Optimized Resource Allocation in Mobile Edge Communication, 5G data multiplexing, and WiFi Channel Access

Lyutianyang Zhang

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:
Sumit Roy, Chair
Payman Arabshashi
Maryam Fazel

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Optimized Resource Allocation in Mobile Edge Communication, 5G data multiplexing, and WiFi Channel Access

Lyutianyang Zhang

Chair of the Supervisory Committee:
Sumit Roy
Electrical and Computer Engineering

Optimizing bandwidth utilization, transmission power and transmission latency while guaranteeing the quality of service (Qos) is oftentimes the objectives of interests in communication networks. For example, modern mobile traffic such as online gaming, mobile virtual/augmented reality (VR/AR) requires low latency, transmission power, and occupied bandwidth for better user experience and less utilized network frequency bandwidth. In this thesis, a novel bidirectional computation task model is proposed as an important use case in mobile edge communication networks, e.g., interactive AR/VR gaming service needs to render the live scene by jointly computing user features such as 3D positions and video data stored at the cloud server. In the bidirectional computation task model, each task is served via three mechanisms, i.e., local computing with local caching, local computing without local caching, and computing at the mobile edge computing server. To minimize the average utilize bandwidth, we formulate and optimize the joint caching and computing optimization problem under the latency, cache size and average power constraints. The proposed Lagrangian Relaxation (LR) plus concave-convex procedure is shown to outperform the baselines such as greedy algorithm and LR algorithm with simulations.

The obtained allocation policy reduces the communication bandwidth. However, which resource blocks (RBs) in orthogonal frequency-division multiple access (OFDMA) are occupied by such allocated bandwidth when multiple devices coexist is not yet answered by this policy, which is currently a challenging topic in both 5G and WiFi. Hence, we are further motivated to consider the enhanced channel accessing mechanism in the next-generation WiFi networks and the RB scheduling problem in 5G with ultra-reliability-low-latency communication (URLLC) requirement. Standardization for recently started IEEE 802.11be (Wi-Fi 7) Working Groups has focused on significant medium access control layer changes that emphasize the role of the access point (AP) for coordinating channel access due to the high collision probability with the distributed coordination function (DCF), especially in dense overlapping Wi-Fi networks. We propose a novel multi-AP coordination system architecture aided by a centralized AP controller (APC). Meanwhile, a deep reinforcement learning channel access (DLCA) protocol is developed to enhance the network throughput by enabling the coordination of APs. On the other hand, 5G New Radio (NR) is envisioned to efficiently support both enhanced mobile broadband (eMBB) and URLLC. In 5G NR, URLLC represents the task files in the application of online gaming and VR/AR with demanding QoS constraints. From the media access control (MAC) layer scheduling perspective, URLLC traffic has higher priority than eMBB traffic because URLLC traffic can pre-emptively puncture the existing eMBB traffic on the time scale of mini-slot to obtain the strict requirements of URLLC (achieve 99.999% reliability within 1 ms latency). The multiplexing of eMBB and URLLC traffic in 5G downlink transmission is investigated with the dual objectives of maximizing eMBB utility like proportional fairness for eMBB users while satisfying URLLC constraints. The resource allocation problem in each mini-slot is formulated as an integer programming (IP) with two solutions: 1) convex relaxation; 2) greedy algorithm. The simulation results show that our algorithms have a higher utility of eMBB users while satisfying the URLLC users' latency and reliability requirements than using the basic round robin in 5G standard.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I thank my supervisor, Dr. Sumit Roy. He has given me a lot of guidance on my research. He can always enlighten me by giving me instructions on how to choose a research topic that is not only academically but also industrially contributing. He is able to keep correcting my research direction and mistakes. I not only thank Dr. Roy for his research guidance but also for his instruction on life advice and career choice.

I thank all of my other current or previous supervisory committee members, Dr. Payman Arabshahi, Dr. Maryam Fazel, and Dr. Kurtis Heimerl for their time in supervising me.

I thank my colleagues and friends, Hao Yin, Liu Cao, Yaping Sun, Sian Jin, Xiangyu Gao, Ran Wei, Kun Su, Eddie Ting, Yujia Chen, Yuxiang Zhang, Yujie Zhou, Linyue Qiao, Xiao Lin, Yihan Jiang, Xuhang Ying, Wenqiu Ma, Ruixuan Wan, Yizhan Huang, Jinhao Tang, Tuoying Chen, Tianyi Zhang, Xiaoxiang Zhang, Yuewen Gao, Rongzhao Qiu, Ying Yu, Yuwei Ma, Hao Chen, Haotian Zhang, Yizhou Wang, Collin Brady, Yonghun Lee, Sachin Nayak, Yue Sun, Yue Yang, Yuanbo Li, Yi Hu, Zhaohao Feng, Yangwei Shi, Mugaged Osman, etc. for their consistent encouragement, collaboration, and accompany.

During my Ph.D. program, I had 2 internships in the industry. My first internship experience was in Nokia Bell Lab. I thank Dr. Moorut Prakash for hiring me and providing me with daily supervision and help. I also appreciate my second internship experience at Celona Inc. I thank my mentor, Dr. Vanlin Sathya, for hiring me and providing me with a lot of advice on 5G O-RAN product prototyping.

The five years of Ph.D. students were challenging. I went through a lot of research failures and self-doubts. Looking back, I feel I am a much stronger person than I was when I entered the University of Washington and I am confident to face future challenges.

DEDICATION

To my dear mom, dad and grandparents

Chapter 1

INTRODUCTION

1.1 Background

In any communication networks, there are various aspects about transmission power, latency, bandwidth utilization and etc. The advent of modern mobile traffic, e.g., online gaming, virtual reality/augmented reality (VR/AR) delivery and etc., incurs ultra-high requirement on the wireless bandwidth [1], latency, and reliability. For example, the mobile VR delivery requires the transmission rate on the order of G bit/s [2]. How to optimize the wireless bandwidth utilization while satisfying user-perceived quality of experience (QoE) becomes one of the most important concerns of network operators. Mobile edge network that equips the edge nodes of the mobile network is also expected to deliver packets with low latency and high reliability. The requirements from applications such as online gaming and VR/AR all point to state-of-the-art 5G and WiFi technologies that emphasize high communication speed, low latency, and high reliability.

The benefits of 5G are clear when taking into account the communication speed. Network latency is the time required for a set of data to travel between two communicating devices, which is much shorter than previous generations of cellular technology. A network with high latency can be inefficient for online gaming, VR/AR user experience. Latency is affected by a variety of factors such as the topology of the network, the available bandwidth, the size of the transmitted data, and etc. 5G enables users to send and download huge amounts of data in near real time. The design of 5G system intends to have extreme reliability greater than 99.999%, and to be capable of supporting ultra-reliable and low latency communications (URLLC) with less than 5 ms end-to-end system delay [3]. The URLLC reliability requirement for transmission of a 32-byte packet is 5×10^{-5} with the latency of 5

ms corresponding to a maximum block error rate (BLER) of 10^{-5} or 0.001%. The latency[1] in WiFi network is normally required to be less than 50 ms [4]. Currently, 5G mobile communication system has been actively investigated all over the world, in order to satisfy the future expected requirements, such as super high bit rate. Here, one of the approaches is to utilize the high-SHF (above 6 GHz) and EHF (mainly 30 – 60 GHz) bands [5]. These bands are very attractive because wider frequency bandwidth is relatively easily obtained. Moreover, for WiFi networks, the 802.11 standard provides several distinct radio frequency ranges for use in Wi-Fi communications: 900 MHz, 2.4 GHz, 3.6 GHz, 4.9 GHz, 5 GHz, 5.9 GHz and 60 GHz bands [6]. Although the operating frequency band has been settled for 5G and Wi-Fi systems, the approach to obtaining the minimum amount of frequency resources used for communication while maintaining the stringent constraints in 5G and Wi-Fi can is yet to be proposed. Furthermore, the optimal allocation policy of frequency resources under the standards of 5G and Wi-Fi is also yet to be proposed.

In this thesis, we consider complicated multi-objective resource allocation problems in different scenarios (Mobile edge networks, 5G NR, and Wi-Fi). Different chapters highlight resource allocation with different aspect; that is, with different focus in different communication networks, we investigate various topics emphasizing different network topology and constraints. In Chapter 2, we mainly consider a general problem that optimizes the bandwidth utilization under the latency, transmission power, computation, and cache constraints in mobile edge communication networks. In Chapter 3, we take a different resource allocation problem in 5G NR that aims to multiplex different types of data traffics under the latency and reliability constraints in 5G standard. In Chapter 4, a new channel accessing method in Wi-Fi network is proposed that optimizes the network throughput and proportional fairness of multiple Access Points, which follows current Wi-Fi 7 protocol with Access Point Coordination.

---

[1]The WiFi network latency is much longer than 5G system is because it is the last-mile latency in the Internet, which consumes much more time to deliver the user requested content.

1.2 Summary of Research Contributions

We proposed a novel bidirectional computation task model for modern communication networks. The joint caching and computing policy optimization problem is formulated to minimize the average consumed bandwidth subject to the latency, average power as well as cache size constraints. Based on the problem structure and convex optimal property, we obtain the optimal property between caching and computing as well as the consumed minimum bandwidth for each task under each service route. Based on the optimal properties, we transform the problem into a multiple dimensional multiple-choice knapsack problem (MMKP) and show that the formulated problem is NP-hard in strong sense. In the homogeneous scenario where the sizes of both the local input data and the remote input data, the output data size, as well as the computation load are the same for each task and the request probability is uniformly distributed, we obtain the optimal policy and the closed-form expression for the minimum average bandwidth. In the heterogeneous scenario, we introduce the Lagrangian relaxation (LR) method to obtain a suboptimal solution to the original problem. Since the obtained LR solution is not guaranteed to be feasible, we then formulate an auxiliary problem using local neighborhood search method based on the LR solution and solve it via Concave-Convex Procedure (CCCP). Based on the LR solution to the original MMKP and CCCP solution to the auxiliary problem, we construct the LR+CCCP solution, which is a feasibility-guaranteed local optimal solution. Our proposed LR-based algorithm is shown to outperform the baselines including the CCCP method as well as the heuristic greedy algorithm in terms of both bandwidth performance and time efficiency.

5G New Radio (NR) is envisioned to efficiently support both enhanced mobile broadband (eMBB) and ultra-reliable low-latency communication (URLLC). Due to the severe constraint of URLLC traffic, the time resources used by eMBB are further divided into mini-slots. From the media access control (MAC) layer scheduling perspective, the resource allocation occurs on different timescales, and URLLC traffic is pre-emptively overlapped at the mini-slot timescale to obtain the strict requirements of URLLC (achieve 99.999% reli-

ability within 1 ms latency). We analyze the multiplexing of eMBB and URLLC traffic in 5G downlink transmission, with the dual objectives of maximizing eMBB utility like proportional fairness for eMBB users while satisfying URLLC constraints. We formulate the resource allocation problem in each mini-slot as the integer programming (IP) problem with two solutions: 1) convex relaxation; 2) greedy algorithm. The simulation results show that our algorithms have a higher utility of eMBB users while satisfying the URLLC users' latency and reliability requirements than using basic round robin. The performance comparison between the two algorithms is extensively studied, and the choice of two algorithms in practice can be decided by the trade-off analysis investigated in Chapter 3.

Carrier sensing multiple access/collision avoidance (CSMA/CA) is the backbone MAC protocol for IEEE 802.11 networks. However, tuning the binary exponential back-off (BEB) mechanism of CSMA/CA in user-dense scenarios so as to maximize aggregate throughput still remains a practically essential and challenging problem. A novel coordinated multi-AP architecture and a corresponding channel access mechanism aligning with IEEE 802.11be is proposed to maximize the aggregate network throughput while preserving fairness among APs, which is based on the application of deep reinforcement learning (DRL) and First-Order Model-Agnostic Meta-Learning (FOMAML) in Chapter 4. The proposed method is named Deep reinforcement learning channel access (DLCA) protocol. A new Monte Carlo (MC) reward updating method for DRL training is proposed and the access history of each station is used to derive a DRL-based MAC protocol that improves the network throughput vis-a-vis the traditional distributed coordination function (DCF). For each AP in the coordinated multi-AP set, DLCA is deployed to contend for channel access. The first AP winning the contention gains the TXOP on its primary channel. The FOMAML is then applied to DLCA to enhance the overall performance. We also develop a greedy algorithm to maintain PF among APs. Simulation results show that the performance of DLCA protocol is verified to have strong stability and outperform baselines such as SH-TXOP and RTS/CTS in terms of the network throughput as well as the network utility in dense overlapping Wi-Fi networks. First-Order Model-Agnostic Meta-Learning further enhances the

network throughput. Subsequently, we also put forward a new greedy algorithm to maintain proportional fairness (PF) among multiple APs.

1.3 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2 of this thesis, we explored application of Lagrangian Relaxation, Concave Convex Procedure and their combinations for bandwidth-energy-latency-cache joint resource allocation in mobile edge computing network. In Chapter 3 of this thesis, we apply greedy algorithm and optimization theory for 5G NR eMBB and URLLC data multiplexing and fair scheduling. In Chapter 4 of this thesis, we use deep reinforcement learning to aid channel access in Wi-Fi 7 network with Access Point Coordination. This thesis includes the material in the author's previous papers published on IEEE [7–10][2].

1.4 Conventions and Notations

Throughout this thesis, we use $(\cdot)^{-1}$, $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^H$ to represent inverse, transpose, conjugate, and conjugate transpose. x, $\mathbf{x}$, and $\mathbf{X}$ denotes the scalar, vector, and matrix respectively. $\mathrm{Tr}(\bullet)$ denotes the trace. $\lceil \bullet \rceil$ and $\lfloor \bullet \rfloor$ denote integer round up and down.

[2]In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University of Washington's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

Chapter 2

BANDWIDTH-ENERGY-LATENCY-CACHE: RESOURCE ALLOCATION IN MOBILE EDGE COMPUTING

2.1 Motivation

The advent of modern mobile traffic, e.g., online gaming, mobile virtual reality (VR)/augmented reality (AR) delivery and etc., incurs ultra-high requirement on the wireless bandwidth [1]. For example, the mobile VR delivery requires the transmission rate on the order of G bit/s [2]. In Fig 2.1, mobile device captures frame at T_1 and communicate with Access Point (AP) for data exchange. Then, AP transmits server data back at T_3 to the mobile device to render the live scene at T_5, which is very common in most of the applications for better quality. Data exchange needs wireless bandwidth utilization and normally user experience asks for low latency; that is, $T_5 - T1$ should be small enough. Hence, how to optimize the bandwidth utilization while satisfying user-perceived quality of experience (QoE) becomes one of the most important concerns of network operators. In this chapter, we consider a resource allocation problem aiming to optimize bandwidth utilization that can be happen in any 5G or Wi-Fi network that is constrained by latency, power consumption and mobile device caching size.

Mobile edge network that equips the edge nodes of the mobile network, e.g., access point (AP) and the mobile devices, with caching and computing resources is deemed as one of the most promising approaches to alleviate the bandwidth burden on the mobile carriers [11]. In particular, mobile edge caching indicates proactively storing popular contents into the network edge nodes to reduce the traffic redundancy and transmission latency [12–14]. Mobile edge computing (MEC) refers to computing the tasks at the network edge nodes to reduce the core network burden and the latency [15–21]. How to efficiently utilize the caching and

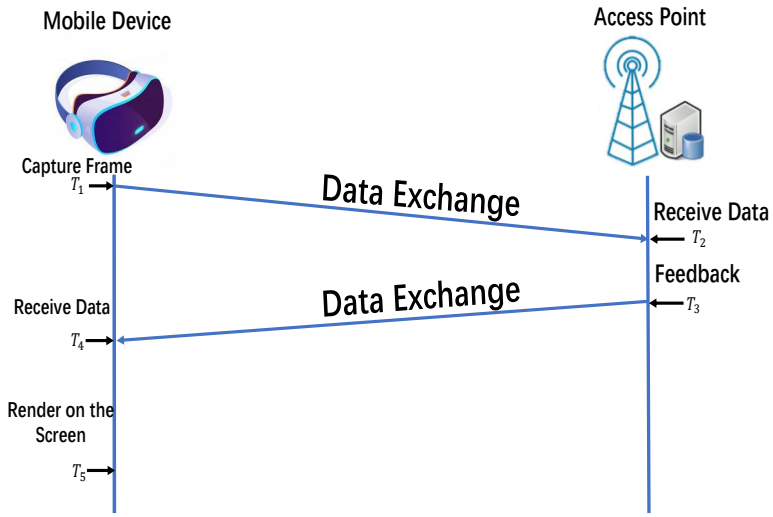


Figure 2.1: VR/AR Example

computing resources in the mobile edge networks triggers the research interests from both the academic and industrial areas [11–13, 18–20, 22–28].

The computation model in the currently existing literature in MEC can be named as *one-way* computation task model. That is, the input data of each computation task is assumed to be either generated at the mobile device [18–20] or originated from the Internet [22]. In particular, in [18–20], the mobile device offloads the input data to the AP for computation and then downloads the output data from the AP. In [22], when the task is computed at the mobile device, the mobile device has to download the input data from the AP first if not cached locally and then computes the input data to obtain the output data. Unlike the existing research on exploiting the MEC to facilitate one-way data computation [18–20, 22], in this chapter, we consider a *bidirectional* computation task model, where the input data consists of two parts, one of which is generated from the mobile device (local input data) and the other of which is originated from the Internet proactively (remote input data). One of the most directly motivating examples is online role playing game (RPG). Suppose one player is controlling a role and choosing which place/map to go. The location of the role selected

by the local player combined with the map information from the AP could help render the picture for the player after some computations. The input data consists of these generated at the mobile device including current player equipment selection, strategy selection as well as role selection, and also those proactively generated from the Internet such as the map information. This rendering task could be done either at the mobile device or at the AP. If the task is computed at the AP, the mobile device has to first upload the player's related information to the AP, and then the AP computes the task and at last, the mobile device downloads the computation result from the AP. On the other hand, if the task is computed at the mobile device, the mobile device has to first download the map information from the AP and then computes the rendering task. Since the required transmission load and the computation frequency when computing at the mobile device are different from those when computing at the AP, the corresponding consumed bandwidth also differs and thus the computing policy requires careful design.

Besides, the joint design of caching and computing policy is drawing increasing attention based on the fact that the input data or the output data of the computation task is cacheable. Resource allocation at the MEC/cloud server has been considered in [23–27]. In particular, [23] designs optimal computing offloading and caching policy for minimizing the latency in a hybrid mobile cloud/edge computation system. [24] studies the network-assisted video rate adaptation problem in an MEC-enabled base station with caching. [25] considers the joint optimization of task caching and offloading on edge cloud to minimize the energy consumption. [26] considers the expected latency minimization for each task service under the cache size constraint in the edge cloud. [27] considers the optimal fine-grained collaborative offloading strategies with caching-enhancements at the AP to minimize the overall execution delay of the mobile device. Moreover, all the works in [23–27] consider one-way computation task model in which the input data of each computation task is generated at the mobile device. On the other hand, resource allocation at the mobile devices has been considered in [22, 28]. In particular, [22] considers the joint caching and computing policy optimization to minimize the bandwidth consumption for the mobile VR delivery in a single-user scenario.

[28] considers the joint design in a multiple-user scenario by taking into account the multicast opportunity. All the works in [22,28] consider one-way computation task model in which the input data of each computation task is originated from the Internet proactively.

From above, we can see that the joint caching and computing design for the bidirectional computation task model to minimize the bandwidth consumption has not been addressed yet. Still, take the RPG as an example. The history of all the players' actions could provide a popularity distribution of the map preferences, e.g., the maps/places the players mostly like to go to. Based on a-priori knowledge of popularity, the popular maps/places could be proactively cached at the mobile device to save the consumed bandwidth. Inspired by this, under the latency, cache size and average power constraints, this paper jointly optimizes the computing and caching policy to minimize the average wireless bandwidth consumption for the bidirectional computation task model, and thereby reveals the tradeoff among communications, computing and caching (3C).

The main contributions of this work are summarized as below.

- *A Novel MEC-based System Model.* The novelty of the proposed MEC-based system model lies in the following aspects. First, the computation task is bidirectional, i.e., the input data of each computation task consists of the locally generated data and the proactively generated remote data. As mentioned above, such a computation task model can be seen in multiple currently existing applications, e.g., online gaming, mobile VR/AR delivery and etc. Secondly, based on the bidirectional computation task model, the service mechanism is novel. In particular, each task can be served via the following three routes: i) local computing with local caching if the remote input data has been cached at the mobile device and the task is computed at the mobile device; ii) local computing without local caching if the remote input data has to be delivered from the AP and the task is computed at the mobile device; iii) MEC computing if the local input data is uploaded to the AP and the task is computed at the AP. Different caching and computing policies at the mobile device may incur different bandwidth

consumptions, including both the uplink and downlink bandwidth, and thus require careful design. Last but not least, we take the average sum of the uplink and downlink consumed bandwidth as the optimization objective and performance metric, which has not been considered before in such a scenario.

