeoff task allocation in NG-RANs. Considering constraints on service latency, quality loss, edge capacity, and task assignment, the problem of joint task offloading, latency, and QLR is formulated in order to minimize the UEs task offloading utility, which was measured by a weighted sum of reductions in task completion time and QLR cost. Additionally, a programmable NG-RAN testbed was presented where the CU, DU, and UE were realized by USRP boards and fully container-based virtualization approaches. Specifically, we used OAI and Docker software platforms to deploy and perform the NG-RAN testbed for different functional split options. Simulation results showed that our algorithm performs significantly improved the network latency over different configurations. Finally, ML has recently proposed as an effective technique for tackling key wireless challenges, especially for resource management and power control in wireless networks. However, how to enable ML to assist wireless networks and UEs in an intelligent and decision-making procedure is still wide open research area in cloud communication systems. Therefore, we introduced a novel Deep Reinforcement Learning Based Resource Allocation (ReLAX) framework to deal with the joint optimization of UE association and power allocation in NG-RAN systems. Considering the dynamic nature of the NG-RAN environment, ReLAx problem was formulated to maximize the network EE under the constraints of QoS, fronthaul link, functional split configuration and transmit power budget. Simulation results showed that the proposed resource allocation solution outperforms competing traditional algorithms, such as ordinary DDPG and WMMSE.

7.2 Future Directions

In the following, we highlight and discuss the key open directions for future research.

Cloud-based Framework for Low-latency wireless mobile systems. With the proliferation of mobile computers (e.g. smart phones, tablets, wearable devices and PDAs), there has been interest in porting more computationally-intensive and delay-sensitive applications to the mobile domain. Some of these applications include face recognition, augmented reality, interactive gaming and video acceleration [12]. However, due to limited computing, memory and battery capacity, it is very challenging for mobile devices to run such tasks and expect the same performance. Several delay-sensitive hardware have been manufactured and promoted such as Google Glasses, Microsoft HoloLens, and the Recon Jet. AT&T has reported [188] that future low-latency applications are projected to target smart mobile devices. It is likely the widespread availability of an established platform is the motivation here. Telecompanies like AT&T plan on using new 5G and MEC to reduce the network latency and power consumption of mobile VR/AR applications. Despite improvements in hardware, mobile delay-sensitive applications still face ongoing struggles on the software side. There are a multitude of decisions as to how to allocate these newly available resources which can be layered and constrained in new ways. Problems such as indoor localization and navigation cannot be solved by simply throwing better hardware at the problem. While Global Positioning System (GPS) technology is highly extant and works well in automobile navigation, indoor navigation and error margins remain wanting [189]. Similarly, inertia based tracking suffers from error drifting due to environmental noise. For now, a hybrid approach using a multitude of sensors seems most promising to an effective low-latency application. Computer vision is more robust in detecting arbitrary features within a frame and does not rely on GPS. Henceforth, we shall use a hybrid approach that combines geographic data with visual data to better localize a user's position in space. Therefore, a novel cloud framework for ultra-reliable and low-latency should be designed for 5G networks to make the best use of the limited resources available in the network and satisfy the real-time service requests from the users and the ever increasing traffic demand.

Communication-efficient Federated Learning Design in NG-RAN. The rapid

progress and significant accomplishments that occurred in the artificial intelligence area have attracted a lot of attention and emerged the potential of machine learning schemes in beyond 5G applications. Plus, the increasing popularity of real-time mobile applications have placed strict constraints on cloud-based wireless access network architectures, such as ultra-low latency, QoS, privacy, and reliability. Moreover, the last decades have witnessed huge growth in portable and IoT equipment, making numerous limited computation capacity devices interact with wireless network systems via cellular channels. It is expected that the connected number of IoT devices increasing to 14.7 billion by 2023 [190]. However, the difficulty of satisfying private constraint and the high cost of transmitting the raw data to the central servers due to high *round trip latency* are driving the need for a highly decentralized machine learning approach. Motivated by this, federated learning has emerged to realize the collaborative training of a machine learning model without requiring to publish the original stream data with any third-party application. In such a scenario, it is possible for machine learning algorithms to gain experience from a vast range of data located at several locations. Enabling federated learning in NG-RAN is beneficial in terms of providing privacy for the end-users while running applications. Accordingly, the DUs and CUs can be enabled to collaborate on the development of tanning models, in a distributed machine learning manner, without needing to directly share sensitive data collected from user devices. Therefor, a novel federated learning algorithm can be efficiently applied in NG-RAN infrastructure to provide the user devices privileges in personal data protection and relieve the burden on fronthaul interface at the network.