- *Problem Formulation.* We formulate the joint caching and computing policy optimization problem to minimize the average consumed bandwidth subject to the latency, average power as well as cache size constraints. Based on the problem structure and convex optimal property, we obtain the optimal property between caching and computing as well as the consumed minimum bandwidth for each task under each service route. Based on the optimal properties, we transform the problem into a multiple dimensional multiple-choice knapsack problem (MMKP) and show that the formulated problem is NP-hard in strong sense.

- *Optimal Policy and Tradeoff Analysis in Homogeneous Scenario.* In the homogeneous scenario where the sizes of both the local input data and the remote input data, the output data size, as well as the computation load are the same for each task and the request probability is uniformly distributed, we obtain the optimal policy and the closed-form expression for the minimum average bandwidth. In this way, we analyze the 3C tradeoff and find that the 3C tradeoff exhibits different properties in different computing and caching capability regions.

- *Algorithm Design in Heterogeneous Scenario.* First, we introduce the Lagrangian relaxation (LR) method to obtain a suboptimal solution to the original problem. Since the obtained LR solution is not guaranteed to be feasible, we then formulate an auxiliary problem using local neighborhood search method based on the LR solution and solve it via Concave-Convex Procedure (CCCP). Based on the LR solution to the original MMKP and CCCP solution to the auxiliary problem, we construct the LR+CCCP solution, which is a feasibility-guaranteed local optimal solution. Our proposed LR-

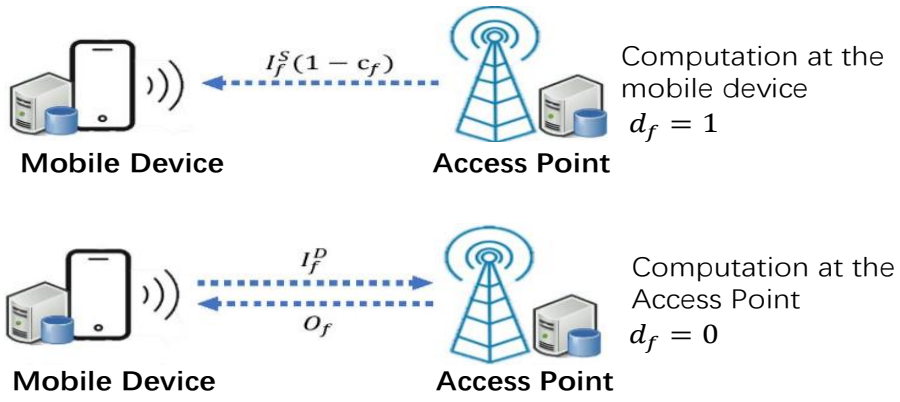


Figure 2.2: System Model

based algorithm is shown to outperform the baselines including the CCCP method as well as the heuristic greedy algorithm in terms of both bandwidth performance and time efficiency.

The rest of this chapter is organized as follows. Section II introduces the system model in terms of the task model, caching and computing model as well as the service mechanism. Section III formulates the bandwidth minimization problem and analyzes the computation complexity. The optimal computing and caching policy is obtained for the homogeneous scenario in Section IV. LR+CCCP algorithm is proposed in Section V to obtain an improved feasible suboptimal solution. Simulation results are shown in Section VI. In the end, the conclusion and future work discussion are given in Section VII.

2.2 System Model

As illustrated in Fig 2.2, we consider a mobile edge network consisting of one AP and one mobile device, both equipped with certain caching and computing abilities. The mobile device is assumed to request one task at each time. Key notations are summarized in Table 2.1.

Table 2.1: Key Notations

I_f^D, I_f^S	Size of the local/remote input data of task f
O_f	Size of the output data of task f
τ	Maximum tolerable service latency
C, $\bar{P}$	Cache size and the average available power at the mobile device
f_D, f_S	Computation frequency of the mobile device and the AP
w_f	Required computation cycles per bit
$d_f \in \{0,1\}$	$d_f = 1$ means that task f is computed at the mobile device and the AP, otherwise
$c_f \in \{0,1\}$	$c_f = 1$ means that the remote input data of task f is cached at the mobile device and not otherwise.

2.2.1 Task Model

Assume that there are F tasks in total to be requested by the mobile device. Denote with $\mathcal{F} \triangleq \{1, 2, \cdots, f, \cdots, F\}$ the task set. Each task $f \in \mathcal{F}$ is characterized by a 5-item tuple $\left\{I_f^D \text{ (in bits)}, \ I_f^S \text{ (in bits)}, O_f \text{ (in bits)}, \ w_f \text{ (in cycles/bit)}, \tau \text{ (in seconds)}\right\}$. In particular, for each task $f \in \mathcal{F}$, I_f^D represents the size of the local input data which is generated at the mobile device. I_f^S represents the size of the remote input data which is originated from the Internet and can be proactively stored. O_f represents the size of the output data. w_f and τ denote the required computation cycles per bit and the maximum tolerable service latency, respectively. Since the input remote data is generated proactively, the task popularity can be learned based on the request history information. The task request process at the mobile device is assumed to conform to the independent reference model (IRM) based on the following assumptions: i) the tasks that the mobile device wants to request is fixed to the set $\mathcal{F}$; ii) the probability of the request for task f at the mobile device, denoted as p_f, is constant and $\sum_{f=1}^{F} p_f = 1$ [28].

For clarity, take the online RPG as an example. The task space $\mathcal{F}$ is constructed according to the map information, each of which corresponds to the remote input data with size I_f^S. The history of all the players' actions could provide a popularity distribution of the map preferences, i.e., $(p_f)_{f\in\mathcal{F}}$. The player's pose information at the mobile device corresponds

to the local input data, the size of which is I_f^D. Even though corresponding to each map information, the local input data may vary from time to time, but the size of the pose information is in general unchanged. The rendered view obtained from combining the player's information with the map scenario information corresponds to the output data, the size of which is O_f. The overall latency corresponds to τ.

2.2.2 Caching and Computing Model

First, consider the cache placement at the mobile device. From the above-mentioned task model, we can see that only caching for the remote input data can be considered.[1] Denote with $c_f \in \{0, 1\}$ the caching decision of task $f \in \mathcal{F}$, where $c_f = 1$ means that the remote input data is cached at the mobile device and $c_f = 0$, otherwise. Denote with C (in bits) the size of the cache space at the mobile device and the caching constraint is given by

$$\sum_{f=1}^{F} I_f^S c_f \leq C. \tag{2.1}$$

All the remote input data are assumed to be proactively cached at the AP considering the storage size at the AP is generally large enough.

Next, consider the computing decision at the mobile device. Denote with $d_f \in \{0, 1\}$ the computing decision of task $f \in \mathcal{F}$, where $d_f = 1$ means that task f is computed at the mobile device and $d_f = 0$ means that task f is computed at the AP. Denote with f_D (in cycles/second) the computation frequency of the mobile device and f_S (in cycles/second) the computation frequency of the AP. The energy consumed for computing one cycle with frequency f_D at the mobile device is μf_D^2, where μ is the effective switched capacitance related to the chip architecture and can indicate the power efficiency of CPU at the mobile device [29]. Denote with $\bar{P}$ in (W) the average available power at the mobile device. Considering the AP is in general connected to a power grid, we assume that there is no power constraint at the AP.

[1]Since the local input data may be generated in real time and vary with time, we do not consider proactive computing either at the AP or the mobile device. Then, neither do we consider output data caching.

2.2.3 Service Mechanism

Based on the joint caching and computing decision $\left(\mathbf{c} \triangleq (c_f)_{f\in\mathcal{F}},\ \mathbf{d} \triangleq (d_f)_{f\in\mathcal{F}}\right)$, each task $f \in \mathcal{F}$ can be served via the following three service routes, each of which yields a unique wireless bandwidth requirement. Denote with $B_{f,j}$ the minimum wireless bandwidth (including both uplink and downlink) required for serving task f via service route j within the deadline τ.

- **Route 1: Local computing with local caching**. When $d_f = 1$ and $c_f = 1$, the mobile device immediately computes task f based on the locally generated input data and the locally cached remote input data. The required latency is the computation latency at the mobile device only, i.e., $\frac{\left(I_f^S+I_f^D\right)w_f}{f_D}d_f$. For satisfying the latency constraint, we assume that the computation at the mobile device can be completed by the deadline, i.e., $\frac{\left(I_f^S+I_f^D\right)w_f}{f_D} < \tau$. Since there is no need of transmission, $B_{f,1} = 0$ without loss of optimality. The energy consumption at the mobile device for task f is the computation energy only, i.e., $\mu f_D^2 w_f \left(I_f^D + I_f^S\right) d_f$.

- **Route 2: Local computing without local caching**. When $d_f = 1$ and $c_f = 0$, the mobile device first downloads the remote input data from the AP and then computes the task locally. The required latency includes the downlink transmission latency and the local computation latency, i.e.,

$$\Big(\frac{I_f^S}{B_f^D log(1+\frac{P_D h^2}{N_0})} + \frac{(I_f^S + I_f^D)w_f}{f_D}\Big)d_f\,(1-c_f) \le \tau, \tag{2.2}$$

  where B_f^D is the downlink bandwidth allocated for the transmission of task f, P_D is the average downlink power spectrum density (PSD) at the AP, h is the channel coefficient and N_0 is the average PSD of the channel noise. In order to satisfy the latency constraint (2.2), the minimum required bandwidth $B_{f,2} = B_f^D = \frac{I_f^S}{\left(\tau - \frac{(I_f^S+I_f^D)w_f}{f_D}\right)\log\left(1+\frac{P_D h^2}{N_0}\right)}$, where B_f^D achives the equality of (2.2). The energy consumption at the mobile device for task f is the computation energy only, i.e., $\mu f_D^2 w_f \left(I_f^D + I_f^S\right) d_f$.

- **Route 3: MEC computing**. When $d_f = 0$, the mobile device first uploads the local input data to the AP. After receiving the data transmitted from the mobile device, the AP computes task f based on both the local and remote input data from its storage and then transmits the output data back to the mobile device. The required latency includes the uplink transmission latency, the computation latency at the AP, and the downlink transmission latency, i.e.,

$$\Big(\frac{I_f^D}{B_f^U log(1+\frac{P_U h^2}{N_0})}+\frac{(I_f^S+I_f^D)w_f}{f_S}+\frac{O_f}{B_f^D log(1+\frac{P_D h^2}{N_0})}\Big)(1-d_f)\leq\tau, \tag{2.3}$$

where B_f^U is the uplink bandwidth allocated to the mobile device and P_U is the average uplink PSD at the mobile device. For satisfying the latency constraint, we assume that the computation at the AP can be completed before the deadline, i.e., $\frac{(I_f^S+I_f^D)w_f}{f_S} < \tau$. In this case, the minimum wireless bandwidth includes both uplink and downlink bandwidth and is given in Property 1.

Property 1 *When* $d_f = 0,\ c_f = 0,$ *i.e.,* $x_{f,3} = 1,\ B_{f,3} = \frac{(\sqrt{a_1}+\sqrt{a_2})^2}{a_3}$, *where* $a_1 = \frac{I_f^D}{\log\left(1+\frac{P_U h^2}{N_0}\right)},\ a_2 = \frac{O_f}{\log\left(1+\frac{P_D h^2}{N_0}\right)}$ *and* $a_3 = \tau - \frac{(I_f^S+I_f^D)w_f}{f_S}$.

Proof 1 *Proof of Property 1 can be seen in Appendix A.*

The energy consumption at the mobile device for task f is the uplink transmission energy only, i.e., $\frac{P_U I_f^D}{\log\left(1+\frac{P_U h^2}{N_0}\right)}(1-d_f)$.

From above, denote with $\mathbf{x} \triangleq (x_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}} \in \{0,1\}^{F\times 3}$ the service mechanism, where $x_{f,j} = 1$ indicates that task f is served via route j, and $x_{f,j} = 0$, otherwise. The first service mechanism, i.e., $x_{f,1} = 1$, corresponds to the local computing with local caching, i.e., $d_f = 1,\ c_f = 1$. The second service mechanism, i.e., $x_{f,2} = 1$, corresponds to the local computing without caching, i.e., $d_f = 1,\ c_f = 0$. The third service mechanism, i.e., $x_{f,3} = 1$,

corresponds to the MEC computing, i.e., $d_f = 0,\ c_f = 0$. The cache size constraint (2.1) can be accordingly rewritten as

$$\sum_{f=1}^{F} I_f^S x_{f,1} \leq C. \tag{2.4}$$

Then, in order to guarantee that each task can be get served, we have

$$\sum_{j=1}^{3} x_{f,j} = 1,\ f \in \mathcal{F}. \tag{2.5}$$

The average power consumption $\bar{P}$ [2] is given by

$$\begin{aligned}
\bar{P} &= \lim_{N\to\infty} \frac{1}{N\tau} \sum_{t=0}^{N-1} E_t \\
&\overset{(a)}{=} \sum_{f=1}^{F} p_f \sum_{j=1}^{3} \frac{E_{f,j}}{\tau} x_{f,j} \\
&= \sum_{f=1}^{F} p_f \Big(\frac{\mu f_D^2 w_f \left(I_f^D + I_f^S\right)}{\tau} (x_{f,1} + x_{f,2}) + \frac{P_U I_f^D}{\tau \log\left(1 + \frac{P_U h^2}{N_0}\right)} x_{f,3} \Big),
\end{aligned}$$

where N denotes the number of time slots, E_t denotes the energy consumption of the mobile device at time slot t, equation (a) holds based on the fact that the task request process conforms to the IRM, and $E_{f,j}$ denotes the energy consumption of the mobile device when task f is served via mechanism j. Under the average power budget constraint, we have

$$\sum_{f=1}^{F} p_f \Big(\frac{\mu f_D^2 w_f \left(I_f^D + I_f^S\right)}{\tau} (x_{f,1} + x_{f,2}) + \frac{P_U I_f^D}{\tau \log\left(1 + \frac{P_U h^2}{N_0}\right)} x_{f,3} \Big) \leq \bar{P}. \tag{2.6}$$

Similarly, the average consumed wireless bandwidth, including both uplink and downlink, is given by

$$\sum_{f=1}^{F} \sum_{j=1}^{3} p_f B_{f,j} x_{f,j}. \tag{2.7}$$

[2]The long-term average power consumption constraint can help make sure that the lifetime of the mobile device battery can last longer and elongate the duration for the mobile user to experience the application [30].

Table 2.2: Service Mechanisms and Corresponding Resource Consumption

Service Method	$d_f = 1, c_f = 1$, i.e., $x_{f,1} = 1$	$d_f = 1, c_f = 0$, i.e., $x_{f,2} = 1$	$d_f = 0$, i.e., $x_{f,3} = 1$
Energy cost	$\mu_f f_D^2 w_f (I_f^D + I_f^S)$	$\mu_f f_D^2 w_f (I_f^D + I_f^S)$	$\frac{P_U I_f^D}{\log\left(1+\frac{P_U h^2}{N_0^2}\right)}$
Bandwidth cost	0	$B_{f,2} = \frac{I_f^S}{\left(\tau - \frac{(I_f^S + I_f^D) w_f}{f_D}\right) \log\left(1+\frac{P_D h^2}{N_0}\right)}$	$B_{f,3} = \frac{(\sqrt{a_1}+\sqrt{a_2})^2}{a_3}$
Cache cost	I_f^S	0	0

For clarity, the above-mentioned service mechanisms and the corresponding resource consumption are summarized in Table II. As shown in Table II, compared to route 1, route 2 saves cache cost but consumes more transmission bandwidth (from 0 to $B_{f,2}$). Compared with route 1 and route 2 (i.e., local computation), route 3 does not consume local computation power but transmission power, and the bandwidth cost is different (from $B_{f,2}$ to $B_{f,3}$). The choice among route 1, route 2 and route 3 is affected by interplay among the local transmission power, local computation power, cache cost, and respective bandwidth cost. From above, the joint computing and caching policy requires careful design.

Remark 1 (Practical implementation) *Firstly, the joint caching and computing decisions* $(\boldsymbol{c}, \boldsymbol{d})$ *are made offline based on a-priori knowledge of system parameters. Note that even though the content of local input data might not be always known ahead of time, the size of local input data is assumed to be known in advance and thus the joint policy can still be made offline without loss of optimality. Next, the obtained remote input data caching decision is implemented when the traffic load is relatively low (e.g., nighttime) and then the cache state at the mobile device remains unchanged. In this way, the transmission cost of caching is negligible from the long term perspective [12]. The obtained computing decision table is assumed to be stored at the mobile device. When a user starts using the mobile device, at each time slot, the task request gets satisfied based on the cache state at the mobile device*

and the computing decision table stored in the mobile device.

2.3 Problem Formulation and Complexity Analysis

2.3.1 Problem Formulation

We formulate the joint caching and computing optimization problem to minimize the average required bandwidth, including both the uplink and downlink bandwidth, subject to the cache size, average power and latency constraints, as shown below.

Problem 1 (Joint Caching and Computing Optimization)

$$\begin{aligned} \min_{(x_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}}} \quad & \sum_{f=1}^{F}\sum_{j=1}^{3} p_f B_{f,j} x_{f,j} \\ s.t. \quad & (2.4), (2.5), (2.6), \\ & x_{f,j} \in \{0,1\}, j \in \{1,2,3\}, f \in \mathcal{F}. \end{aligned}$$

Denote with $\mathbf{x}^* \triangleq (x^*_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}}$ the optimal solution to Problem 1, $\left(\mathbf{c}^* \triangleq (c^*_f)_{f\in\mathcal{F}}, \mathbf{d}^* \triangleq (d^*_f)_{f\in\mathcal{F}}\right)$ the corresponding optimal joint caching and computing policy and B^* the corresponding optimal average bandwidth. Note that $\mathbf{c}^*$ can be directly determined from $\mathbf{x}^*$ according to Table II.

2.3.2 Computation Complexity

In the following, we show that Problem 1 is NP-hard in strong sense via transforming it into an MMKP without loss of equivalence, as shown below.

Problem 2 (Equivalent MMKP Problem)

$$\begin{aligned}
\max_{(x_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}}} \quad & \sum_{f}^{F}\sum_{j=1}^{3} v_{f,j}x_{f,j} \\
s.t. \quad & \sum_{f=1}^{F}\sum_{j=1}^{3} w_{f,j}^{k}x_{f,j} \le R^{k},\ k\in\{1,2\}, && (2.8)\\
& \sum_{j=1}^{3} x_{f,j} = 1,\ f\in\mathcal{F}, && (2.9)\\
& x_{f,j}\in\{0,1\},\ f\in\mathcal{F}, j\in\{1,2,3\}, && (2.10)
\end{aligned}$$

where

$$a_{f,j} \triangleq \begin{cases} -p_f B_{f,1}, & j=1,\\ -p_f B_{f,2}, & j=2,\\ -p_f B_{f,3}, & j=3, \end{cases} \tag{2.11}$$

$v_{f,j} \triangleq a_{f,j} + \max\limits_{\{f\in\mathcal{F},j\in\{1,2,3\}\}} |a_{f,j}|$, and $|\cdot|$ denotes the absolute value,[3]

$$w_{f,j}^{1} \triangleq \begin{cases} I_f^S, & j=1,\\ 0, & j=2,\\ 0, & j=3, \end{cases} \tag{2.12}$$

denotes the cache size cost of task f under the j-th choice, and

$$w_{f,j}^{2} \triangleq \begin{cases} p_f \frac{\mu f_D^2 w_f \left(I_f^D + I_f^S\right)}{\tau}, & j=1,\\ p_f \frac{\mu f_D^2 w_f \left(I_f^D + I_f^S\right)}{\tau}, & j=2,\\ p_f \frac{P_U I_f^D}{\tau \log\left(1+\frac{P_U h^2}{N_0}\right)}, & j=3, \end{cases} \tag{2.13}$$

denotes the power cost of task f under the j-th choice, $R^1 \triangleq C$ and $R^2 \triangleq \bar{P}$.

[3]It is obvious that the optimal policy does not change if we use $v_{f,j}$ instead of $a_{f,j}$ which corresponds to the original problem. But the reason we do such coefficient transformation is that the coefficients have to be positive for the Lagrangian relaxation algorithm of the MMKP to work [31].

We can see that Problem 2 is a 3-choice 2-dimensional knapsack problem. In particular, for each task $f \in \mathcal{F}$, the 3-choice corresponds to the above mentioned three service mechanisms and the 2-dimension corresponds to the cache size and average power constraints, respectively. Since MMKP is NP-hard in strong sense, we can conclude that Problem 2 is NP-hard in strong sense, and so is Problem 1. In addition, the computation complexity of the MMKP problem is $\mathcal{O}(3^F)$, which grows exponentially as the number of tasks F increases. Hence, the brute-force search method is not practical and we have to design a low-complexity high-performance algorithm.

2.4 Optimization in Homogeneous Scenario

In this section, for ease of insight illustration, we study the homogeneous scenario in which each task is assumed to be of the same input data size, output data size, computation load as well as request probability. The analytic expression for the optimal policy is derived and the 3C tradeoff is analyzed. The numerical results of the homogeneous scenario are implemented to exemplify the optimal policy and its properties.