References

[1] J. Fan, Q. Yin, G. Y. Li, B. Peng, and X. Zhu, "MCS selection for throughput improvement in downlink LTE systems," in *Proc. IEEE ICCCN*, pp. 1–5, 2011.

[2] 3GPP TR 38.801, "Study of new radio access technology: Radio access architecture and interfaces," *Release 14*, 2017.

[3] G. P. Fettweis, "A 5G wireless communications vision," *Microwave J.*, vol. 55, no. 12, pp. 24–36, 2012.

[4] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 65–75, 2014.

[5] N. Zhang, N. Cheng, A. P. T. Gamage, K. Zhang, J. W. Mark, and X. Shen, "Cloud assisted HetNets toward 5G wireless networks," *IEEE Commun. Mag.*, vol. 53, no. 6-Supplement, pp. 59–65, 2015.

[6] N. Zhang, N. Cheng, N. Lu, H. Zhou, J. W. Mark, and X. S. Shen, "Risk-aware cooperative spectrum access for multi-channel cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 3, pp. 516–527, 2014.

[7] 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.

[8] 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.

[9] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140, 2013.

[10] FP7 European Projec, "Distributed Computing, Storage and Radio Resource Allocation Over Cooperative Femtocells (TROPIC)." Available: http://www.ict-tropic.eu/, 2012.

[11] K. Wang, M. Shen, J. Cho, A. Banerjee, J. Van der Merwe, and K. Webb, "MobiScud: A fast moving personal cloud in the mobile network," in *Proc. Workshop All Things Cellular Oper. Appl. Challenge*, pp. 19–24, 2015.

[12] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[13] ETSI, "Mobile edge computing (MEC): Framework and reference architecture," *DGS MEC*, vol. 3, 2016.

[14] I. Chih-Lin, Y. Yuan, J. Huang, S. Ma, C. Cui, and R. Duan, "Rethink fronthaul for soft RAN," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 82–88, 2015.

[15] L. Wang and S. Zhou, "Flexible functional split and power control for energy harvesting cloud radio access networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1535–1548, 2019.

[16] A. Younis, T. X. Tran, and D. Pompili, "Bandwidth and energy-aware resource allocation for cloud radio access networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6487–6500, 2018.

[17] 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.

[18] A. Younis and D. Pompili, "PhD Forum: Resource allocation and task offloading in cloud-assisted wireless networks," in *Proc. IEEE WoWMoM*, pp. 1–3, 2019.

[19] A. Younis, T. X. Tran, and D. Pompili, "Fronthaul-aware Resource Allocation for Energy Efficiency Maximization in C-RANs," in *Proc. IEEE ICAC*, pp. 91–100, 2018.

[20] A. Younis, T. Tran, and D. Pompili, "Energy-efficient resource allocation in C-RANs with capacity-limited fronthaul," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 473–487, 2021.

[21] A. Younis, T. X. Tran, and D. Pompili, "On-demand video-streaming quality of experience maximization in mobile edge computing," in *Proc. IEEE WoWMoM*, pp. 1–9, 2019.

[22] A. Younis, T. X. Tran, B. Qiu, and D. Pompili, "Demo abstract: Mobile augmented reality leveraging cloud radio access networks," in *Proc. IEEE WoWMoM*, pp. 1–3, 2019.

[23] A. Younis, B. Qiu, and D. Pompili, "Latency and quality-aware task offloading in multi-node next generation RANs," *Computer Commun.*, vol. 184, pp. 107–117, 2021.

[24] A. Younis, B. Qiu, and D. Pompili, "QLRan: Latency-qualitytradeoffs and task offloading in multi-node next generation RANs," in *Proc. IEEE WONS*, pp. 1–8, 2021.

[25] A. Younis, T. X. Tran, B. Qiu, and D. Pompili, "Energy-latency-aware task offloading and approximate computing at the mobile edge," in *Proc. IEEE MASS*, pp. 1–9, 2019.

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

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] T. X. Tran and D. Pompili, "Octopus: A cooperative hierarchical caching strategy for cloud radio access networks," in *Proc. IEEE MASS*, pp. 154–162, 2016.

[34] 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.

[35] 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 Comput.*, 2018.

[36] ORACLE Cloud, "Enterprise-grade cloud solutions: SaaS, PaaS, and IaaS." Available: https://cloud.oracle.com/, 2018.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] H. Wu, Y. Sun, and K. Wolter, "Energy-efficient decision making for mobile cloud offloading," *IEEE Trans. Cloud Comput.*, 2018.

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

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] R. Schooler, "Transforming networks with NFV and SDN," *Intel Architecture Group*, 2013.

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

[49] EURECOM, "OpenAirInterface." http://www.openairinterface.org/.

[50] 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.

[51] 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.

[52] 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.

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

[54] 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.

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

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

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

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

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

[67] MOSEK Aps, *The MOSEK optimization toolbox v 9*, 2019.

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

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

[70] 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.

[71] Visual Networking Index Cisco, "Global mobile data traffic forecast update, 2016–2021," *Tech. Rep*, 2017.

[72] T. E. Bogale and L. B. Le, "Massive MIMO and mmWave for 5G wireless HetNet: Potential benefits and challenges," *IEEE Veh. Technol. Mag.*, vol. 11, no. 1, pp. 64–75, 2016.

[73] China Mobile Research Institute, "C-RAN: The road towards green RAN," *White Paper*, vol. 3, 2013.

[74] T. X. Tran, A. Hajisami, and D. Pompili, "QuaRo: A queue-aware robust coordinated transmission strategy for downlink C-RANs," in *Proc. IEEE SECON*, pp. 1–9, 2016.

[75] 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.

[76] C. Liu, K. Sundaresan, M. Jiang, S. Rangarajan, and G.-K. Chang, "The case for re-configurable backhaul in cloud-RAN based small cell networks," in *Proc. IEEE INFOCOM*, pp. 1124–1132, 2013.

[77] B. Dai and W. Yu, "Energy efficiency of downlink transmission strategies for cloud radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 1037–1050, 2016.

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

[79] X. Huang, G. Xue, R. Yu, and S. Leng, "Joint scheduling and beamforming coordination in cloud radio access networks with QoS guarantees," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5449–5460, 2016.

[80] 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.

[81] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. Leith, "Joint optimization of edge computing architectures and radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2433–2443, 2018.

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

[83] C. Pan, H. Zhu, N. J. Gomes, and J. Wang, "Joint precoding and RRH selection for user-centric green MIMO C-RAN.," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2891–2906, 2017.

[84] C. Pan, H. Zhu, N. Gomes, and J. Wang, "Joint user selection and energy minimization for ultra-dense multi-channel C-RAN with incomplete CSI," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 8, pp. 1809–1824, 2017.

[85] P. Rost, S. Talarico, and M. C. Valenti, "The complexity–rate tradeoff of centralized radio access networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6164–6176, 2015.

[86] D. Bega, A. Banchs, M. Gramaglia, X. Costa, and P. Rost, "CARES: Computation-aware scheduling in virtualized radio access networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 7993–8006, 2018.

[87] M. Khan, R. S. Alhumaima, and H. S. Al-Raweshidy, "Reducing energy consumption by dynamic resource allocation in C-RAN," in *Proc. IEEE EuCNC*, pp. 169–174, 2015.

[88] G. Miao, N. Himayat, and G. Y. Li, "Energy-efficient link adaptation in frequency-selective channels," *IEEE Trans. Commun.*, vol. 58, no. 2, 2010.

[89] K. G. Murty and S. N. Kabadi, "Some NP-complete problems in quadratic and nonlinear programming," *Mathematical Programming*, vol. 39, no. 2, pp. 117–129, 1987.

[90] W. Dinkelbach, "On nonlinear fractional programming," *Management Science*, vol. 13, no. 7, pp. 492–498, 1967.

[91] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed l0 norm," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 289–301, 2009.

[92] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted L 1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5, pp. 877–905, 2008.

[93] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, pp. 4331–4340, 2011.

[94] B. K. Sriperumbudur, D. A. Torres, and G. R. Lanckriet, "A majorization-minimization approach to the sparse generalized eigenvalue problem," *Machine learning*, vol. 85, no. 1, pp. 3–39, 2011.