2.4.1 Homogeneous Problem Formulation

We consider the homogeneous scenario, i.e., for each task $f \in \mathcal{F}$, $(B_{f,1}, B_{f,2}, B_{f,3}, I_f^S, I_f^D, O_f, w_f, \tau) = (B_1, B_2, B_3, I^S, I^D, O, w, \tau)$ and $p_f = \frac{1}{F}$. In such a scenario, Problem 2 can be reformulated as below without loss of equivalence.

Problem 3 (Optimization in Homogeneous Scenario)

$$
\begin{aligned}
\min_{(x_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}}} \quad & \frac{1}{F}\sum_{f=1}^{F}\sum_{j=1}^{3} B_j x_{f,j} \\
s.t. \quad & I^S \sum_{f=1}^{F} x_{f,1} \le C, && (2.14)\\
& \frac{\mu f_D^2 w\left(I^D+I^S\right)}{\tau F}\sum_{f=1}^{F}(x_{f,1}+x_{f,2})+ \\
& \frac{P_U I^D}{F\tau \log\left(1+\frac{P_U h^2}{N_0}\right)}\sum_{f=1}^{F} x_{f,3} \le \bar{P}, && (2.15)\\
& \sum_{j=1}^{3} x_{f,j} = 1, f\in\mathcal{F}, && (2.16)\\
& x_{f,j}\in\{0,1\}, j\in\{1,2,3\}, f\in\mathcal{F}.
\end{aligned}
$$

Denote with $X_j \triangleq \sum_{f=1}^{F} x_{f,j},\ j\in\{1,2,3\}$ the number of tasks served via the j-th route. Since each task is independent of each other and homogeneous, given $(X_j)_{j\in\{1,2,3\}}$, $(x_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}}$ can be obtained via

$$
x_{f,1} = \begin{cases} 1 & f = 1,\cdots,X_1, \\ 0 & \text{otherwise}, \end{cases} \tag{2.17}
$$

$$
x_{f,2} = \begin{cases} 1 & f = X_1+1,\cdots,X_1+X_2, \\ 0 & \text{otherwise}, \end{cases} \tag{2.18}
$$

$$
x_{f,3} = \begin{cases} 1 & f = X_1+X_2+1,\cdots,X_1+X_2+X_3, \\ 0 & \text{otherwise}. \end{cases} \tag{2.19}
$$

Via replacing $(x_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}}$ with $(X_j)_{j\in\{1,2,3\}}$, Problem 3 can be transformed into Problem 4 without loss of equivalence.

Problem 4 (Equivalent Optimization in Homogeneous Scenario)

$$\min_{(X_j)_{j\in\{1,2,3\}}} X_1B_1 + X_2B_2 + X_3B_3 \tag{2.20}$$

$$s.t. \qquad I^S X_1 \le C, \tag{2.21}$$

$$k_1(X_1 + X_2) + k_2X_3 \le \bar{P}, \tag{2.22}$$

$$X_1 + X_2 + X_3 = F, \tag{2.23}$$

$$0 \le X_1 \le F, \tag{2.24}$$

$$0 \le X_2 \le F, \tag{2.25}$$

$$0 \le X_3 \le F, \tag{2.26}$$

where $k_1 \triangleq \frac{\mu f_D^2 w\left(I^D+I^S\right)}{\tau F}$ *and* $k_2 \triangleq \frac{P_U I^D}{F\tau \log\left(1+\frac{P_U h^2}{N_0}\right)}$ *represent the average power consumed at the mobile device of each task for local computing and uplink transmission, respectively.*

From Problem 4, we can see that the optimal policy $(X_j)_{j\in\{1,2,3\}}$ is determined by the interplay among B_1, B_2 and B_3 as well as the relationship between the local computation power cost k_1 and uplink transmission power cost k_2.

2.4.2 Optimal Policy

Theorem 1 *(Optimal joint policy when $k_1 > k_2$) If $B_3 > B_2$, the optimal joint policy is given as*

$$\begin{aligned} X_1 &= \min\left\{\left\lfloor\frac{C}{I^S}\right\rfloor, F, \left\lfloor\frac{\bar{P}-Fk_2}{k_1-k_2}\right\rfloor\right\}, \\ X_2 &= \max\left\{0, \min\left\{F, \left\lfloor\frac{\bar{P}-Fk_2}{k_1-k_2}\right\rfloor\right\} - X_1\right\}, \\ X_3 &= F - X_1 - X_2, \end{aligned} \tag{2.27}$$

where $\lfloor \bullet \rfloor$ denotes the round-down function. The minimum bandwidth is $B_2X_2 + B_3X_3$. If $B_3 \leq B_2$, the optimal joint policy is given as

$$\begin{aligned} X_1 &= \min\left\{ \left\lfloor \frac{C}{I^S} \right\rfloor, F, \left\lfloor \frac{\bar{P} - Fk_2}{k_1 - k_2} \right\rfloor \right\}, \\ X_2 &= 0, \\ X_3 &= F - X_1. \end{aligned} \tag{2.28}$$

The minimum bandwidth is B_3X_3.

Proof 2 *[32]*

Theorem 2 *(Optimal joint policy when $k_1 \leq k_2$) If $B_3 > B_2$, the optimal joint policy is given as*

$$\begin{aligned} X_1 &= \left\lfloor \frac{C}{I^S} \right\rfloor, \\ X_2 &= F - X_1, \\ X_3 &= 0. \end{aligned} \tag{2.29}$$

The minimum bandwidth is B_2X_2. If $B_3 \leq B_2$, the optimal joint policy is given as

$$\begin{aligned} X_1 &= \left\lfloor \frac{C}{I^S} \right\rfloor, \\ X_2 &= \max\left\{ 0, \left\lceil \frac{\bar{P} - Fk_2}{k_1 - k_2} \right\rceil - X_1 \right\}, \\ X_3 &= F - X_1 - X_2. \end{aligned} \tag{2.30}$$

The minimum bandwidth is given as $B_2X_2 + B_3X_3$.

Proof 3 *[32]*

2.5 Optimization in Heterogeneous Scenario

In this section, we consider the heterogeneous scenario for the system model, i.e., all the tasks are of different data sizes, computation loads and request probabilities. The MMKP problem is firstly solved by the LR method. Based on the LR solution, we formulate an

auxiliary problem and solve it via CCCP. Next, we propose the LR+CCCP method and obtains a feasible local optimal solution to the original problem. In the end, we compare our proposed LR-based methods with the CCCP and the greedy algorithm to present the advantage of our method in both bandwidth performance and time efficiency.

2.5.1 Lagrangian Relaxation Method

In this subsection, we obtain a suboptimal solution to Problem 2 based on LR method [31].[4] Firstly, the LR problem of Problem 2 is formulated via introducing the Lagrangian multipliers and then relaxing the constraint (2.8) into the objective value of Problem 2, while keeping the constraints in (2.9) and (2.10) active as below.

Problem 5 (Lagrangian Relaxation Problem)

$$B_{LR}(\boldsymbol{\lambda}) \triangleq \max_{(x_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}}} \sum_{f}^{F}\sum_{j=1}^{3} v_{f,j}(\boldsymbol{\lambda})x_{f,j} + \sum_{k=1}^{2}\lambda_k R^k$$
$$s.t. \qquad (2.9), (2.10),$$

where $v_{f,j}(\boldsymbol{\lambda}) \triangleq v_{f,j} - \sum_{k=1}^{2}\lambda_k w_{f,j}^k$ *and* $\boldsymbol{\lambda} \triangleq (\lambda_1, \lambda_2)$ *are the introduced Lagrangian multipliers.*

Given $\boldsymbol{\lambda}$, denote with $\mathbf{x}^*(\boldsymbol{\lambda}) \triangleq \left(x_{f,j}^*(\boldsymbol{\lambda})\right)_{f\in\mathcal{F},j\in\{1,2,3\}}$ the optimal solution to Problem 5 and $B_{LR}(\boldsymbol{\lambda})$ the maximal objective value of Problem 5. Obviously, $\mathbf{x}^*(\boldsymbol{\lambda})$ can be obtained as below. For each $f \in \mathcal{F}$,

$$x_{f,j}^*(\boldsymbol{\lambda}) \triangleq \begin{cases} 1, & j = \arg\max_{j'\in\{1,2,3\}} v_{f,j'}(\boldsymbol{\lambda}), \\ 0, & \text{otherwise.} \end{cases} \tag{2.31}$$

That is, for each $f \in \mathcal{F}$, the service mechanism with the maximum value of $v_{f,j}(\boldsymbol{\lambda})$ is chosen. In addition, we can observe that the optimal objective value of Problem 5, i.e., $B_{LR}(\boldsymbol{\lambda})$, is an upper bound of the optimal value of the original problem B^*, i.e., $B_{LR}(\boldsymbol{\lambda}) \geq B^*$.

Then, we try to obtain a tighter upper bound of B^* via minimizing $B_{LR}(\boldsymbol{\lambda})$. That is,

[4] It is noteworthy that the prerequisite of LR method is that any elements of $v_{f,j}$ must be non-negative [31], and that explains the transformation from $a_{f,j}$ to $v_{f,j}$.

Problem 6 (Lagrangian Dual Problem)

$$B_{LR}(\boldsymbol{\lambda}^*) \triangleq \min_{\boldsymbol{\lambda} \in \boldsymbol{R}_+^2} B_{LR}(\boldsymbol{\lambda}),$$

where λ^* is the optimal solution to Problem 6.

The optimal Lagrangian multiplier $\boldsymbol{\lambda}^*$ can be obtained via solving Problem 6 with subgradient method. In particular, given an initial value $\boldsymbol{\lambda}^0 \triangleq (\lambda_1^0, \lambda_2^0)$, a sequence $\{\boldsymbol{\lambda}^t\}$ can be iteratively determined by the following formula:

$$\lambda_k^{t+1} = \max\left\{\lambda_k^t - q_t g_k^t, 0\right\},\ k \in \{1,2\}, t \in \{1, \cdots, T\}, \tag{2.32}$$

where $g_k^t \triangleq R^k - \sum_{f=1}^{F}\sum_{j=1}^{3} w_{f,j}^k x_{f,j}^*(\lambda_k^t)$ denotes the subgradient of $B_{LR}(\boldsymbol{\lambda})$ at λ_k^t, q_t is a positive scalar step size and can be computed according to

$$q^t = \frac{\epsilon_t (B_{LR}(\boldsymbol{\lambda}^t) - B_{LR}^{'}(\boldsymbol{\lambda}^t))}{||\mathbf{g}^t||^2}, \tag{2.33}$$

where ϵ_t is a constant such that $0 \leq \epsilon_t \leq 2$, $t \in \{1, \cdots, T\}$, $B_{LR}^{'}(\boldsymbol{\lambda}^t)$ is a lower bound on $B_{LR}(\boldsymbol{\lambda}^t)$ and $||\mathbf{g}^t||$ is the l^2 norm of $\mathbf{g}^t \triangleq (g_1^t, g_2^t)$. In this way, $\boldsymbol{\lambda}^t$ is guaranteed to converge to the optimal value $\boldsymbol{\lambda}^*$ of Problem 5, i.e., $\lim_{t\to\infty} \boldsymbol{\lambda}^t = \boldsymbol{\lambda}^*$ [31].

Denote with $\check{\boldsymbol{\lambda}} \triangleq (\check{\lambda}_1, \check{\lambda}_2)$ the converged value of the Lagrangian multipliers. Given $\check{\boldsymbol{\lambda}}$, the final policy, named as LR solution and denoted as $\check{\mathbf{x}} \triangleq (\check{x}_{f,j})_{f\in\mathcal{F}, j\in\{1,2,3\}}$, is obtained from Equation (2.31).

2.5.2 Auxiliary Problem and the LR+CCCP Method

The solution obtained from the LR method $\check{\mathbf{x}}$ may not be local optimal or feasible for the original problem, i.e., Problem 1. In what follows, by combining CCCP and LR, we obtain a feasible local optimal solution to Problem 1 via formulating an auxiliary problem.

Firstly, given the LR solution $\check{\mathbf{x}}$, we establish an auxiliary problem of Problem 2 using local neighbourhood search (LNS) as follows [33]:

Problem 7 *[Auxiliary Problem] Given the LR solution* $\check{\boldsymbol{x}}$*,*

$$\max \sum_{f=1}^{F}\sum_{j=1}^{3} v'_{f,j}x_{f,j}$$

$$s.t. \sum_{f=1}^{F}\sum_{j=1}^{3} w'^{k}_{f,j}x_{f,j} \le R'^{k}, k \in \{1,2\}, \tag{2.34}$$

$$\sum_{j=1}^{3} x_{f,j} \le 1,\ f \in \{1,\ldots,F\}, \tag{2.35}$$

$$x_{f,j} \in \{0,1\},\ f \in \{1,\ldots,F\}, j \in \{1,2,3\}, \tag{2.36}$$

where for each $f \in \{1,\ldots,F\},\ j \in \{1,2,3\},\ k \in \{1,2\}$*,*

$$v'_{f,j} \triangleq v_{f,j} - v_{f,\check{j}(f)},$$
$$w'^{k}_{f,j} \triangleq w^{k}_{f,j} - w^{k}_{f,\check{j}(f)},$$
$$R'^{k} = R^{k} - \sum_{f=1}^{F} w^{k}_{f,\check{j}(f)},$$

with $\check{j}(f) \triangleq \arg\max_{j\in\{1,2,3\}} \check{x}_{f,j}$.

For example, if $F = 2,\ \check{x}_{1,1} = 0,\ \check{x}_{1,2} = 1,\ \check{x}_{1,3} = 0,\ \check{x}_{2,1} = 1,\ \check{x}_{2,2} = 0$ and $\check{x}_{2,3} = 0$, we have $j(1) = 2$, $j(2) = 1$, $v'_{1,j} = v_{1,j} - v_{1,2}$, $v'_{2,j} = v_{2,j} - v_{2,1}$, $w'^{k}_{1,j} = w^{k}_{1,j} - w^{k}_{1,2}$, $w'^{k}_{2,j} = w^{k}_{2,j} - w^{k}_{2,1}$ for $j \in \{1,2,3\}$, and $R'^{k} = R^{k} - w^{k}_{1,2} - w^{k}_{2,1},\ k \in \{1,2\}$.

Denote with $\hat{\mathbf{x}} \triangleq (\hat{x}_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}}$ a feasible solution to the auxiliary problem, i.e., Problem 7. Note that even though the auxiliary problem is relatively less complex than the original MMKP via relaxing the equality constraint (2.9) to the inequality constraint (2.35), the computation complexity of obtaining the optimal solution to the auxiliary problem via brute force search is still $\mathcal{O}(3^F)$, which increases with F exponentially. Hence, the brute-force search-based optimal solution is not practical.

Then, based on the LR solution, we utilize CCCP to obtain a local optimum to the auxiliary problem, i.e., Problem 7, while guaranteeing the feasibility of the solution unless the original problem is infeasible naturally, as shown below.

- Rewrite the discrete constraint in (2.36) as the following constraints without loss of equivalence, i.e.,

$$x_{f,j} \in [0,1],\ f \in \mathcal{F},\ j \in \{1,2,3\}, \tag{2.37}$$

$$\sum_{f=1}^{F}\sum_{j=1}^{3} x_{f,j}(1-x_{f,j}) \leq 0. \tag{2.38}$$

- Transform Problem 7 into the following problem by penalizing the concave constraint (2.38) to the objective function, i.e.,

Problem 8 (Penalized Joint Policy Optimization)

$$\begin{aligned} \min\ & \sum_{f=1}^{F}\sum_{j=1}^{3} -v'_{f,j}x_{f,j} - \alpha\sum_{f=1}^{F}\sum_{j=1}^{3} x_{f,j}(x_{f,j}-1) \\ s.t.\quad & (2.34),\ (2.35),\ (2.37), \end{aligned} \tag{2.39}$$

which is proven to be equivalent to Problem 7 with $\alpha \geq \alpha_0$ [34], where

$$\alpha_0 \triangleq \frac{\sum_{f=1}^{F}\sum_{j=1}^{3} -v_{f,j}x^0_{f,j} - \gamma}{\max_{\mathbf{x}}\{\sum_{f=1}^{F}\sum_{j=1}^{3} x_{f,j}(x_{f,j}-1) : (2.34),(2.35),(2.37)\}}, \tag{2.40}$$

with $(x^0_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}}$ as an initial solution satisfying (2.34),(2.35),(2.37), and $\gamma = \min_{\mathbf{x}}\{\sum_{f=1}^{F}\sum_{j=1}^{3} -v'_{f,j}x_{f,j} : (2.34),(2.35),(2.37)\}$.

- Solve Problem 8 with the CCCP method based on the fact that Problem 8 is a difference of convex problem since its objective function is a difference of a linear function, i.e., $\sum_{f=1}^{F}\sum_{j=1}^{3} -v'_{f,j}x_{f,j}$, and a convex function, i.e., $\alpha\sum_{f=1}^{F}\sum_{j=1}^{3} x_{f,j}(x_{f,j}-1)$, and its constraints (2.34), (2.35), (2.37) are linear [35]. In particular, CCCP involves iteratively solving a sequence of linear problems and a local optimum of the auxiliary problem can be obtained. The initial feasible point of the CCCP can be set as $x^0_{f,j} = 0,\ f \in \mathcal{F},\ j \in \{1,2,3\}$.

The previous method is the proposed LR+CCCP method. Now, we show how the solution to the auxiliary problem $\hat{\mathbf{x}}$ improves the Lagrangian solution $\check{\mathbf{x}}$. The LR+CCCP solution to the original problem, denoted as $\bar{\mathbf{x}} \triangleq (\bar{x}_{f,j})_{f\in\mathcal{F},j\in\{1,2,3\}}$, is constructed as below:
For each $f \in \{1,\cdots,F\}$ *and* $j \in \{1,2,3\}$, $\bar{x}_{f,j} = 1$ *if one of the following two conditions is satisfied: (i)* $\hat{x}_{f,j} = 1$ *or (ii)* $\check{x}_{f,j} = 1$ *and* $\hat{x}_{f,j'} = 0, \forall j' \in \{1,2,3\}$. $\bar{x}_{f,j} = 0$, *otherwise.*

Condition (i) indicates that the exchange between $\check{\mathbf{x}}$ and $\hat{\mathbf{x}}$ happens for task f while condition (ii) indicates that no exchange is realized. The overall LR+CCCP method is shown in Algorithm 1. Next, we analyze the feasibility, optimality and the computation complexity of the proposed LR+CCCP method. The following two lemmas in [33] guarantee the feasibility and optimality of the LR+CCCP method.

Lemma 1 *$\bar{\boldsymbol{x}}$ is a feasible solution to the original MMKP Problem 2 if $\hat{\boldsymbol{x}}$ is a feasible solution to the auxiliary problem Problem 7.*

Lemma 2 *$\bar{\boldsymbol{x}}$ is an optimal solution to the original MMKP Problem 2 if $\hat{\boldsymbol{x}}$ is an optimal solution to the auxiliary problem Problem 7.*

Since the CCCP method provides a feasible local optimum of the auxiliary problem, we can observe that the LR+CCCP method provides at least a feasible solution to the original problem with better performance compared with the LR method based on Lemma 1 and 2.

Remark 2 *The computation complexity of brute-force to the original problem is $\mathcal{O}(3^F)$. As for the LR solution $\check{\boldsymbol{x}}$ based on subgradient method, the computation complexity is $\mathcal{O}(F)$ from (2.31) and (2.32) at each iteration. Secondly, to obtain the CCCP solution $\hat{\boldsymbol{x}}$, the computation complexity is $\mathcal{O}(F^2)$ at each iteration.*

Algorithm 1: Lagrangian Relaxation (LR) + Concave-Convex Procedure (CCCP).

Input : $(v_{f,j}, w^k_{f,j}, R^k), f \in \mathcal{F}, j \in \{1,2,3\}, k \in \{1,2\}$

Output: $\bar{\mathbf{x}}$

$\check{x}_{f,j} = 0$, for all f and j, randomly initialize λ^0_1 and λ^0_2, and $t = 0$;

while $||\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^t||^2_2 <= \epsilon_1$ **do**

$$q_t = \frac{\epsilon_t(B_{LR}(\boldsymbol{\lambda}^t) - B'_{LR}(\boldsymbol{\lambda}^t))}{||\mathbf{g}^t||^2}, \tag{2.41}$$

$$\lambda^{t+1}_k = \max\left\{\lambda^t_k - q_t g^t_k, 0\right\}, \tag{2.42}$$

$$\check{x}_{f,j}(\lambda) \triangleq \begin{cases} 1, & j = \arg\max_{j' \in \{1,2,3\}} v_{f,j'}(\lambda), \\ 0, & \text{otherwise.} \end{cases} \tag{2.43}$$

$t = t + 1$;

end

Formulate the auxiliary problem based on $\check{\mathbf{x}}$, $t = 0$, and set $\hat{x}$ to all zero;

while $O^{t+1} - O^t <= \epsilon_2$ **do**

$$\min_{\hat{x}^{t+1}} \sum_{f=1}^{F}\sum_{j=1}^{3} -v'_{f,j}\hat{x}^{t+1}_{f,j} - \alpha \sum_{f=1}^{F}\sum_{j=1}^{3}(2\hat{x}^t_{f,j} - 1)(\hat{x}^{t+1}_{f,j} - \hat{x}^t_{f,j})$$

$$s.t. \quad (2.34),\ (2.35),\ (2.37),$$

$t = t + 1$

end

Denote the objective value as O^{t+1};

Construct $\bar{\mathbf{x}}$ based on $\hat{\mathbf{x}}$ and $\check{\mathbf{x}}$. If $\bar{\mathbf{x}}$ is infeasible, then conclude the infeasibility of original problem. Otherwise output $\bar{\mathbf{x}}$;

2.6 Simulation Result

In this section, we do simulation for the proposed LR+CCCP method and compare it with the heuristic greedy method shown in Algorithm 2, the LR method and the CCCP method with the initial point obtained from the greedy method. In particular, the system parameters are chosen as below unless otherwise mentioned. $\frac{P_U h^2}{N_0} = 19.4$ dB, $\frac{P_D h^2}{N_0} = 24.67$ dB [36], $f_S = 20$ GHz, $\bar{P} = 7.5$ W, $\tau = 0.05$ s, $\mu = 10^{-27}$, $w_f \sim U[2,3,4,5,6,7,8,9,10]$, $I_f^D \sim U[1,2,3] * 10^6$ bits, $I_f^S \sim U[1,2,3] * 10^6$ bits, $O_f^S = (I_f^S + I_f^D) * 0.75$, $F = 800$, $\epsilon_t = 0.5$ and $\alpha = 10^5$. Probability of each task p_f is given as $p_f = \frac{|p'_f|}{\sum_{f=1}^{F} |p'_f|}$, where p'_f is a random value from the normal distribution $\mathcal{N}(0, \sigma^2)$. The standard deviation σ is set as 1. The uplink transmit PSD $P_U = \frac{250 \text{ mW}}{180 \text{ KHz}}$, and the downlink transmit PSD $P_D = \frac{5 \text{ W}}{180 \text{ KHz}}$. According to 4G standard [37], resource block has bandwidth of 180 KHz. Transmission power of each RF chain is 250 mW [36].

The computation complexity of the optimal policy via brute-force is $\mathcal{O}(3^F)$. When F is large, the computation complexity of computing the optimal policy is not acceptable. Therefore, we choose small $F = 12$ in simulating the performance of the optimal policy via brute-force. Fig 2.3 illustrates the average bandwidth consumption versus the computation frequency f_D. We can see that the performance of our proposed LR+CCCP is very close to that of the optimal policy at small F. Moreover, we can see that at small F, our proposed LR+CCCP still outperforms the other three baselines.

Fig 2.4(a) illustrates the impact of C on B^*. We can see that the proposed LR+CCCP method outperforms the three baselines. Define the bandwidth gain of each *policy* as $\frac{B^*_{greedy} - B^*_{policy}}{B^*_{greedy}}$. At $C = 700$ Mbits, the bandwidth gains of the LR+CCCP method, the greedy algorithm, CCCP, as well as the LR method, are 29%, 0%, 21% and 24%, respectively. Fig 2.4(b) illustrates the impact of f_D on B^*. We can observe that at the beginning, as f_D increases, the average bandwidth consumption decreases. This is because when f_D is relatively small, increasing f_D mainly reduces the latency for computing at the mobile device and thus more tasks will be allocated for local computation. When f_D is relatively large, the

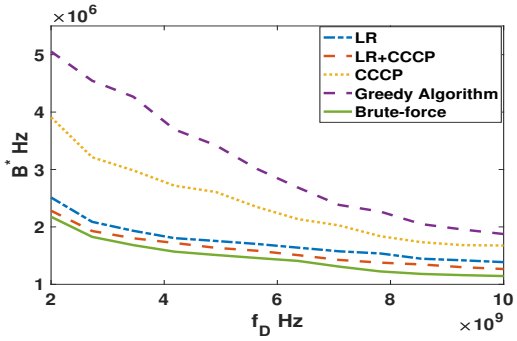
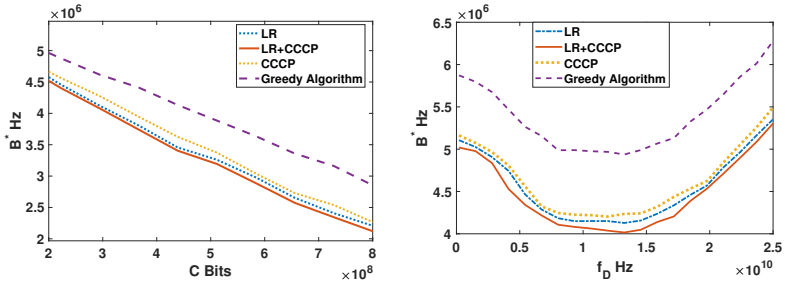


Figure 2.3: Impact of f_D on B^*, $C = 20$ Mbits, $\bar{P} = 7.5$ W, $\tau = 0.02$ s, $F = 12$, $\sigma = 1$.



(a) Impact of C on B^*. $f_D = 4$ GHz, $\bar{P} = 7.5$ W, $\tau = 0.025$ s, $F = 800$, $\sigma = 1$.

(b) Impact of f_D on B^*, $C = 400$ Mbits, $\bar{P} = 7.5$ W, $\tau = 0.025$ s, $F = 800$, $\sigma = 1$.

Figure 2.4: Impact of C and f_D on B^*

average bandwidth consumption increases. This is because increasing f_D mainly increases the energy consumption for local computation. Under the average power constraint, there would be less tasks allocated for local computation and more tasks allocated for MEC computing. Fig 2.5 compares the convergence time of LR+CCCP and the three baselines versus F. First, intuitively, we can see that the convergence rate of LR+CCCP is slower than the LR since it utilizes CCCP to guarantee the feasibility of the LR solution. Considering the bandwidth gain obtained from the LR+CCCP, we can observe that the increase of the computation complexity is acceptable. If running time does not allow, we can choose to use LR

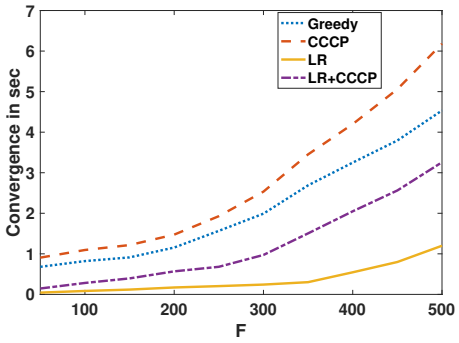
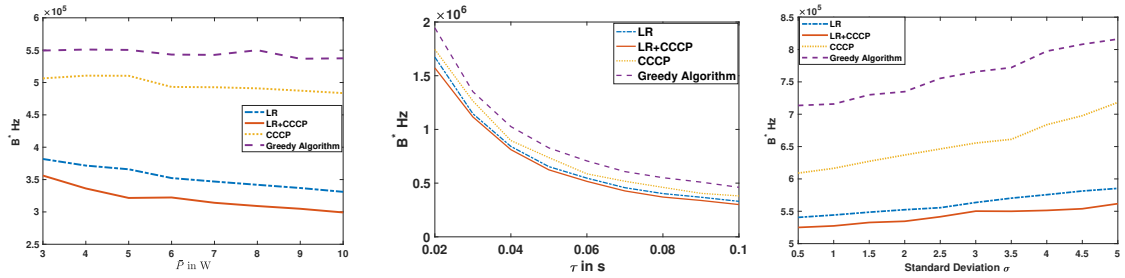


Figure 2.5: Convergence time. $f_D = 4$ GHz, $C = 400$ Mbits, $F \in [50, 500]$, $\bar{P} = 7.5$ W, $\tau = 0.025$ s, $\sigma = 1$.

instead. For the scenario when the LR solution is not feasible, we then use LR+CCCP to obtain a feasible local optimal solution. Next, the LR+CCCP method has faster convergence rate than the CCCP and the greedy algorithm. This indicates that the LR+CCCP method can be applied to a larger system than the CCCP or the greedy algorithm.



(a) Impact of $\bar{P}$ on B^*, $f_D = 4 \times 10^9$ Hz, $C = 300$ Mbits, $\tau = 0.05$ s, $F = 400$, $\sigma = 1$.

(b) Impact of τ on B^*, $f_D = 4 \times 10^9$ Hz, $C = 400$ Mbits, $\bar{P} = 7.5$ W, $F = 800$, $\sigma = 1$.

(c) Impact of σ on B^*, $f_D = 4$ G Hz, $C = 400$ Mbits, $\bar{P} = 7.5$ W, $\tau = 0.05$ s $F = 800$.

Figure 2.6: Impact of C and f_D on B^*

In Fig 2.6(a), the impact of the power constraint on the performance of all algorithm is demonstrated. As $\bar{P}$ increases, LR and LR+CCCP can utilize the relaxed power constraint

better than CCCP and the greedy algorithm to save more bandwidth. CCCP and greedy algorithm are insensitive to the relaxation of the power constraint. In Fig 2.6(b), the impact of time constraint on the performance of all algorithms is demonstrated. The bandwidth consumption decreases with τ since when task input/output data size and other system parameters are fixed, the time duration available for transmission increases with τ and thus the data rate decreases with τ. As τ increases, every algorithm is sensitive and LR+CCCP still outperforms all the other algorithms. Hence, LR and LR+CCCP are sensitive to both power and time constraints, which indicates the performance superiority of LR-based algorithms over the greedy algorithm and CCCP. Fig 2.6(c) illustrates the impact of the heterogeneity of file popularity, i.e., the standard deviation σ, on B^*. As the level of heterogeneity of the file popularity increases, the average consumed bandwidth B^* increases as well for every algorithm. Moreover, the increasing slops of the LR and LR+CCCP are lower than those of CCCP and greedy algorithm. This indicates that the entropy (degree of chaos) can worsen the performance of the policy, while the LR-based algorithms behave more robust.

2.6.1 Summary

We firstly consider a novel bidirectional computation task model, the input data of which consists of real-time local input data and proactively generated remote input data. We formulate the joint caching and computing policy optimization problem to minimize the average bandwidth consumption under the latency, cache size and average power constraints. In the homogeneous scenario, we derive the optimal policy and the analytical expression for the minimum bandwidth. In the heterogeneous scenario, we propose the LR-based methods and show that they outperform the state-of-the-art CCCP and the heuristic greedy algorithm from both the bandwidth performance and the time efficiency.

Appendix A: Proof of Property 4

For each task $f \in \mathcal{F}$, when $d_f = 0$, we have $\frac{I_f^D}{B_f^U \log(1+\frac{P_U h^2}{N_0})} + \frac{(I_f^S+I_f^D)w_f}{f_S} + \frac{O_f}{B_f^D \log(1+\frac{P_1 h^2}{N_0})} \leq \tau$. Hence, $B_{f,3}$ can be obtained via solving the following optimization problem:

$$\begin{aligned} \min_{B_f^U, B_f^D} \quad & B_f^U + B_f^D \\ s.t. \quad & \frac{a_1}{B_f^U} + \frac{a_2}{B_f^D} \leq a_3, \\ & B_f^U > 0, \\ & B_f^D > 0, \end{aligned} \tag{2.44}$$

where $a_1 = \frac{I_f^D}{\log(1+\frac{P_U h^2}{N_0})} > 0$, $a_2 = \frac{(I_f^S+I_f^D)w_f}{f_0} > 0$ and $a_3 = \tau - \frac{O_f}{\log(1+\frac{P_D h^2}{N_0})} > 0$. We can see that Problem 10 is a convex minimization problem. Denote with $B_f^{U^*}$ and $B_f^{D^*}$ the optimal solution to Problem 10. In order to solve Problem 10, let us first consider a modified version of the above convex problem as below.

$$\begin{aligned} \min_{B_f^U, B_f^D} \quad & B_f^U + B_f^D \\ s.t. \quad & \frac{a_1}{B_f^U} + \frac{a_2}{B_f^D} \leq a_3. \end{aligned} \tag{2.45}$$

If the solution to Problem 11 satisfies $B_f^U > 0$ and $B_f^D > 0$, then it is also a solution to Problem 10. Based on KKT conditions of Problem 11, we get an optimal solution to Problem 11 as below.

$$B_f^{U^*} = \frac{a_1 + \sqrt{a_1 a_2}}{a_3} > 0, \tag{2.46}$$

$$B_f^{D^*} = \frac{a_2 + \sqrt{a_1 a_2}}{a_3} > 0. \tag{2.47}$$

Therefore, we get $B_f^{U^*}$ and $B_f^{D^*}$ of Problem 10, and then $B_{f,3} = B_f^{U^*} + B_f^{D^*}$. The proof ends.

Chapter 3

MULTIPLEXING URLLC TRAFFIC WITHIN EMBB SERVICES IN 5G NR: FAIR SCHEDULING

3.1 Motivation

Emerging 5G networks are expected to support a variety of applications and services, notably the three main classes: (i) Enhanced mobile broadband (eMBB) services for longer duration flows requiring high network throughput; (ii) Ultra-reliable low-latency communication (URLLC) is to support mission critical applications such as real-time telemedicine and autonomous vehicles; and (iii) massive machine type communications (mMTC) within the Internet of Things (IoT) [38]. For URLLC communications, the most stringent service target is one millisecond (ms) system latency with 99.999% reliability [3]. Simultaneously, multimedia flows utilizing eMBB desire bandwidth resources to achieve transmission rates of up to 10 Gbps.

A key enabler is the emergence of *network slicing* concepts that allocate specific radio access network (RAN) functionality to each of the supported services within the same network infrastructure. In our work, RAN slicing is implemented at the packet level, where the scheduler decides the physical resource allocation [39]. The scheduling algorithms devised for long-term evolution (LTE) must be adapted to 5G scenarios to answer engineering challenges such as: how to integrate newly arriving URLLC users within an ongoing frame already allocated to eMBB users. Due to their delay constraints, URLLC users will need to be pre-emptively scheduled rather than wait for the next slot.

In this chapter, we leverage the *superposition/puncturing feature in 5G New Radio (NR) to multiplex these two types of traffic in 5G* [40], shown in Fig 3.1. This implies that more flexible resource management algorithms are required to fulfill the competing Quality of

service (QoS) requirements while maintaining fairness. The scheduling algorithms widely used in LTE networks - Round Robin (RR), Max Rate (MR) and Proportional Fair (PF) - are all designed to exploit the current channel conditions and maximize a time-averaged utility (e.g., sum rate). Other schedulers like modified largest weighted delay first (M-LWDF) [41] consider delay as part of the utility function.

A key degree of freedom in 5G for joint scheduling of eMBB and URLLC is the flexible numerology (μ) [5] that allows sub-carrier spacing to scale as $2^{\mu} \times 15$ kHz to provide a balance between different service requirements: QoS, latency and rate. Since the frame (sub-frame) durations are preserved - 10(1) ms, respectively - as in LTE, with an increase in the numerology index μ, the number of slots in a subframe increase, therefore increasing the number of symbols sent as shown in Fig 3.1. A standard slot has 14 orthogonal frequency-division multiplexing (OFDM) symbols. In contrast, mini-slots can contain 7, 4, or 2 OFDM symbols and also start immediately without needing to wait for slot boundaries, enabling quick delivery of low-latency payloads. While resource blocks (RB) over slots are allocated to each eMBB users, URLLC users are allocated smaller RBs using mini-slots. To satisfy delay and reliability constraints, resources originally allocated to eMBB users are re-allocated over mini-slot duration by 'puncturing', i.e., setting the transmission power of eMBB traffic to zero in the mini-slot. Alternatively, choosing non-zero transmission powers for both eMBB and URLLC traffic may be used to achieve superposition. In either case, the consequence is a reduction of eMBB user throughput. We investigate a scheduler design that maximizes the original eMBB users' aggregate utility while guaranteeing the QoS of URLLC users.

3.1.1 Related Work

Reducing end-to-end delay while achieving ultra-high reliability for URLLC continues to be a challenge over lossy networks, as it typically implies the need for retransmissions (or other

[1]There are two types of URLLC traffic to consider for an ongoing eMBB slot: 1) Strictly constrained URLLC traffic (less than 1 ms and 99.999% reliability) that requires puncturing current eMBB transmission; 2) Softly constrained URLLC traffic (e.g. 99.999% and 5 ms in V2X and 99.9% and 10 ms in VR/AR) [3] that can wait for the next empty eMBB slot.

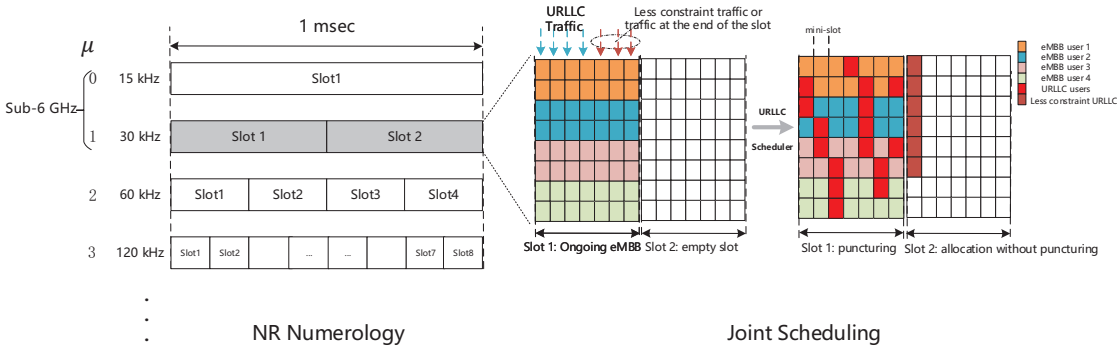


Figure 3.1: NR numerology, mini-slot and the superposition/puncturing framework for joint eMBB and URLLC traffic[1]. Each mini-slot contains 2 OFDM symbols, so a normal slot contains 7 mini-slots.

similar time-consuming methods) that increase latency. To meet URLLC latency constraints, adaptive frame structures (that provide for mini-slots) and methods to reduce the processing times have been proposed by 3GPP [5]. Shorter user equipment (UE) and next-generation NodeB (gNB) processing times are achieved by introducing a new structure for the data channel and new channel codes [42, 43]. In [44], the authors provide an overview of URLLC and analyze the latency components in 5G NR downlink. Further, to reduce the block error rate (BLER) over a link, more accurate channel estimation by UE is studied in [45, 46], while configurable BLER targets for one-shot transmission given the channel condition is proposed in [47]. What's more, Hybrid Automatic Repeat reQuest (HARQ) is a known effective mechanism within LTE/NR to achieve efficient transmission and low residual BLER. However, opportunities for HARQ retransmissions in URLLC will be limited due to the latency constraints. Hence, to reduce the need to wait for round trip times for HARQ operation, *blind repetitions* whereby multiple packets are sent without waiting for HARQ acknowledgment (ACK) are adopted as a solution in 3GPP [47]. A null-space based preemptive scheduler for joint URLLC and eMBB traffic is proposed for dense 5G networks in [48]; non-orthogonal

multiple access (NOMA) based multiplexing of URLLC and eMBB at the edge nodes is studied in [49, 50]. While these techniques could support simultaneous transmission for eMBB and URLLC users (rather than puncturing eMBB users), these methods are effective when the interference among users is modest. In several use cases, the above is infeasible[2] and puncturing is a practical alternative. Besides [49], the successive interference cancellation algorithm required as a result of NOMA at the receiver could introduce significant additional latency, making it difficult to meet strict latency constraints.

Prior work that focuses solely on URLLC performance includes [51, 52] that proposes a resource allocation and HARQ optimization scheme to minimize the necessary bandwidth for URLLC traffic. Similarly, [53] considers blind retransmission and proposed a model to reduce resource consumption. However, the need is to consider such allocation to URLLC *in conjunction with other existing 5G traffic* and not alone - thus, joint scheduling approaches are needed. In [54,55], a RAN resource allocation problem is formulated as a sum-rate maximization problem subject to the orthogonality constraint (i.e., service isolation), latency-related constraint and minimum rate constraint while maintaining the reliability constraint with the incorporation of adaptive modulation and coding. Common physical network infrastructure is sliced into multiple end-to-end logical networks, each acting as a dedicated network for a specific service. In [56], an optimization problem is formulated that maximizes the minimum expected rate of eMBB users over the long run while meeting the URLLC requirements, but without considering fairness among the eMBB users.