[95] S. K. Joshi, P. C. Weeraddana, M. Codreanu, and M. Latva-Aho, "Weighted sum-rate maximization for miso downlink cellular networks via branch and bound," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 2090–2095, 2012.

[96] Evolved Universal Terrestrial Radio Access (E-UTRA), "Medium Access Control (MAC) Protocol Specification," 2012.

[97] L. Valcarenghi, K. Kondepu, A. Sgambelluri, F. Cugini, P. Castoldi, R. A. Morenilla, D. Larrabeiti, and B. Vermeulen, "SDN-controlled energy-efficient mobile fronthaul: An experimental evaluation in federated testbeds," in *Proc. IEEE EuCNC*, pp. 298–301, 2016.

[98] L. Shi, B. Mukherjee, and S.-S. Lee, "Energy-efficient PON with sleep-mode ONU: progress, challenges, and solutions," *IEEE Netw.*, vol. 26, no. 2, pp. 36–41, 2012.

[99] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update," *White Paper*, March 28, 2017.

[100] Huawei Technologies Co., Ltd, "4.5G, opening Giga mobile word, empowering vertical markets," *White Paper*, 2016.

[101] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Tran. IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[102] T. Stockhammer, "Dynamic adaptive streaming over HTTP–: standards and design principles," in *Proc. ACM Multimedia systems*, pp. 133–144, 2011.

[103] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. ACM Multimedia systems*, pp. 157–168, 2011.

[104] W. Shi and S. Dustdar, "The promise of edge computing," *IEEE Computer*, vol. 49, no. 5, pp. 78–81, 2016.

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

[106] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, 2016.

[107] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, pp. 1107–1115, 2012.

[108] R. K. Mok, X. Luo, E. W. Chan, and R. K. Chang, "QDASH: a QoE-aware DASH system," in *Proc. ACM MMSys*, pp. 11–22, 2012.

[109] T. C. Thang, H. T. Le, H. X. Nguyen, A. T. Pham, J. W. Kang, and Y. M. Ro, "Adaptive video streaming over HTTP with dynamic resource estimation," *IEEE J. Commun. Netw.*, vol. 15, no. 6, pp. 635–644, 2013.

[110] F. Jokhio, A. Ashraf, S. Lafond, I. Porres, and J. Lilius, "Prediction-based dynamic resource allocation for video transcoding in cloud computing," in *Proc. IEEE PDP*, pp. 254–261, 2013.

[111] D. K. Krishnappa, M. Zink, and R. K. Sitaraman, "Optimizing the video transcoding workflow in content delivery networks," in *Proc. ACM MMSys*, pp. 37–48, 2015.

[112] H. A. Pedersen and S. Dey, "Enhancing mobile video capacity and quality using rate adaptation, RAN caching and processing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 996–1010, 2016.

[113] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "QoE-driven cache management for HTTP adaptive bit rate streaming over wireless networks," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1431–1445, 2013.

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

[115] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[116] C. Ge, N. Wang, S. Skillman, G. Foster, and Y. Cao, "QoE-driven DASH video caching and adaptation at 5G mobile edge," in *Proc. ACM Information-Centric Netw.*, pp. 237–242, 2016.

[117] T. Tran and D. Pompili, "Adaptive bitrate video caching and processing in mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, 2018.

[118] V. R. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the k-user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, 2008.

[119] K. Gomadam, V. R. Cadambe, and S. A. Jafar, "A distributed numerical approach to interference alignment and applications to wireless interference networks," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3309–3322, 2011.

[120] V. Joseph and G. de Veciana, "NOVA: QoE-driven optimization of DASH-based video delivery in networks," in *Proc. IEEE INFOCOM*, pp. 82–90, 2014.

[121] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, 2000.

[122] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, "Mixed-integer nonlinear optimization," *Acta Numerica*, vol. 22, pp. 1–131, 2013.

[123] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, "Notes on decomposition methods," *Notes for EE364B, Stanford University*, pp. 1–36, 2007.

[124] J. Clausen, "Branch and bound algorithms—principles and examples," *Dept. Computer Science, U. Copenhagen*, pp. 1–30, 1999.

[125] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS long term evolution: From theory to practice*. John Wiley & Sons, 2011.

[126] 3GPP TS 136.213, "Physical layer procedures," *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*, 2016.

[127] ETSI TS 136 101, "LTE; Evolved universal terrestrial radio access (E-UTRA); User equipment (UE) radio transmission and reception," *3GPP TS 36.101 ver. 14.3.0 Rel. 14*, 2017.

[128] O. Østerbø, "Scheduling and capacity estimation in LTE," in *Proc. International Teletraffic Congress*, pp. 63–70, 2011.

[129] D. Schoolar, "Mobile video requires performance and measurement standards," *White Paper*, 2015.

[130] I. A. Alimi, A. L. Teixeira, and P. P. Monteiro, "Toward an efficient C-RAN optical fronthaul for the future networks: A tutorial on technologies, requirements, challenges, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 708–769, 2017.

[131] 3GPP TS 38.300 V2.0.0, "NR; NR and NG-RAN overall description; Stage 2," *Release 15*, 2017.

[132] Y. Li, Y. Chen, T. Lan, and G. Venkataramani, "MobiQoR: Pushing the envelope of mobile edge computing via quality-of-result optimization," in *Proc. IEEE ICDCS*, pp. 1261–1270, 2017.

[133] A. Younis, T. X. Tran, and D. Pompili, "Energy-latency-aware task offloading and approximate computing at the mobile edge," in *Proc. IEEE MASS*, pp. 299–307, 2019.

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

[135] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, *et al.*, "MEC in 5G networks," *ETSI white paper*, vol. 28, pp. 1–28, 2018.

[136] CISCO, "Fog computing and the Internet of things: Extend the cloud to where the things are," *whit paper*, pp. 1–6, 2015.

[137] O-RAN alliance, "O-RAN use cases and deployment scenarios," *White Paper*, 2020.

[138] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, 2018.

[139] M. Qin, N. Cheng, Z. Jing, T. Yang, W. Xu, Q. Yang, and R. R. Rao, "Service-oriented energy-latency tradeoff for IoT task partial offloading in MEC-enhanced multi-RAT networks," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1896–1907, 2021.

[140] Z. Zhao, S. Bu, T. Zhao, Z. Yin, M. Peng, Z. Ding, and T. Q. Quek, "On the design of computation offloading in fog radio access networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 7136–7149, 2019.

[141] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2536–2549, 2020.

[142] V. Kshirsagar, M. Baviskar, and M. Gaikwad, "Face recognition using eigenfaces," in *Proc. IEEE ICCRD*, pp. 302–306, 2011.

[143] I. de Fez, R. Belda, and J. C. Guerri, "New objective QoE models for evaluating ABR algorithms in DASH," *Elsevier Computer Communications*, vol. 158, pp. 126–140, 2020.

[144] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, 2014.

[145] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, 2016.

[146] Y. W. Bernier, "Latency compensating methods in client/server in-game protocol design and optimization," in *Game Developers Conference*, 2001.

[147] C.-P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Mathematical Programming*, vol. 66, no. 1-3, pp. 181–199, 1994.

[148] E. D. Andersen and K. D. Andersen, "Presolving in linear programming," *Springer Mathematical Programming*, vol. 71, no. 2, pp. 221–245, 1995.

[149] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, 2017.

[150] "Docker." `https://docs.docker.com/`, 2021.

[151] "OAI tutorials." `https://gitlab.eurecom.fr/oai/openairinterface5g/blob/develop/doc/FEATURE_SET.md#enb-phy-layer`, 2021.

[152] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[153] P. Pandey and D. Pompili, "Exploiting the untapped potential of mobile distributed computing via approximation," *Pervasive Mobile Comput.*, vol. 38, pp. 381–395, 2017.

[154] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proc. IEEE ISCC*, pp. 59–66, 2012.

[155] Cisco, "Cisco Annual Internet Report (2018–2023)," *White paper*, 2020.

[156] ETSI, "NG-RAN; Architecture description," *3GPP TS 38.401 Ver. 15.2.0 Rel. 15*, 2018.

[157] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, 2016.

[158] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 2018.

[159] V. Sadhu, C. Sun, A. Karimian, R. Tron, and D. Pompili, "Aerial-DeepSearch: Distributed multi-agent deep reinforcement learning for search missions," in *Proc. IEEE MASS*, pp. 165–173, 2020.