There have been very few studies in this regard; models for eMBB rate loss associated with simultaneous URLLC is first proposed in [57] and is the closest precursor to our effort. However, in that work, the authors propose new schedulers based on the assumption of *fixed set of backlogged eMBB users that all receive resources without using slot aggregation.* First, eMBB traffic is dynamic with users entering and leaving the system (hence not a fixed set). Second, our preferred solution is to come up with a URLLC RB allocation

[2]For example, the passengers are playing video (eMBB) on an auto-driving car (URLLC).

policy in NR given the already existing eMBB allocation, as a convenient upgrade path for systems operation. The proposed resource allocation for URLLC traffic works on a mini-slot scale and maintains current eMBB schedulers on the slot scale. Methodologically, our approach is similar to [58, 59] that also proposed a constraint-based formulation that maximizes the eMBB aggregate data rate while satisfying the URLLC constraints. We differ from the above in two significant ways: the aggregate rate maximization often results in loss of fairness among eMBB users. We thus incorporate proportional fairness concepts within the aggregate utility maximization. Further, we also incorporate HARQ retransmission, which is not considered in the above. In this way, we push the state-of-art for a joint scheduler design while balancing multiple objectives (eMBB rate and latency constraints) while also preserving fairness.

3.1.2 Contributions and Organization

The novel contributions of this work are:

- Based on the existing eMBB scheduler, a model for *joint scheduling* of punctured eMBB and URLLC traffic as an optimization problem that maximizes the eventual aggregate utility of the eMBB users subject to latency constraints for the URLLC users.

- Model for the delay and reliability of URLLC traffic from a media access control (MAC) layer perspective.

- Two new resource allocation algorithms to align with practical implementation for downlink scheduling in the 5G system.

Section 3.2 summarizes the utility maximization framework for eMBB subject to the constraints of URLLC traffic. Section 3.3 describes the proposed system model for joint scheduling. In Section 3.4, two different problem formulations and respective algorithmic solutions are developed and evaluated via simulation testing in Section 3.5.

3.2 Balancing eMBB Utility and URLLC requirements

3.2.1 Review: PF Schedulers for Maximizing eMBB Aggregate Utility in LTE/NR

In 3GPP cellular networks, the MAC scheduler controls the slot-based resource allocation for users based on their respective traffic demands. Three different algorithms are widely used in 4G LTE - round robin, max rate and proportional fair, which have been naturally extended to 5G NR networks. Among those scheduling algorithms, PF scheduler is widely used in cellular networks for the following reason. In any communication network, different users normally experience different channel conditions, so the scheduler shall attempt to exploit the temporal variations by *allocating resources to users with the best conditions for the upcoming time period*. The PF scheduler not only aims to increase the system throughput considering the channel condition but also maintains the long-term allocation proportional fairness between users [60], which is superior to RR and Max Rate scheduler. In this chapter, we analyze and use the well-known PF scheduler algorithm that is widely adopted by NR and LTE for the downlink scheduling of eMBB users. Denote $x(n)$ the data rate of the eMBB user n and $U_n(x(n))$ the utility of the eMBB user n. From [61], the PF schedule's solution $x(n)$ satisfies

$$\sum_{n \in N} \frac{x(n) - x^*(n)}{x^*(n)} \leq 0, \tag{3.1}$$

where $x^*(n)$ denotes the optimal PF solution and for any other feasible $x(n)$, the aggregate proportional fairness [3] gain is non-positive.

We first present the PF scheduler algorithm used in the downlink of several cellular network standards, and discuss its connection to the network utility maximization formulation. Consider a cellular downlink with N users, $R_t(n)$ denotes the average data rate user n has received up to time t in a window of T time slots according to

$$R_t(n) = \left(1 - \frac{1}{T}\right) R_{t-1}(n) + \frac{1}{T} x_{t-1}(n), \tag{3.2}$$

[3] It is noteworthy that a more general α fairness utility function can be expressed as $U(x) = \frac{x^{1-\alpha}}{1-\alpha}$. When $\alpha = 1$, the alpha-fairness is reduced to proportional fairness $U(x) = \log(x)$ which can be derived using L'Hopital's rule.

where $x_{t-1}(n)$ denoting the actual data rate of user n at time $t-1$.

The PF scheduler allocates resources to user n^* who maximizes the ratio of achievable instantaneous data rate over average received data rate, as shown below:

$$n^* = \arg\max \{P_t(n)\}, \tag{3.3}$$

where

$$P_t(n) = \frac{DRC_t(n)}{R_t(n)}, \tag{3.4}$$

where $\mathrm{DRC}_t(n)$ denotes the achievable *instantaneous* data rate for user n at time slot t. The scheduler finds the user with the highest $P_t(n)$ to allocate all RBs in the current time slot t. Then, the user corresponding to the second largest $P_t(n)$ is allocated and so on until there are no RBs left in current slot. As example in Fig 3.1 shows ongoing eMBB allocation pattern, with orange representing allocation to user n^* (top two rows in the resource block table), blue representing RBs allocated to the user with the second largest $P_t(n)$ etc. The PF algorithm [62] has been shown (see App. 3.7.1) to maximize the aggregate eMBB utility while guaranteeing the asymptotic proportional fairness.

3.2.2 URLLC Requirements and Resource Allocation

The QoS requirements for downlink URLLC traffic is modeled as follows: a UE requests the BS to transmit a packet of size p successfully within an end-to-end delay of τ and the probability of failure at most σ [51] [63]. The target reliability σ depends on the users' QoS requirements, while δ is the BLER for each transmission. The δ for single downlink transmission is affected by the channel condition and choice of the Modulation Coding Scheme (MCS). If BLER δ is higher than σ, it may take several retransmissions to achieve the reliability requirement.

To ensure the probability of successful delivery, the delay caused by the retransmission scheme like HARQ in LTE/5G NR must be considered. In this chapter, we analyze the URLLC for one-shot transmission as a function of selected MCS, so the δ is known and

configured according to the URLLC requirements. This method is extended to include retransmissions if we consider the retransmission packets as more constrained URLLC users, i.e., the latency of the retransmission is a reduction of original latency by the passing time. Further, the model could also be extended to optimize the choice of MCS and HARQ processing.

In response to URLLC latency constraints, a series of measures to reduce the processing time in each layer is described in Section 3.1. Specifically, in this work, we leverage the additional mini-slot feature in NR, where each mini-slot is composed of fewer OFDM symbols (a minimum of two) and also able to carry control signals/channels at the beginning and/or end [64].

3GPP standards introduce different use cases for URLLC [3], depending on traffic type and requirements. For example, information exchange in intelligent transportation between a UE supporting V2X application and a V2X application server requires a max end-to-end latency of 5ms and reliability of 99.999% with 1 Mbps downlink data rate. In such cases, while requirements of bandwidth for each user are modest, increasing user numbers and different traffic types will combine (as in a downtown rush-hour scenario) to render the joint scheduling problem considered in this work of increasing significance.

Two downlink resource allocation schemes, Type 0 and Type 1 are supported in 5G NR [47]. Type 0 is *discontinuous* frequency domain resource allocation, but instead of allocating based on the single RB, the resource is further grouped into Resource Block Group (RBG) of size V according to the upper layer protocol (the minimum size of V is 2). In Type 1 downlink resource allocation, RBs are allocated to a scheduled UE *contiguously*. In this chapter, we consider Type 0 schemes to avoid the extra effort in the grouping, i.e., URLLC incoming traffic within an ongoing eMBB slot could be allocated *discontinuously* which provides the flexibility to achieve fairness between eMBB users.

Notation: Lower-case and upper-case boldface letters denote vectors and matrices, respectively; $(\bullet)^T$, $(\bullet)^{-1}$ denote transpose and inverse of a matrix, respectively; the number of non-zero entries of vector $\mathbf{x}$, i.e,. cardinality of vector $\mathbf{x}$ is defined as $\mathbf{card}(\mathbf{x})$. $\lceil \bullet \rceil$ (resp.

$\lfloor \bullet \rfloor$) denotes the ceiling (floor) function, respectively. Other key notations are summarized in Tab. 3.1.

Table 3.1: Key Notations

m / M	URLLC user m / Total URLLC user numbers
n / N	eMBB user n / Total eMBB user numbers
$\mathbb{P}_{\text{fail}}$	the final probability of transmission failure
δ	the probability of transmission failure for each transmission or retransmission
σ	the reliability constraint
T_{delay}	the total delay
TTI	Transmission Time Interval, each delay is an integer multiple of TTI
α	integer, variable, the delay of waiting in the queue, determined by schedule policy
β	integer, constant, the delay of processing in gNB and UE

3.3 System Model

In LTE and NR, frequency-time resources are split into several blocks in the MAC layer - Virtual Resource Block (VRB) - that are subsequently mapped into physical resource blocks (PRBs)[4]. In this section, we first describe the joint scheduling architecture we aim to construct. Then, the latency budget for PRBs allocated to URLLC users is used to formulate

[4]While the mapping details depends on the implementation according to the 3GPP standard [47], different methods have no impact on the scheduling performance as long as we adhere to resource allocation schemes in Section 3.2. Therefore we only focus on PRB allocations in this work.

a latency constrained resource allocation problem for each mini-slot, whereby the aggregate utility for eMBB users is maximized.

3.3.1 Joint Scheduling Architecture

Two different approaches to joint scheduler designers for eMBB and URLLC users have been conceptualized. The first is to leverage the superposition/puncturing framework and schedule based on different time scales, i.e., NR slot and mini-slot, as shown in Fig 3.1. The eMBB scheduler decides at each time slot the PRB resources allocated, and the URLLC scheduler allocates the mini-slots resources within the ongoing eMBB slot, based on the eMBB allocation results. The URLLC scheduler receives the URLLC user demand information within the current slot. Then based on the results from eMBB allocation, the URLLC scheduler determines the allocation policy for each mini-slot. Since each mini-slot has a duration of less than 0.125 ms, it implies that the URLLC scheduler algorithm complexity must be for real-time operation. This approach does not impact the current schedule for eMBB traffic but leads to a reduction of eMBB data rate after puncturing. The second approach is to place both user types in the same queue and jointly schedule on the same time scale without puncturing. This will require more scheduling slots and higher numerology to fulfill the URLLC requirements. While this approach could maintain the data rate of eMBB users, it will require a more complex scheduler to fulfill the low latency requirements of URLLC. Further, in the sub-6 GHz bands, the numerology index is restricted to $\{0, 1\}$, implying that scheduling decisions are based on 1 or 0.5 ms slots, limiting the latency constraints achieved for URLLC traffic. In light of the above considerations, we choose to investigate the first approach in this work. Since already allocated eMBB users' PRBs will be punctured to multiplex URLLC users, thus inevitably reducing eMBB users' data rate, the objective of our scheduler is to maximize the aggregate utility of eMBB users while meeting the latency and reliability requirements of URLLC users. Thus we split the URLLC scheduler into two parts - user selection and resource allocation. We first decide which mini-slots can meet the latency and reliability constraints for an arriving URLLC user and then decide how to

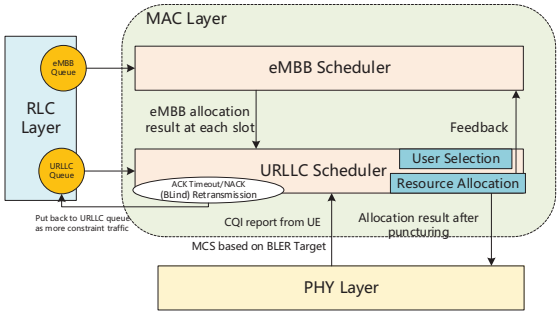


Figure 3.2: Joint scheduler of eMBB and URLLC users

allocate the URLLC users within these mini-slots so as to maximize the aggregate utility of the eMBB users. The joint scheduler architecture is shown in Fig 3.2. The Radio Link Control (RLC) layer reports the buffer status at the beginning of the slot/mini-slot to the scheduler. The Physical (PHY) layer will report the channel condition for different users and the selected MCS from the channel quality indicator (CQI)[5] reported from UE. The eMBB scheduler selects users for transmission and allocates resources for them on a slot basis and reports to the URLLC scheduler. Then the URLLC scheduler seeks to send out its scheduled packets in each mini-slot based on the eMBB result and return it to the eMBB users after its allocation. The URLLC scheduler will notify the PHY layer to set the eMBB users' transmission power to zero on the resource allocated to URLLC users as puncturing and notify the eMBB scheduler of the allocation results. If the URLLC scheduler receives the NACK or ACK timeout, it will start retransmission or blind retransmission by immediately scheduling at the current mini-slot.

[5]There are two types of CQI transmission - periodic and aperiodic - for both eMBB and URLLC. Aperiodic mode causes more delay since BS must request the UE to send their report, so we consider the only periodic transmission. CQI is transmitted periodically with a minimum periodicity of 2 ms - the value in the simulation - specified by a higher layer message. The scheduler uses the latest CQI value for choosing the MCS as per the standard.

3.3.2 User Selection Based on URLLC Constraint

URLLC Reliability Constraint

In 5G downlink scheduling, we expect the channel condition between each user and base station, i.e. BLER to be below acceptable limits - 10^{-5} in NR, 10^{-1} in LTE. The UE estimates the downlink channel and reports back CQI for BS to choose the MCS to meet the BLER requirement δ for its transmission. Thus, during each scheduling procedure, the BS knows the BLER for given MCS according to the configuration. Note that most URLLC traffic has strict latency constraint target (less than 1 ms) with error rate less than 10^{-5}. In practice, the BLER target is set to 10^{-3} to achieve higher throughput but also allow retransmissions for lower BLER, thereby allowing adaptation for different service requirements. Consider the architecture in Fig 3.2, the URLLC scheduler assumes that the inputs from the PHY layer are accurate and estimates the BLER probability for the MCS chosen. Thereafter, it resorts to blind retransmissions if the ACK/NACK from HARQ is not received within a time-out interval. Each transmission/retransmission is independent, and the probability of failure is given by

$$\mathbb{P}_{\text{fail}}(m) = (\delta_m)^{k_{\text{trans}}(m)} \leq \sigma_m, \tag{3.5}$$

where σ_m is the reliability constraint, $k_{\text{trans}}(m) = k_{\text{retrans}}(m) + 1$, denoting the total number of transmission and retransmissions for user m for BLER target δ, respectively. k_{trans} can be obtained from Eq. (3.5) to indicate the # of blind retransmissions:

$$k_{\text{trans}}(m) = \left\lceil \frac{\log(\sigma_m)}{\log(\delta_m)} \right\rceil. \tag{3.6}$$

URLLC Latency Constraint

LTE system latency in the user plane is typically measured as a multiple of Transmission Time Interval (TTI). The analysis for NR can reuse the same approach but with different system parameters due to enhanced hardware capability, summarised by Samsung [65] and

[66]. The NR TTI length is equal to the slot (eMBB) or mini-slot (URLLC) length[6]. The total latency can be split into several parts shown in Fig 3.3.

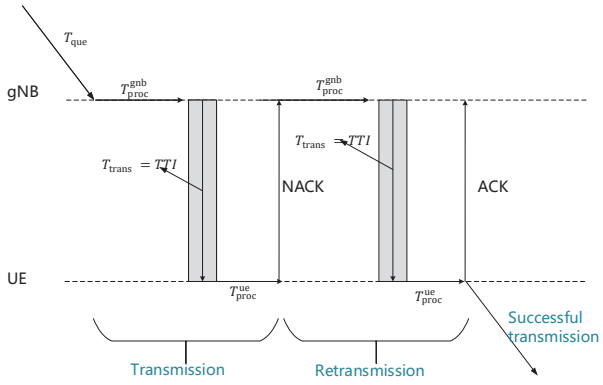


Figure 3.3: Illustration of latency components for DL transmissions in NR.

One-shot transmission: The total delay depends upon 4 components - i) queuing time before allocation T_{que}, ii) processing time at gNB $T_{\text{proc}}^{\text{gnb}}$, iii) transmission time T_{trans}, and iv) processing time at UE $T_{\text{proc}}^{\text{ue}}$. The transmission time $T_{\text{trans}} = TTI$ and we assume that all these delay components are a multiple of URLLC TTI (same length as mini-slot) as suggested above. Then the total delay of URLLC user m is calculated in Eq. (3.7).

$$\begin{aligned} T_{\text{delay}}^{\text{once}}(m) &= T_{\text{que}}(m) + T_{\text{proc}}^{\text{gnb}}(m) + T_{\text{proc}}^{\text{ue}}(m) + T_{\text{trans}}(m) \\ &= T_{\text{que}}(m) + T_{\text{proc}}(m) + T_{\text{trans}}(m) \\ &= (\alpha_m + \beta_m + 1) \times TTI, \end{aligned} \tag{3.7}$$

where $\alpha_m \in \mathcal{Z}^{++}$ is a non-negative integer and represents the multiplier of queuing delay which depends on the scheduler policy, and $\beta_m \in \mathcal{Z}^{++}$ is multiplier of total processing delay, normally $\beta = 4$ [65, 67]. Since retransmission have the highest priority and are immediately scheduled, there is no queuing delay, thus $T_{\text{retrans}}^{\text{once}}(m) = T_{\text{proc}}(m) + T_{\text{trans}}(m) = (\beta_m + 1) \times TTI$.

[6]For example, for a 2-symbol mini-slot, the TTI is 1/7 ms when using numerology 0 whose slot length is 1 ms.

So the total delay for the transmission and k_{retrans} retransmissions must satisfy

$$\begin{aligned} T_{\text{delay}} &= T_{\text{delay}}^{\text{once}}(m) + k_{\text{retrans}}(m) \times T_{\text{retrans}}^{\text{once}}(m) \\ &= (\alpha_m + \beta_m + 1 + k_{\text{retrans}}(m) \times (\beta_m + 1)) \times TTI \\ &= (\alpha_m + k_{\text{trans}}(m) \times (\beta_m + 1)) \times TTI \\ &\leq \tau_m, \end{aligned} \tag{3.8}$$

From the above, we conclude that if the system BLER is lower than the acceptable probability of failure, then only one transmission without any retransmission is sufficient. For VR/AR applications with 10 ms latency and 99.9% reliability requirements, the value of α_m can be increased such the URLLC packets can be transmitted in the *following* eMBB slot instead of the current mini-slot. This mitigates the impact on eMBB users, as shown in Fig 3.1; because the next eMBB slot has not been allocated, the URLLC allocation does not puncture the eMBB users in the current slot. Thus we relax the presumption in many current works that URLLC be immediately allocated resources when these packets arrive, depending on the use case.

Let the actual delay T_{delay} equal the latency constraint τ in Eq. (3.8), then we obtain the upper bound of queuing delay α_m for user m, i.e.,

$$\alpha_m = \left\lfloor \frac{\tau}{TTI} \right\rfloor - k_{\text{trans}}(m) \times (\beta_m + 1). \tag{3.9}$$

Thus, the latency requirement is met by allowing user m to transmit within α_m mini-slots starting from the URLLC user arrival mini-slot d[7]. However, if $\alpha_m \leq 0$, the URLLC scheduler has no choice but to transmit instantly even though the latency requirement may not be satisfied. The value of queuing delay also indicates the priority of each URLLC user, i.e., larger the α_m, lower the priority. If $d + \alpha_m$ falls into the next eMBB slot, this URLLC transmission will be scheduled in the next slot to avoid puncturing the existing eMBB users,

[7]We split the eMBB slot into several mini-slots, for example, seven mini-slots and the order is from 1-7. If $d = 3$, it means the URLLC user comes in at the third mini-slot. Note that α_m is an integer and TTI length equals to the mini-slot length, so if $d + \alpha_m > 7$, it means that the URLLC could wait for the next eMBB slot to be scheduled.

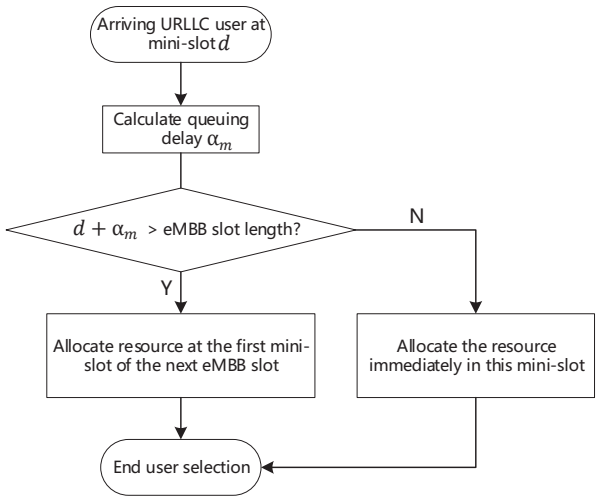


Figure 3.4: The flow chart of the user selection procedure.

which leads to the retransmission of punctured eMBB users. For other users, if $d + \alpha_m$ still falls in the slot, we schedule these users in the current mini-slot to reduce the latency and avoid overload in the next mini-slot. In this way, we select the users to transmit on different mini-slots; the flow chart of the user selection procedure is shown in Fig 3.4.

3.3.3 Resource Allocation: Maximize eMBB Aggregate Utility

Consider PRB level resource allocation for both eMBB and URLLC users. Denote $r(i)$ as the number of PRBs user i needs defined in Eq. (3.10).

$$r(i) = \left\lceil \frac{p(i)}{\psi(i)} \right\rceil, \tag{3.10}$$

where $p(i)$ denotes the packet size of user i, and $\psi(i)$ denotes the data amount carried by this PRB for chosen MCS. Hence $\psi(i) = TTI \times b \times \rho(i)$ where b denotes the bandwidth of the PRB, TTI is the TTI (mini-slot/slot) length of the PRB. $\rho(i)$ denotes the spectral efficiency of chosen MCS of user m with in unit bit/s/Hz. For a selected numerology, TTI and b are the same for all users, so the $\rho(i)$ determines the capacity of the PRB. For each

eMBB user, the data rate loss results from the amount of PRBs replaced by the URLLC users in each slot. The eventual data rate of eMBB user n, $\phi(n)$ after the allocation of the URLLC users in a slot is

$$\phi(n) = \left(1 - \frac{\sum\limits_{m \in \mathcal{S}(n,m)} \tilde{r}_n(m)}{r(n)}\right) x(n), \tag{3.11}$$

where $x(n)$ denotes the original data rate of eMBB user n. $x(n)$ will be updated by $\phi(n)$ once the optimal $\phi(n)$ is obtained so that the optimization for the next mini-slot includes the information from the previous mini-slot, $\tilde{r}_n(m)$ denotes that the number of PRBs of URLLC user m overlapping with PRBs of eMBB user n, and $\mathcal{S}(n,m)$ denotes the index set of URLLC users m whose PRBs overlap with PRBs of the eMBB user n. Note that $\phi(n)$ only represents the data rate of the harmed eMBB user n within a given mini-slot. To achieve the proportional fairness for long term, we need to consider the data rate reduction of eMBB user over sufficiently large number of time slots, i.e. average eMBB user data throughput over T time slots is

$$\Phi(n) = \frac{1}{T}\phi(n) + (1 - \frac{1}{T})R_{t-1}(n), \tag{3.12}$$

where the current mini-slot is in time slot t, $R_{t-1}(n)$ denotes the average data rate of eMBB user n in the previous $T-1$ time slots.

Denote the allocation result of each URLLC user m with a row vector $\tilde{\mathbf{x}}_m^T \in \{0,1\}^{1\times L}$ where $\tilde{\mathbf{x}}_{m,l}^T = 1$ means that URLLC user m takes up l^{th} resource block and 0 otherwise. L denotes the total number of PRBs available for scheduling. Then we get the final allocation result in a mini-slot for all M URLLC users as a new matrix $\mathbf{X} \in \{0,1\}^{M\times L}$, expressed as

$$\mathbf{X} = \begin{bmatrix} \tilde{\mathbf{x}}_1^T \\ \vdots \\ \tilde{\mathbf{x}}_m^T \\ \vdots \\ \tilde{\mathbf{x}}_M^T \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_l & \cdots & \mathbf{x}_L \end{bmatrix}. \tag{3.13}$$

Since in each mini-slot, one URLLC user can be allocated, implies that any column of the matrix $\mathbf{X}$ has at most one "1", i.e., $\mathbf{card}(\mathbf{x}) <= 1$. Since $\mathbf{x}$ is binary $0-1$ vector, it

follows that

$$\mathbf{1}^T\mathbf{x}_l \leq 1, \;\; \mathbf{card}(\mathbf{x}) \leq 1, \; \forall l \in \{1, \ldots, L\}. \tag{3.14}$$

For the PRB allocation for N eMBB users, $\mathbf{Y}$ is defined as follows,

$$\mathbf{Y} = \left[\mathbf{y}_1 \cdots \mathbf{y}_n \cdots \mathbf{y}_N \right], \tag{3.15}$$

where $\mathbf{Y} \in \{1\}^{1\times L}$. Denote $[0 \ldots \mathbf{y}_n \ldots 0]$ the matrix with all other $\mathbf{y}_i = \mathbf{0}$, $i \in \{1, \ldots, N\} - \{n\}$, i.e., only $\mathbf{y}_n$ remains unchanged and all the other matrices are set to zero. For example, for $\mathbf{Y} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$ where $\mathbf{y}_1 = \begin{bmatrix} 1 \end{bmatrix}$, $\mathbf{y}_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}$, and $\mathbf{y}_3 = \begin{bmatrix} 1 \end{bmatrix}$. $[0 \ \mathbf{y}_2 \ 0] = [0 \ 1 \ 1 \ 0]$ denotes that 2^{nd} and 3^{rd} resource blocks are taken by eMBB user 2 corresponding to $\mathbf{y}_2 = [1 \ 1]$. To quantify the amount eMBB users is harmed by URLLC, we calculate the number of the overlapping part between $\mathbf{y}_n$ and matrix $\mathbf{X}$ for eMBB user n, i.e.,

$$\sum_{m\in\mathcal{S}(n,m)} \tilde{r}_n(m) = \mathrm{Tr}(\mathbf{X}(:, \mathrm{col}([0 \ldots \mathbf{y}_n \ldots 0]))^T(\mathbf{1}_{M\times 1}\mathbf{1}_{L\times 1}^T)), \tag{3.16}$$

where col($\bullet$) returns non-zero column indexes of matrix, and $\mathbf{X}(:, \mathrm{col}(\mathbf{Y}))$ returns the matrix corresponding to col($\mathbf{Y}$) indices. The data rate after puncturing each eMBB user n is expressed as follows:

$$\phi(n) = (1 - \frac{\mathrm{Tr}(\mathbf{X}(:, \mathrm{col}([0 \ldots \mathbf{y}_n \ldots 0]))^T(\mathbf{1}_{M\times 1}\mathbf{1}_{L\times 1}^T))}{r(n)})x(n). \tag{3.17}$$

Eq. (3.17) is a convex function in $\mathbf{X}$, since $\mathbf{X}(:, \mathrm{col}([0 \ldots \mathbf{y}_n \ldots 0]))$ and Tr($\bullet$) are both linear operation. Then we express the problem as an Integer Program (IP) as shown below.

Problem 9 (Original Resource Allocation Problem)

$$\max \sum_{n=1}^{N} U(\frac{1}{T}((1 - \frac{Tr(\mathbf{X}(:, col([0 \ldots \mathbf{y}_n \ldots 0]))^T(\mathbf{1}_{M\times 1}\mathbf{1}_{L\times 1}^T))}{r(n)})x(n)) + (1 - \frac{1}{T}R_{t-1}(n))) \tag{3.18}$$

$$s.t. \ (3.14), \mathbf{X}_{m,l} = \{0, 1\} \ \ \forall m \in \{1, \ldots, M\} \ and \ l \in \{1, \ldots, L\}. \tag{3.19}$$

Note that **Problem** 1 is an IP problem because each entry of $\mathbf{X}$ can only be 0 or 1. The utility function $U(\Phi(n)) = \log \Phi(n)$, for example, when we aim to achieve proportional fairness.

3.4 URLLC Resource Allocation in Mini-slot

In this section, we first outline a combinatorial brute-force solution for Eq. (3.18) for URLLC resource allocation in each mini-slot to maximize the eMBB utility functions and quickly move towards a pragmatic solution by reformulating Eq. (3.18) into a difference of convex problem, which can be solved using the concave-convex procedure. Finally, an online greedy algorithm with lower time complexity is proposed based on proportional fairness (PF) scheduler.

3.4.1 Combinatorial Approach

Assume that the number of PRBs for all URLLC in the current mini-slot is less than the number of total PRBs L in the current mini-slot, so we have the following:

$$L - \sum_{m=1}^{M} r(m) > 0. \tag{3.20}$$

Now we analyze the complexity of the brute-force algorithm. If we fix the URLLC PRB allocation area with only the freedom to permute the order of each URLLC PRB, then there are $\{\sum_{m=1}^{M} r(m)\}!$ different permutations of URLLC PRB allocations. Hence, we know the time complexity of such brute-force Algorithm is obviously much larger than $\mathcal{O}\left(\{\sum_{m=1}^{M} r(m)\}!\right)$. Therefore, the brute-force ordering algorithm is an NP-hard problem.

3.4.2 Convex Relaxation

In this section, we reformulate the natural Integer Programming (IP) problem as equivalent to difference of convex (DC) programming that can be solved with an exact solution exploiting concave-convex procedure (CCCP) efficiently. First, note that an equivalent constraint to Eq.(3.19) is

$$\begin{aligned} &\mathbf{X}_{m,l}(1 - \mathbf{X}_{m,l}) \leq 0, \mathbf{X}_{m,l} \in [0, 1], \\ &\forall\ m \in \{1, \ldots, M\}, \forall\ l \in \{1, \ldots, L\}. \end{aligned} \tag{3.21}$$

Problem 10 (Equivalent Resource allocation Problem)

$$\max \sum_{n=1}^{N} U(\Phi(n))$$
$$s.t.\ (3.14), (3.17), (3.21).$$

Note that while it is now cast as a problem over continuous variables, Problem 2 is still not a convex optimization problem since constraint (3.21) represents a disconnected set. To be more specific, $\mathbf{X}_{m,l}(1-\mathbf{X}_{m,l})$ is a concave function, so any sub-set is obviously a disconnected set.

Next in the interests of tractability, we further transform the above by penalizing the concave constraint into the objective function.

Problem 11 (Penalized Resource Allocation Optimization)

$$\min -\sum_{n=1}^{N} U(\Phi(n)) + \zeta \sum_{m=1}^{M}\sum_{l=1}^{L} \mathbf{X}_{m,l}(1-\mathbf{X}_{m,l})$$
$$s.t.\ (3.14), (3.17) \tag{3.22}$$

where we introduced a penalty parameter $\zeta > 0$. The above function can be visualized as a linearly constrained indefinite quadratic problem (IQP) which falls into the category of different of convex (DC) problem. The following lemma shows the equivalency of the DC problem to the original problem [34].

Lemma 3 *If* $u > \zeta_0$ *where*

$$\zeta_0 = \frac{-\sum_{n=1}^{N} U(\Phi(n)) - \gamma}{max_{\mathbf{X}}\{\sum_{m=1}^{M}\sum_{l=1}^{L} \mathbf{X}_{m,l}(\mathbf{X}_{m,l}-1) : (3.14), (3.17)\}}, \tag{3.23}$$

where

$$\gamma = min_{\mathbf{X}}\{-\sum_{n=1}^{N} U(\Phi(n)) : \mathbf{X}_{m,l}(1-\mathbf{X}_{m,l}) \geq 0\}, \tag{3.24}$$

then Problem 2 is equivalent to Problem 3.

Algorithm 2: CCCP Algorithm

Initialize Find an initial feasible matrix $\mathbf{X}^{(0)}$ of Problem 3 and set $i = 0$;

while *Error* ϵ *does not converge to the acceptable value* **do**

Solve for $\mathbf{X}^{(i+1)}$ as the optimal solution in the following convex problem:

$$\begin{aligned} &\min\{-\sum_{n=1}^{N} U(\Phi(n)) - \zeta \mathbf{F}(\mathbf{X}; \mathbf{X}^{(i)})\} \\ &s.t. (3.14), (3.17) \end{aligned} \tag{3.25}$$

Where

$\mathbf{F}(\mathbf{X}; \mathbf{X}^{(i)}) \triangleq \sum_{l=1}^{L} \sum_{m=1}^{M} \mathbf{X}_{m,l}^{(i)}(1 - \mathbf{X}_{m,l}^{(i)}) + \sum_{l=1}^{L} \sum_{m=1}^{M} (\mathbf{X}_{m,l} - \mathbf{X}_{m,l}^{(i)})(1 - 2\mathbf{X}_{m,l}^{(i)})$.;

$i \leftarrow i + 1$;

end

It is noteworthy that $\mathbf{X}_{m,l}(1 - \mathbf{X}_{m,l}) \geq 0$ is a convex constraint. The local optima of Problem 3 can be obtained in finite steps via DC algorithms (DCA). Moreover, the objective function in Problem 3 is differentiable, so DCA is naturally reduced to CCCP [68], as shown in Algorithm 2. CCCP iteratively solves convex problems obtained by linearizing the penalty term of the objective function in IQP. Taking derivative of the penalty term gives us $1 - 2\mathbf{X}_{m,l}$, then we apply first-order linear approximation and we obtain $\mathbf{X}_{m,l}^{(t)}(1 - \mathbf{X}_{m,l}^{(t)}) + (1 - 2\mathbf{X}_{m,l})(\mathbf{X}_{m,l} - \mathbf{X}_{m,l}^{(t)})$, where $\mathbf{X}_{m,l}^{(t)}$ denotes the solution of t^{th} CCCP iteration. The time complexity of each iteration of the CCCP algorithm is lower bounded by $\mathcal{O}((ML)^{2+\frac{1}{6}} \log(\frac{ML}{e}))$ [69], where e denotes the relative accuracy.

3.4.3 Greedy Algorithm

Although the above CCCP indeed yields a near-optimal solution, the computation efficiency may not suffice to output the solution within each mini-slot. In order to solve the resource allocation more efficiently, we propose a greedy algorithm 3, which maximizes the aggregate

utility for each arriving users inspired by the PF scheduler shown in Algorithm. 3. The complexity of this algorithm is $\mathcal{O}(MLN)$. We use the same formula to calculate the PF weight $P(n)$ in Eq. (3.4) for eMBB users. The main difference between the traditional PF algorithm and our proposed greedy algorithm is that the traditional PF will sort all eMBB users according to the $P(n)$ and then sequentially allocate one by one. In our proposed greedy algorithm based on PF scheduler, we allocate the RB requested by URLLC one-by-one by puncturing the RBs which has already been allocated to the existing eMBB users, by choosing those that have the least impact on original $P(n)$. It is noteworthy that, after each allocation, the $P(n)$ varies in response to the changes of eMBB users' data rates, which in turn will lead to different allocation decisions for the future RBs.

3.5 Simulations

In this section, we analyze the performance of the basic RR, CCCP and dynamic greedy algorithm for downlink URLLC scheduling [8]. Basic RR, as a baseline algorithm, only allocates PRBs from top to bottom within each mini-slot, so it does not consider the impact upon the fairness among eMBB users. RR algorithm also does not help maximize the eMBB users' data rate since it does not consider which eMBB is more susceptible to data rate reduction from URLLC users' PRBs. The trade-off between the CCCP algorithm and the dynamic greedy algorithm is analyzed in terms of the performance and time efficiency.

3.5.1 Simulation Setup

The simulation is implemented on a computer with a single CPU (Intel(R) Core(TM) i7-7700K) and implemented using Matlab R2019b with the CVX toolbox. To accelerate overall runtime, we implement the greedy algorithm in C on the same machine. We consider a system with a total of 120 PRBs (60 RBGs) per eMBB slot, with seven mini-slots per eMBB slot and numerology 1. The system has five eMBB users, and the RBG numbers needed to

[8]Note that the basic RR algorithm is not applied to eMBB but URLLC scheduling.

Algorithm 3: Greedy Algorithm using PF scheduler in each mini-slot. $R_t(n)$ denotes the low-pass filtered average data-rate user n has received up to time t.

$\phi(n) \leftarrow x(n)$;
if *packets of URLLC user m arrival* **then**
  for *each PRB of URLLC user m* **do**
    $$R_t(n) \leftarrow \left(1 - \frac{1}{T}\right) R_{t-1}(n) + \frac{1}{T}\phi(n), \tag{3.26}$$
    where T is the number of time slots we average for data rate.
    $$n^* \leftarrow \arg\max \{P_t(n)\}, \tag{3.27}$$
    where
    $$P_t(n) = \frac{\phi(n)}{R_t(n)}, \tag{3.28}$$
    Puncture eMBB n^* with the current PRB of URLLC user and the data rate of the eMBB user is updated as follows:
    $$\phi(n^*) \leftarrow \phi(n^*) - \frac{1}{r(n^*)}\phi(n^*). \tag{3.29}$$
    Set the $\phi(n)$ as the original data rate $x(n)$ in the next mini-slot, i.e., $x(n) \leftarrow \phi(n)$;
  end
else
  Do nothing;
end

achieve their maximum DRC are $\{10, 12, 12, 16, 18\}$. We assume these remain the same during each simulation such that, for example, eMBB user one always request 10 RBGs to achieve its maximum DRC in each slot. Note that the total number of RBGs requested by the eMBB users is larger than the system can afford. Hence, in order to achieve the long-time fairness among eMBB users, we apply the PF principle that forces the eMBB scheduler to choose (out of 60 RBGs), which chunk of RBGs is going to be occupied by which eMBB user.

Cases: Different channel conditions for the eMBB users are used to demarcate cases. In Case 1, all the eMBB users have same channel condition, $\psi = 2$ Kbits, i.e., each RBG carries 2 Kbits data. In case 2, the channel condition varies - User 1 has the best channel condition - $\psi(1) = 3$ Kbits - while users 2 & 3 remains the same as case 1, i.e., $\psi(2) = \psi(3) = 2$ Kbits. User 4 and user 5 have worse channel condition such that $\psi(4) = 1.75$ Kbits and $\psi(5) = 1.5$ Kbits. DRC in unit of Mbps for each user can be expressed as follows:

$$DRC(n) = \frac{r(n) \times \psi(n)}{2 \times 0.5 \times 10^{-3} \text{ s}}, \tag{3.30}$$

where $r(n)$ is the PRB numbers (thus $\frac{r(n)}{2}$ is the RBG numbers) requested by eMBB user n and the slot length in our simulation is 0.5×10^{-3} s.

The RBG numbers required by URLLC users are uniformly distributed with the mean value of 2 RBGs from the set {1, 2, 3} RBGs. In this setup, the simulation explores the extent to which the number of URLLC users impacts the performance of three algorithms for simulation parameters in Table 3.2.

Table 3.2: Simulation Parameters

Parameters	**Value**		
eMBB Users	Numbers		5
	$\psi(n)$ (Kbits)	Case1	{2,2,2,2,2}
		Case2	{1,2,2,1.75,1.5}
	RBG Numbers Request Each slot		{10, 12, 12, 16, 18}
URLLC users	RBG Numbers Request Each Mini-Slot		uniformly distribution from {1,2,3}
System	Total PRB Numbers		120 (60 RBGs)
	Numerology		1
	Mini-Slot Symbol Number		2

3.5.2 URLLC User Selection: Latency and Reliability

For the current setting with Numerology 1, slot length 0.5 ms and mini-slot length equals to 2 OFDMA symbols, the TTI quals to 0.0714 ms for URLLC users and processing time

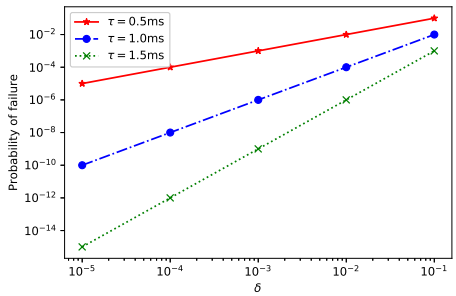
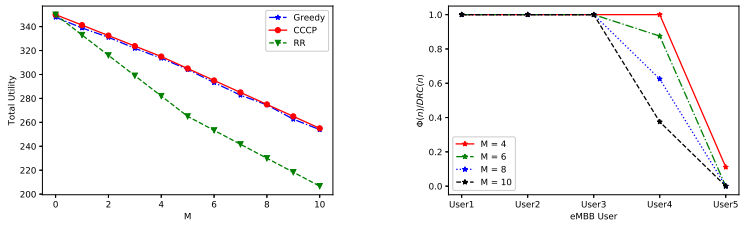


Figure 3.5: URLLC Users' probability of failure with latency less than τ ms vs. BLER target δ.

coefficient β equals to 4. Each simulation run duration is 200s and is repeated a total of 100 times to acquire statistics. The QoS requirements for downlink URLLC traffic is modeled as follows: a UE requests the BS to transmit successfully within an end-to-end delay of τ and the probability of failure at most σ. In the simulation, we consider the cases with different τ and calculate the corresponding failure probability. We conduct the simulation of 10 URLLC users arriving in each mini-slot versus different BLER target configuration from 10^{-5} to 10^{-1}. The percentage of the transmission failure within 1ms delay versus different BLER settings is plotted to verify how well the user selection performs to meet the URLLC QoS requirements. The results for the URLLC constraint is shown in Fig 3.5; it is observed that the BLER target impacts the final result of the reliability within τ ms latency constraint. For the current numerology setting (0.5 ms slot), the maximum transmission and retransmission times k_{trans} are {1, 2, 3} for $\tau = \{0.5, 1, 1.5\}$ ms respectively. As the BLER target σ increases, the reliability drops, and more retransmission is required, increasing the latency. It cannot fulfill the requirements ($\sigma \leq 10^{-5}$) after $\delta \geq 10^{-1}$ for all different τ, even if all the packets are immediately scheduled in the current setting.



(a) Total utility vs URLLC users M per mini-slot.

(b) The $\frac{\Phi(n)}{DRC(n)}$ of each eMBB user.

Figure 3.6: Simulation result with the objective of maximizing total rate, i.e., $U(\Phi(n)) = \Phi(n)$.