[160] China Mobile Research Institute, "C-RAN: The road towards green RAN," *White paper*, 2013.

[161] IEEE Standard Association, "IEEE 1914.3 Standard for Radio over Ethernet Encapsulations and Mappings," 2018.

[162] A. Younis, T. Tran, and D. Pompili, "Energy-efficient resource allocation in C-RANs with capacity-limited fronthaul," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 473–487, 2021.

[163] B. Xu, P. Zhu, J. Li, D. Wang, and X. You, "Joint long-term energy efficiency optimization in C-RAN With hybrid energy supply," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11128–11138, 2020.

[164] F. Fang, Z. Ding, W. Liang, and H. Zhang, "Optimal energy efficient power allocation with user fairness for uplink MC-NOMA systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1133–1136, 2019.

[165] X. Huang, W. Fan, Q. Chen, and J. Zhang, "Energy-efficient resource allocation in fog computing networks with the candidate mechanism," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8502–8512, 2020.

[166] F. Meng, P. Chen, and L. Wu, "Power allocation in multi-user cellular networks with deep Q learning approach," in *Proc. IEEE ICC*, pp. 1–6, 2019.

[167] X. Liao, J. Shi, Z. Li, L. Zhang, and B. Xia, "A model-driven deep reinforcement learning heuristic algorithm for resource allocation in ultra-dense cellular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 983–997, 2019.

[168] X. Wang, Y. Zhang, R. Shen, Y. Xu, and F.-C. Zheng, "DRL-based energy-efficient resource allocation frameworks for uplink NOMA systems," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7279–7294, 2020.

[169] A. U. T. Yajima, T. Uchino, and S. Okuyama, "Overview of O-RAN Fronthaul Specifications," *NTT Docomo Tech. J.*, vol. 21, no. 1, 2019.

[170] W. Erik, "4G/5G RAN architecture: How to a split can make the difference," *Ericsson Technol. Rev*, vol. 93, no. 6, pp. 1–15, 2016.

[171] L. Liang, J. Kim, S. C. Jha, K. Sivanesan, and G. Y. Li, "Spectrum and power allocation for vehicular communications with delayed CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 458–461, 2017.

[172] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, 2013.

[173] J.-i. Kani, J. Terada, K.-I. Suzuki, and A. Otaka, "Solutions for future mobile fronthaul and access-network convergence," *IEEE J. Lightwave Technol.*, vol. 35, no. 3, pp. 527–534, 2016.

[174] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 146–172, 2019.

[175] Y. Zhou, J. Li, Y. Shi, and V. W. Wong, "Flexible functional split design for downlink C-RAN with capacity-constrained fronthaul," *IEEE Trans. Veh. Technol*, vol. 68, no. 6, pp. 6050–6063, 2019.

[176] V. N. Ha, L. B. Le, *et al.*, "Coordinated multipoint transmission design for cloud-RANs with limited fronthaul capacity constraints," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7432–7447, 2015.

[177] S. Burer and A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surveys Operations Research Manag. Sci.*, vol. 17, no. 2, pp. 97–106, 2012.

[178] Z. Zhou, M. Dong, K. Ota, J. Wu, and T. Sato, "Energy efficiency and spectral efficiency tradeoff in device-to-device (D2D) communications," *IEEE Wireless Commun. Lett.*, vol. 3, no. 5, pp. 485–488, 2014.

[179] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[180] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[181] R. E. Bellman, *Adaptive control processes: A guided tour*. Princeton U., 2015.

[182] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv:1706.05098*, 2017.

[183] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv:1611.01144*, 2016.

[184] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *arXiv preprint arXiv:1706.02275*, 2017.

[185] "Openair-epc-fed." `https://github.com/OPENAIRINTERFACE/openair-epc-fed`.

[186] "Docker Compose." `https://docs.docker.com/compose/`, 2021.

[187] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv:1912.01703*, 2019.

[188] AT&T Labs, "AT&T Edge Cloud (AEC) ," *White Paper*, 2017.

[189] M. Billinghurst, A. Clark, G. Lee, *et al.*, "A survey of augmented reality," *Foundations Trends Human-Comput. Interaction*, vol. 8, no. 2-3, pp. 73–272, 2015.

[190] Cisco Visual Networking Index, "Cisco annual internet report, 2018–2023," *Cisco white paper, USA*, 2020.