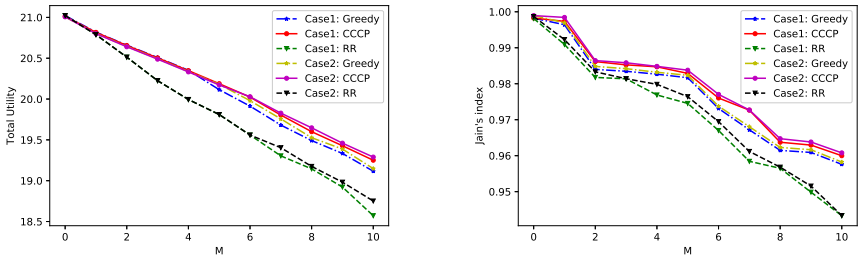
3.5.3 URLLC Resource Allocation

In this section, we first present a simple case with the objective of maximizing the aggregated throughput of eMBB users, i.e., $U(\Phi(n)) = \Phi(n)$ with the same setting as in case 2 above. Fig 3.6 shows that using greedy algorithm and CCCP algorithm yield (nearly) the same result, because with this objective function, the scheduler only needs to select the user with the worst channel condition (user with the smallest $\psi(n)$) with the least harm. As the number of URLLC users per mini-slot M increases, the last two users see throughput drop (even to 0), hence there is no fairness between users.

Proportional Fairness: Next we simulate the 2 different cases mentioned above with the objective to achieve the proportional fairness, i.e., $U(\Phi(n)) = \log(\Phi(n))$ with 200s simulation run durations and compare the results of different algorithms proposed in Section 3.4. In Fig 3.7(a), eventual aggregate eMBB utility affected by URLLC allocation versus the number of URLLC users are plotted. As the number of URLLC users increases, the utility drops for all RR, CCCP, and greedy algorithms. The performance of the CCCP algorithm is the best, especially when the density of the URLLC network increases. For the RR algorithm, the same eMBB users are constantly punctured because RR always punctures eMBB users from

top to bottom (from orange to green PRBs) in Fig 3.1. In other words, no matter how high the priority, the first user is punctured by the RR algorithm. Hence, there is no fairness, and the total utility drops, as expected in Fig 3.7(a). We conclude that the RR algorithm is not suitable for the URLLC scheduler, although it is useful in eMBB scheduling.

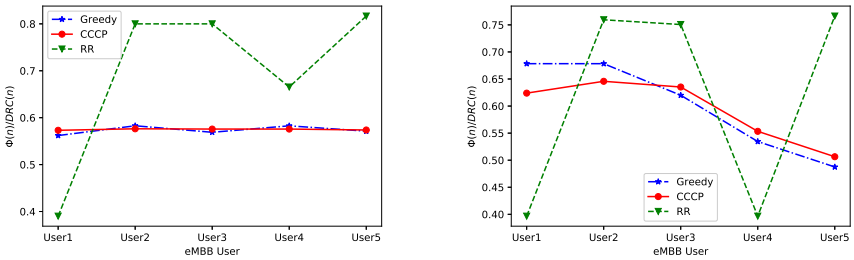
The CCCP and greedy algorithms consider fairness among the eMBB users and have much better performance than the RR algorithm in general. However, greedy algorithms decide on a single PRB allocation at a time, given all URLLC demands for PRBs. Hence, its performance is compromised compared to CCCP, which makes decisions based on all information of requested URLLC PRBs. In Fig 3.7(a), we observe that the slope of CCCP does not change as the number of URLLC users increases while the slope for the greedy algorithm drops slightly, supporting the previous analysis. The fairness among eMBB users is calculated based on the Jain's fairness index and shown in Fig 3.7(b); increasing the URLLC numbers per mini-slot leads to greater unfairness among the eMBB users and hence reduces Jain's index.



(a) Total utility $\sum_{n=1}^{N} U(\Phi(n))$ (b) Jain's fairness index

Figure 3.7: Total utility and Jain's fairness index vs the number of URLLC user numbers per mini-slot. The difference between the two cases is in case 2, $\psi(n)$ varies among eMBB users while $\psi(n)$ is equal for all eMBB users. M denotes the total URLLC numbers in each mini-slot.

Another approach to proportional fairness for eMBB users is the observation that the



(a) Case1 (b) Case2

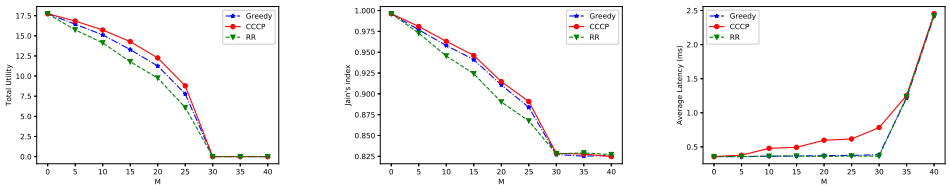
Figure 3.8: The $\frac{\Phi(n)}{DRC(n)}$ of each eMBB, $M = 8$ per mini-slot.

average throughput of the eMBB users is proportional to its DRC, as in Eq. (3.36). Thus $\frac{\Phi(n)}{DRC(n)}$ is computed after each 200s simulation run when the number of URLLC users arriving at each mini-slot is 8. The results are shown in Fig 3.8(a) and Fig 3.8(b). In case 1, $\psi(n)$ of each eMBB user n is the same according to Tab. 3.2, i.e., the channel condition is the same for all users, and the resulting loss is only related to how many PRBs of the eMBB user is punctured. After the allocation of URLLC users, the proportion remains constant between eMBB users using the CCCP algorithm. The greedy algorithm achieves nearly as a good PF metric as CCCP even though the greedy algorithm only allocates one PRB at a time, while RR achieves no fairness.

Case 2 in Fig 3.8(b) is more complex as the $\psi(n)$ varies for each eMBB user according to Tab. 3.2. As we can observe, the CCCP outperforms the greedy algorithm but is unable to achieve ideal PF (constant). This is understandable for the following reason: each eMBB user takes a *different* number of RBGs per mini-slot. If there is no URLLC user, then the throughput of each eMBB user is always proportional to DRC. However, in the presence of URLLC user, to obtain the perfect proportional fairness, i.e., $\frac{\Phi(1)}{DRC(1)} = \cdots = \frac{\Phi(n)}{DRC(n)} = \cdots = \frac{\Phi(N)}{DRC(N)} = c$, we must have $\Phi(n) = c \times DRC(n)$. Note that in this case, ideal PF (constant value for eMBB users) is not achieved, although the Fairness index is considerably better than RR. Since the original problem is naturally an integer programming, i.e. the

punctured RBs are integers, we cannot achieve the desired ideal PF solution obtained over the continuous real domain, and is the fundamental reason for the approximate PF values achieved.

The results in Fig 3.9 demonstrate that as we increase the URLLC number to the limit the system can afford (up to 30 in current setting[9]), all the resources are occupied by the URLLC. When the URLLC traffic load exceeds the maximum resources the system can provide, the utility of the eMBB users drop to 0. While the eMBB users still get scheduled, their actual throughput is 0, so the priority among the eMBB users and the Jain's fairness index remains the same. The average latency for 99.999% reliability is shown in Fig 3.9(c). As the URLLC number increases, the complexity of RR and greedy algorithms increase only marginally and the resulting latency is mainly dependent on the mini-slot that is scheduled. However, the complexity of the CCCP algorithm increases rapidly, which results in longer execution times and increases the total latency. For URLLC numbers beyond 30, all URLLC users could not be scheduled to meet the latency requirement of 1 ms; the latency is caused mainly by the queuing rather than the URLLC scheduling delay, hence the change of the latency after reaching the maximum of 30 in Fig 3.9(c) behaves the same for different schedulers.



(a) Total utility (b) Jain's fairness index (c) Average latency

Figure 3.9: Total utility, Jain's fairness index and average latency vs the number of URLLC user numbers per mini-slot for various CQI.

[9]The total number of RBG in the simulation is 60, and each URLLC user attempts to take up 2 RBGs each mini-slot. So when the URLLC numbers beyond 30, all the RBGs are taken by URLLC users.

3.5.4 Algorithm Complexity

As shown in Sec. IV, the time complexity of greedy algorithm is $\mathcal{O}(MLN)$ and the CCCP algorithm $\mathcal{O}((ML)^{2+\frac{1}{6}}\log(\frac{ML}{e}))$. We can conclude that the CCCP algorithm has a higher order than the greedy algorithm in terms of the time complexity. Hence, we summarize the following trade-off: when the number of URLLC users is large, CCCP has better performance but longer run times. On the other hand, the greedy algorithm can time efficiently allocate URLLC users' PRBs with some performance sacrifices. The choice of CCCP vs. greedy algorithm also depends on the performance of the embedded system employed and use case latency requirements. For less strict latency constrained scenarios implemented on embedded systems with a fast CPU, the CCCP algorithm may be uniformly preferred for its performance. If the latency constraint is stringent, and the embedded system has a constrained CPU, the greedy algorithm is likely to be preferred.

3.6 Summary

To study how data is allocated in 5G system resource pool, a new downlink scheduler for URLLC is proposed. The integrity of the current existing eMBB PF scheduling is preserved for compatibility with existing 5G standard oriented practices. The reliability and latency requirements of URLLC are transformed into an optimal policy formulation for user selection as the first step in the URLLC scheduler design. The subsequent resource allocation step is formulated as an integer programming problem, which can be solved by the proposed CCCP and greedy algorithm. Both algorithms achieve proportional fairness among multiple eMBB users while maximizing the aggregate eMBB users' data rate after puncturing. The performance and running time of two algorithms are extensively compared in the trade-off analysis and evaluated for use in different scenarios.

3.7 The proof of Eq. (3.36)

3.7.1 Asymptotic Proportional Fairness

Denote $\pi_t(n)$ the probability of the transmission of the user n at time slot t. $DRC(n)$ denotes the achievable instantaneous data rate for user n. Let $R(n)$ denote the average rate allocated to user n, so

$$R(n) = \sum_t \pi_t(n) DRC(n). \tag{3.31}$$

Therefore, the network utility maximization problem [62] is

$$\max \sum_n \log(\sum_t \pi_t(n) DRC(n)) \tag{3.32}$$

$$\text{s.t. } \sum_n \pi_t(n) \leq 1, \quad \forall t, \pi_t(n) \geq 0, \quad \forall n, t. \tag{3.33}$$

Using Lagrange multipliers, we obtain

$$\sum_n \log(\sum_t \pi_t(n) DRC(n)) - \sum_t \lambda_t (\sum_n \pi_t(n) - 1). \tag{3.34}$$

So the optimal solution satisfies

$$\frac{DRC(n)}{R^*(n)} - \lambda^* = 0 \quad \text{if} \quad \pi_t^*(m) > 0, \tag{3.35}$$

Asymptotically, the PF preserves the proportional fairness between the eMBB users, i.e.,

$$\lim_{t \to \infty} \frac{R_t(1)}{DRC(1)} = \cdots = \lim_{t \to \infty} \frac{R_t(n)}{DRC(n)}. \tag{3.36}$$

So, our joint allocation approach preserves the proportional fairness between the eMBB users.

Chapter 4

MULTI-ACCESS POINT COORDINATION FOR NEXT-GEN WI-FI NETWORKS AIDED BY DEEP REINFORCEMENT LEARNING

4.1 Motivation

Orthogonal frequency-division multiple access (OFDMA) adopted in 802.11ax has significantly enhanced the medium access control (MAC). OFDMA works on top of the legacy carrier-sense multiple access with collision avoidance (CSMA/CA) to provide extra features when access point (AP) contends for channel access, e.g., the use of trigger frame helps control the uplink transmission of stations [70] for greater efficiency in a single basic service set identifier (BSS-ID). The overlap of APs is defined as the intersection of their cells and operating frequency bands. In such a cluster of overlapping AP, channel access collision happens more frequently as the number of APs increases, especially when each AP has no prior knowledge of other APs' channel accessing policies. Thus, enabling some degree of collaboration among neighboring APs will permit more efficient utilization of the limited time and frequency resources, i.e., lower collision probability, higher network throughput. To this end, the next-generation standard 802.11be (Wi-Fi 7) introduces some additional features such as multi-AP coordination to further improve aggregate throughput in dense overlapping layout scenarios [71].

The emphasis of EHT WG is on *aggregate throughput in dense networking scenarios* and hence - building on the numerous physical layer (PHY) advances made in 802.11ac/ax - notably new control frames for coordination among APs that requires information exchange among the APs belonging to the coordinated AP set. Besides, the new 6 GHz bands opened up in the US and Europe are new green fields for the future Wi-Fi 7 standard, which gives

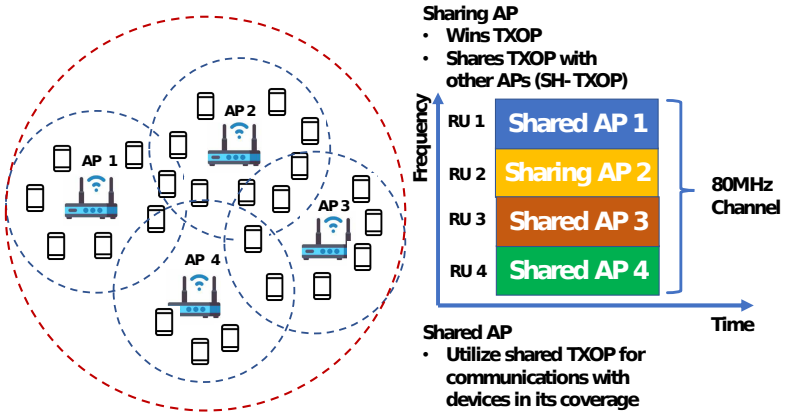


Figure 4.1: Multi-AP Coordination with SH-TXOP: 4 APs share a 80 MHz universal channel. Each AP has its own coverage of stations (STAs) and operates on its own primary channel. Each resource unit (RU) represents a 20 MHz channel.

more freedom in architecture and protocol design. In Fig.4.1, an example of the current 11be structure with Shared Transmission Opportunity (SH-TXOP) operation is introduced [72]. Four APs simultaneously operate on the shared 80-MHz bandwidth. Distributed coordination function (DCF) applied in SH-TXOP allows all APs to contend for channel access. In this example, AP 2 successfully gains the TXOP and becomes a sharing AP. AP 2 then collects information about the channel status and the traffic backlog from shared APs in the candidate shared AP set. Afterwards, the sharing AP will share the wide-band TXOP with the shared APs. Each AP must have a primary channel to operate in the dense overlapping network so that it can contend for TXOPs for the communication within its own coverage. Note that primary channels allocated to different APs are not necessarily the same. In such a scheme considering collaboration, there is no frequency overlap among four channels because each AP operates within its own allocated channel independently. The bandwidth of each channel occupied by a shared AP or sharing AP is 20 MHz.

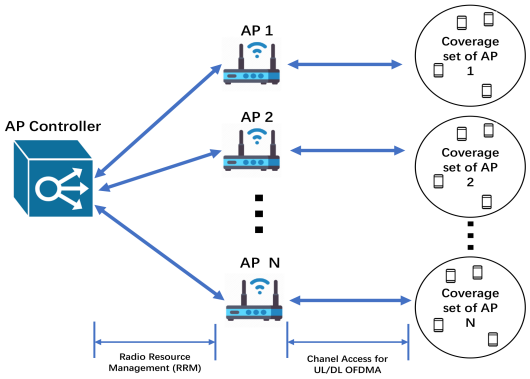


Figure 4.2: Proposed Architecture for Multi-AP Coordination.

4.1.1 Multi-AP Coordination Architecture and Related Work

DCF with CSMA/CA is a traditional MAC protocol for channel access in Wi-Fi networks. It has a long history of analysis using Markov models [73,74] developed originally for a single (isolated) cell (e.g., single home networks) with saturated nodes.

In [75], a novel multi-AP coordination transmission scheme is proposed for 802.11be. TXOP in the proposed scheme is acquired after the AP completes its backoff procedure. Then, it performs a wide-band transmission if the secondary channel is idle during the point coordination function (PCF) inter-frame spacing (PIFS). This contention-based method for TXOP is similar to DCF Request-to-Send/Clear-to-Send (RTS/CTS) [74]. Then, the sharing AP sends an Announcement Trigger Frame (ATF) to the shared APs to allocate all 20 MHz channels, including the duration of TXOP and the scheduled channel information. This proposed scheme also aligns with Coordinated OFDMA (C-OFDMA). That is, each shared AP utilizes partial TXOP assigned by the sharing AP for its uplink/downlink (UL/DL) OFDMA transmission with its associated STAs.

As the backoff procedure for granting the wide-band TXOP is performed solely by the initiating AP, all responding APs should terminate the entire transmission sequence before the TXOP duration, indicated by the received ATF. However, this leads to profound performance loss as the number of APs grows larger. In [75], only 2 APs operate on a 40 MHz

channel. The backoff procedure not followed by ACK can lead to the following situation: more than one AP may think it has won the TXOP and start sending ATF, which will lead to a collision. Since there is no feedback such as ACK, in the end, the whole TXOP is wasted because more than one AP send different RU assignment in ATF and any AP will obtain confusing allocation scheme. As a result, the collision probability increases significantly with the increasing number of APs. Moreover, it is challenging to design a feedback mechanism such as ACK for the TXOP contention method. Responding to the sharing AP by all shared APs is a huge burden to the system performance because any failure reception of the ATF to any AP will lead to re-transmission.

A novel system architecture is thereby proposed for multi-AP coordination in 802.11be to decrease the collision probability of channel access, as is shown in Fig.4.2. The centralized AP controller (APC) implements channel configuration, i.e., assigning primary channels to all APs with consideration of proportional fairness (PF). Under this system architecture, APs do not need to contend for the wide-band TXOP, and ACK is much easier to design. The core function provided by APC is called radio resource management (RRM) [76], which automatically monitors traffic, capacity, and reliability of operating APs. RRM can periodically reconfigure the 802.11 networks for best efficiency by performing functions such as radio resource monitoring and dynamic channel assignment. Each AP contends for TXOP on the assigned primary channel which is further utilized for the communications with its associated STAs by UL/DL OFDMA.

The suffering from the traditional DCF characterized by collision probability in dense overlapping networks also prompts the applications of state-of-the-art machine learning techniques. Deep reinforcement learning (DRL) is a machine learning technique that enables an agent to take actions in an environment aiming to maximize the cumulative rewards. Reinforcement learning (RL) principles have shown potential on optimizing resource allocation in various aspects in wireless communication, see [77–80]. Nonetheless, the application of DRL in wireless networks must be made wisely, as the network utility unavoidably oscillates due to the unstable nature of RL [81]. Moreover, in the overlapping multi-AP network, the network

throughput is affected by the different channel qualities, the number of users, etc. Therefore, the designed AP coordination algorithm should also account for those factors. Adversarial RL-based method [82] is proposed as the solution to single-band multi-AP coordination in 11be. Deep Q-network (DQN) is investigated as an enhancement (higher utility) for CSMA in heterogeneous networks in which more than one multiple access protocol coexist [83, 84]. However, this work assumes that all stations run on the same frequency band. As a more ambitious study, multiple-agent deep learning multiple access with imperfect transmission feedback [85] is studied. The main objective in [85] is to recover the lost transmission feedback between AP and STA due to imperfect channel condition. In [86], multi-channel access for multiple STAs and one AP based on deep reinforcement learning is investigated. This chapter concentrates on that all mobile users utilize improved channel access approaches to communicate with one AP. Our work differs from [85, 86] by considering the throughput maximization problem for multiple overlapping and collaborating APs following 11be standards in which AP is in charge of channel access for TXOP. In [7], we consider a single channel access problem for the communication between only one AP and multiple STAs. As an extended version to [7], we propose in this chapter a novel system architecture that consists a centralized APC with a PF solution to multi-channel allocation problem.

By contrast, our work also differs from the above research work by investigating a new multi-band multi-AP coordination network with APC based on IEEE 802.11be, and our proposed protocol simultaneously enables TXOP contentions at different frequency bands.

4.1.2 Contribution

This chapter proposes a novel coordinated multi-AP architecture and a corresponding channel access mechanism aligning with IEEE 802.11be to maximize the aggregate network throughput while preserving fairness among APs. The major contributions of this chapter are listed as below:

- We propose a multi-AP system with APC as well as formulate a dynamic resource

allocation and channel access optimization problem. The resource allocation process is considered as a Markov decision process (MDP). We choose the previous observation of channels and actions as the state, transmission at a channel as the action, and successful/unsuccessful transmissions at multiple channels as positive/negative rewards.

- Deep reinforcement learning channel access (DLCA) protocol is proposed. For each AP in the coordinated multi-AP set, DLCA is deployed to contend for channel access. The first AP winning the contention gains the TXOP on its primary channel. The First-Order Model-Agnostic Meta-Learning (FOMAML) is then applied to DLCA to enhance the overall performance. We also develop a greedy algorithm to maintain PF among APs.

- Simulation results show that the performance of DLCA protocol is verified to have strong stability and outperform baselines such as SH-TXOP and RTS/CTS in terms of the network throughput as well as the network utility in dense overlapping Wi-Fi networks.

4.2 Review: IEEE 802.11 DCF basic and RTS/CTS

This section briefly introduces two packet mode channel access protocols as follows:

- *Distributed coordination function (DCF) basic*: For stations employing DCF basic method [74], they contend for the channel using a CSMA mechanism with collision avoidance (CSMA/CA). Suppose a station wants to access the channel. It waits until the channel is sensed idle for a distributed inter-frame space (DIFS). Then, a backoff process is initiated. Backoff intervals are slotted, and the discrete backoff time is uniformly distributed in the range $[0, W-1]$, where W is defined as the contention window size, and CW_{min} represents the minimum contention window. The backoff counter is utilized for station to decide whether to access channel at the current time slot. The backoff counter value is initialized by uniformly choosing an integer from the

range $[0, W-1]$. Then, it is decremented by one at the end of each idle slot. Note that the backoff counter will be frozen when a packet transmission is detected on the channel and will be reactivated until the channel is sensed idle again for a DIFS period. The station contends for channel when its backoff counter reaches zero. The ACK follows after the completion of the transmission unless collision happens. If collision happens, contention window size W is doubled after each unsuccessful transmission, up to a maximum value $CW_{max} = 2^m CW_{min}$, where m represents the largest times the contention window size can be doubled.

- *DCF Request-to-Send/Clear-to-Send (RTS/CTS)*: For stations employing DCF RTS/CTS method, station transmits a short frame of RTS to AP after the backoff counter is decremented to zero. When AP detects an RTS frame, it responds, after a short inter-frame space (SIFS), with a CTS frame. The station can only access the channel if the CTS frame is correctly received. The RTS and CTS frames also carry the information of the data duration to be transmitted. This information can be heard by any listening stations, which can then update a network allocation vector (NAV) containing the information that the duration of the channel being busy. Therefore, a station can suitably delay further transmission by detecting just one frame among the RTS and CTS frames and thus avoid collision. The major difference between DCF basic and DCF RTS/CTS is that DCF RTS/CTS will send a RTS and decide to access chanel only after a CTS is received from the AP. DCF basic, on the other hand, contends for channel without sending RTS. Hence, when collision happens, DCF basic wastes a whole data transmission duration while DCF RTS/CTS only wastes a RTS/CTS duration. Hence, DCF RTS/CTS mechanism is very effective in terms of network throughput. involved in the contention process [74].

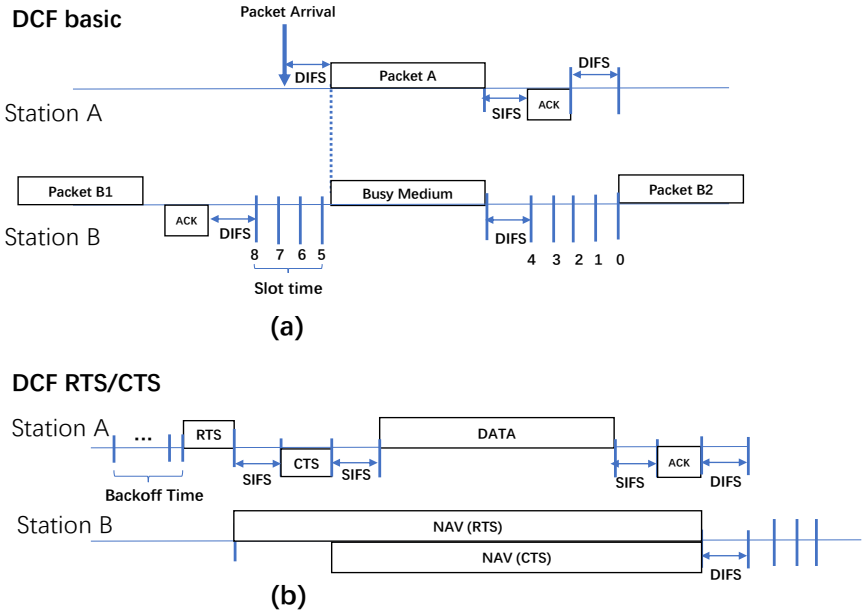


Figure 4.3: IEEE 802.11 DCF basic and RTS/CTS.

4.3 System Model and Problem Formulation of DLCA

In this section, we firstly introduce the multi-AP network with APC. Then the DCF RTS/CTS is introduced and the novel DLCA is proposed. We formulate each AP's dynamic resource allocation and channel access optimization problem as MDP.

4.3.1 Proposed Multi-AP network

In 802.11be system, the aggregation of 5 and 6 GHz spectrum allows simultaneous operation on different bands or channels (orthogonal frequency resource allocation). Our proposed network model aligning with 11be protocol is shown in Fig.4.2. The coordinated multi-AP set is defined as $\mathcal{N} = \{1, \ldots, n, \ldots, N\}$, and $\mathcal{F} = \{1, \ldots, f, \ldots, F\}$ denotes the available orthogonal channel set. Each AP can be allocated with a different channel from the available channel sets. Suppose, at the t^{th} contention for TXOP, AP n observes the channel states of its allocated primary channel (the f^{th} channel of 20 MHz), which yields the observation

Table 4.1: Main Acronyms

TXOP	Transmission Opportunity
SH-TXOP	Shared Transmission Opportunity
STA	Station
RU	Resource Unit
DIFS	Distributed Inter-Frame Space
SIFS	Short Inter-Frame Space
RTS/CTS	Request-to-Send/Clear-to-Send

vector $o_t^n(f) \in \{0, 1\}$ where 0 and 1 denote the IDLE and BUSY channel state, respectively. Action vector is denoted as $a_t^n(f) \in \mathcal{A} \triangleq \{0, 1\}$ where $a_t^n(f) = 1$ represents AP n contends for the f^{th} channel at t^{th} contention for TXOP and $a_t^n(f) = 0$ represents AP n does not contend for the f^{th} channel at t^{th} contention for TXOP. A successful transmission occurs if a sole AP occupies a TXOP. In the following section, the action and observation vector are simplified as a_t^n and o_t^n, respectively, because the f^{th} channel is implicitly linked to AP n after APC has made the channel allocation decision.

4.3.2 Channel Access Mechanism

This section briefly introduces three packet mode channel access protocols that can be potentially utilized in our proposed multi-AP network with APC, including our proposed DLCA protocol. They are described as follows:

- *Distributed coordination function (DCF) basic [74]*: Suppose an AP wants to occupy the TXOP on its primary channel. It waits until the channel is sensed idle for a distributed inter-frame space (DIFS). Then, a backoff process is initiated. Backoff intervals are slotted, and the discrete backoff time is uniformly distributed in the range

$[0, W-1]$, where W is defined as the contention window size, and CW_{min} represents the minimum contention window. The backoff counter is utilized for AP to decide whether to access channel at the current time slot. The backoff counter value is initialized by uniformly choosing an integer from the range $[0, W-1]$. Then, it is decremented by one at the end of each idle slot. Note that the backoff counter will be frozen when a packet transmission is detected on the channel and will be reactivated until the channel is sensed idle again for a DIFS period. The AP contends for TXOP when its backoff counter reaches zero. The ACK follows after the completion of the TXOP unless collision happens. If unsuccessful TXOP happens, contention window size W is doubled after each unsuccessful transmission, up to a maximum value $CW_{max} = 2^m CW_{min}$, where m represents the largest times the contention window size can be doubled.

- *DCF Request-to-Send/Clear-to-Send (RTS/CTS)*: AP transmits a short frame of RTS to APC after the backoff counter is decremented to zero. When APC detects an RTS frame, it responds, after a short inter-frame space (SIFS), with a CTS frame. The AP can only occupy the TXOP of its primary channel if the CTS frame is correctly received. The RTS and CTS frames also carry the information of the TXOP duration to be transmitted. This information can be heard by any listening AP, which can then update a network allocation vector (NAV) containing the information that the duration of the channel being busy. Therefore, an AP can suitably delay further transmission by detecting just one frame among the RTS and CTS frames and thus avoid collision. The major difference between basic and RTS/CTS is that RTS/CTS will send a RTS and decide to contend for the TXOP only after a CTS is received from the APC. DCF basic, on the other hand, contends for TXOP without sending a RTS. Hence, when collision happens, DCF basic wastes a whole TXOP duration while DCF RTS/CTS only wastes a RTS/CTS duration. It is noteworthy that TXOP can take up to 8.16 ms and RTS/CTS only take up to 0.4 ms. Hence, DCF RTS/CTS mechanism is very effective in terms of network throughput, especially for large data load in TXOP, as it

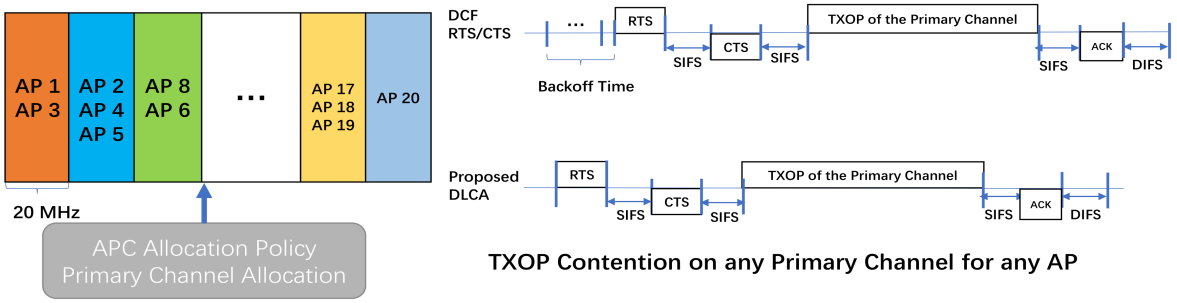


Figure 4.4: The primary channels of APs are assigned by the APC. F channels from 5 and 6 GHz bands are available for IEEE 802.11be. Then APs in each 20 MHz band contends for TXOP.

reduces the average number of wasted time slots involved in the contention process [74].

- *Deep reinforcement learning channel access (DLCA)*: In the DLCA protocol, APC periodically assigns the primary channel to each AP considering PF (as formulated in section 4.4.3). Then, each AP senses the channel and obtains an observation from its primary channel environment, indicating the channel is BUSY or IDLE. Based on the observed results, each AP implements inference regarding the next action utilizing its trained deep Q learning model to maximize the network throughput in its coverage (as formulated in section 4.4.1). It is noteworthy that once the AP employing DLCA decides to contend for TXOP, the similar protocol to DCF RTS/CTS in Fig.4.4 is followed. The only difference is that DLCA has no backoff time, and it transmits RTS as long as it determines to contend for TXOP, which makes the Coordinated-OFDMA [87] possible because each contention process has a constant duration and can be visualized as a time slot (summation of RTS/CTS, TXOP length, and ACK). For each time slot, either one of APs wins the TXOP, or all APs that send the RTS do not receive CTS from APC, leading to Time_Out which indicates that RTS was not approved by APC.

In Fig.4.4, we can have either DCF RTS/CTS or DLCA as the packet mode channel access method. If the DCF basic method is utilized, the transmission result will only be known to the AP until the TXOP duration is finished. The data load within the TXOP is much larger than the traditional scenario in which the DCF basic method is applied, leading to intolerable performance loss [74].

4.3.3 Maximum Achievable Data Rate

Each AP's action $\mathbf{a}_t^n$ is directly related to the throughput of its coverage. The more TXOPs each AP gains, the higher throughput is reached. However, the number of successful TXOP contention is not the only factor to the throughput. The channel conditions between each AP and its associated STAs on different frequency bands are different, and they also vary over time. In this work, we consider that the overall spectral efficiency of AP on its primary channel can be obtained by taking the average of the individual spectral efficiencies between AP n and all the associated STAs. The channel spectral efficiency is assumed to be known by APC and denoted as $\mathbf{C}_t^n \in \mathcal{R}^F$ (bit/s/Hz). The spectral efficiency $\mathbf{C}_t^n(f)$ represents the maximum data that AP n can achieve at time slot t on channel f.

4.3.4 Access Point Model

Define a Markov decision process (MDP) for an AP over a finite state space $\mathcal{S} \in \{0,1\}^{2L} \times \mathcal{Z}$, where L denotes the state size. The finite state space $\mathcal{S}$ is a set that contains concatenations of observation vectors, action vectors, and $c_t^n \in \mathcal{Z}$ that represents the total number of APs contending for TXOPs in the AP n's operating frequency channel (including AP n itself), i.e., $\mathbf{s}_{t+1}^n = [a_{t-L+1}^n, o_{t-L+2}^n, \ldots, a_t^n, o_{t+1}^n, c_t^n]$. The transition function $\delta(\mathbf{s}_t^n, \mathbf{s}_{t+1}^n; a_t^n)$ denotes the probability that the state $\mathbf{s}_t^n$ transfers to the state $\mathbf{s}_{t+1}^n$ after taking action a_t^n. $r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) \in \mathcal{R}$ denotes the reward of AP n at t^{th} TXOP contention results from its state-action-state pair $\left(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n\right)$. The accumulated discounted reward $R_t^n \in \mathcal{R}$ for AP n

can be expressed as

$$R_t^n = \sum_{k=0}^{\infty} \gamma^k r(\mathbf{s}_{t+k}^n, a_{t+k}^n, \mathbf{s}_{t+k+1}^n), \tag{4.1}$$

where $\gamma \in (0, 1]$ is a discounting factor. The policy of AP n $\pi(n) : \mathcal{S} \rightarrow \mathcal{A}$ is assumed to be stationary, and the decision of the policy only depends on the current state. Hence, each AP aims to solve the following problem:

$$\underset{\pi(n)}{\operatorname{argmax}} \mathbf{E}_\delta[R_t | \mathbf{s}_t^n = \mathbf{s}, a_t^n = a, \pi(n)], \tag{4.2}$$

which is the objective of each AP and the expectation with respect to the transition probability function δ is denoted as $E_\delta[\bullet]$. Since each AP simultaneously takes actions on its primary channel where each action is associated with an objective (maximization of the accumulated reward), this is overall a multi-agent problem. The calculation of the reward corresponding to various state-action pairs is detailed in the next section.

4.3.5 System Reward

Every action made by an AP has corresponding feedback (CTS/Time_Out). The system reward is calculated as follows:

$$r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) = \sum_{l=0}^{L-1} \eta^l y_{t-l}^n, \tag{4.3}$$

where $y_{t-l}^n = 1$ denotes the successful feedback for the action a_{t-l}^n and $y_{t-l}^n = -1$ for an unsuccessful contention, and $\eta \in [0, 1]$ is a factor such that the more recent action is, the more weight it will have in the system reward. The overall reward estimation algorithm is shown in Appendix 4.7.

4.4 DLCA Protocol

In this section, we develop the following steps: a) Q-learning satisfying the Bellman optimality condition is introduced; b) we introduce deep reinforcement learning in which a deep Q network is utilized as a model for the action-value function; c) First-Order Model-Agnostic

Meta-Learning (FOMAML) is applied to enhance the convergence rate and the stability of the deep Q network; d) The greedy algorithm considering PF is proposed for multi-AP coordination.

4.4.1 Q-learning

In this section, we introduce standard Q-learning and ϵ-greedy policy as the foundation for the following deep Q-learning. As defined in the above section, each action a_t^n transfers the current state $\mathbf{s}_t^n$ of AP n to $\mathbf{s}_{t+1}^n$ with reward r_{t+1}^n. The action-value function of AP n is denoted as follows:

$$J(\mathbf{s}_t^n, \{A_t^n\}) \triangleq \mathbf{E}_\delta[\sum_{k=0}^{\infty} \gamma^k r(\mathbf{s}_{t+k}^n, A_{t+k}^n, \mathbf{s}_{t+k+1}^n)|\mathbf{s}_t^n], \tag{4.4}$$

where the action-value function $J(\mathbf{s}_t^n, a_t^n) : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$ outputs the accumulated reward with respect to the state $\mathbf{s}_t^n$ and the corresponding action a_t^n. The optimal value function is defined as:

$$V^*(\mathbf{s}_t^n) = \max_{\mathcal{A}_t^n} J(\mathbf{s}_t^n, \mathcal{A}_t^n), \tag{4.5}$$

where $V^*(\mathbf{s}_t^n)$ can be further written as the Bellman optimality equation:

$$\begin{aligned} &V^*(\mathbf{s}_t^n) = \\ &\max_{a_t^n \in \mathcal{A}} \sum_{\mathbf{s}_{t+1}^n \in \mathcal{S}} \delta(\mathbf{s}_t^n, \mathbf{s}_{t+1}^n; a_t^n)[r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) + \gamma V^*(\mathbf{s}_{t+1}^n)], \end{aligned} \tag{4.6}$$

where $\delta(\mathbf{s}_t^n, \mathbf{s}_{t+1}^n; a_t^n)$ represents the transition probability from state $\mathbf{s}_t^n$ to $\mathbf{s}_{t+1}^n$ after taking action a_t^n. The optimal Q-function can then be expressed as the follows:

$$\begin{aligned} &Q^*(\mathbf{s}_t^n, a_t^n) = \\ &\sum_{\mathbf{s}_{t+1}^n \in \mathcal{S}} \delta(\mathbf{s}_t^n, \mathbf{s}_{t+1}^n; a_t^n)\{r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) + \gamma V^*(\mathbf{s}_{t+1}^n)\}, \end{aligned} \tag{4.7}$$

in which the Q-function is a fixed point of a contraction operator $\mathcal{H}$ [88], defined for a generic function $Q : \mathcal{X} \times \mathcal{A} \to \mathcal{R}$ as the follows:

$$\begin{aligned}(\mathcal{H}Q)(\mathbf{s}_t^n, a_t^n) = \sum_{\mathbf{s}_{t+1}^n \in \mathcal{S}} \delta(\mathbf{s}_t^n, \mathbf{s}_{t+1}^n; a_t^n)\{r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) \\ + \gamma \max_{a_{t+1}^n \in \mathcal{A}} Q(\mathbf{s}_{t+1}^n, a_{t+1}^n)\}.\end{aligned} \tag{4.8}$$

In the case of model-free reinforcement learning, the above Q-function is impossible to obtain since the transition probability is unknown. Hence, the Q-learning algorithm searches the optimal Q-function with samplings from the episodes of the MDP. Then, the Q-learning algorithm utilizes the following updating rule:

$$\begin{aligned}Q(\mathbf{s}_t^n, a_t^n) \leftarrow & Q(\mathbf{s}_t^n, a_t^n) + \beta\{r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) \\ & + \gamma \max_{a_{t+1}^n \in \mathcal{A}} Q(\mathbf{s}_{t+1}^n, a_{t+1}^n) - Q(\mathbf{s}_t^n, a_t^n)\},\end{aligned} \tag{4.9}$$

where the learning rate is denoted by β. While each AP updates $Q(\mathbf{s}_t^n, a_t^n)$, it also makes decisions based on $Q(\mathbf{s}_t^n, a_t^n)$, i.e., choosing the action corresponding to the largest Q-value. For the ϵ-greedy policy, the optimal action is given by

$$a_t^n = \begin{cases} \text{argmax}_{a_t^n} Q(s_t^n, a_t^n), & P = 1 - \epsilon. \\ \text{random action}, & P = \epsilon, \end{cases} \tag{4.10}$$

where ϵ denotes the probability of choosing random action. The greedy policy helps the Q-learning policy to search for more possibilities of actions randomly. It can help the policy converge faster and prevent the policy from being stuck at a sub-optimum. Q-learning is proven to converge to the optimum action-values with probability 1 so long as all actions are repeatedly sampled in all states, and the action-values are represented discretely [89].

Algorithm 4: DLCA Algorithm.

Data: $\boldsymbol{\theta}^n$, $\mathbf{s}_0^n$, and $t = 0$.

while $t \geq 0$ **do**

  if *mod*(t, T) *is not* $T - 1$ **then**

$$a_t^n = \begin{cases} \text{argmax}_{a_t^n} Q(\mathbf{s}_t^n, a_t^n), & P = 1 - \epsilon. \\ \text{random action}, & P = \epsilon \end{cases}$$

    Obtain $r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n)$ according to Algorithm 6 and store the tuple $(\mathbf{s}_t^n, a_t^n, r_t^n, \mathbf{s}_{t+1}^n)$ to the training batch $\mathcal{D}$;

    if *Update* **then**

      1. Sample d_s^n transitions from $\mathcal{D}^n$.

      2. Calculate the target value as follows:
      $v = r_t^n + \gamma \left(\max_{a_{t+1}^n} Q(\mathbf{s}_{t+1}^n, a_{t+1}^n; \boldsymbol{\theta}^n) \right)_Q$.

      3. For each randomly sample tuple in the training batch with d_s samples, update $\boldsymbol{\theta}^n$ with the following gradient descent method:

$$\boldsymbol{\theta}^n \leftarrow \boldsymbol{\theta}^n - \rho \nabla_{\boldsymbol{\theta}^n} L(\boldsymbol{\theta}^n).$$

    end

  else

    4. Each AP send its QNN weights to APC that obtains the global QNN $\boldsymbol{\theta}_g = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\theta}_n$.

    5. Sample d_s^g from $\{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$ and update $\boldsymbol{\theta}_g$ with the following gradient descent method:

$$\boldsymbol{\theta}_g \leftarrow \boldsymbol{\theta}_g + \frac{\rho}{N} \sum_{n=1}^{N} \left[v - Q(s_t^n, a_t^n; \boldsymbol{\theta}^n) \right] \nabla Q(s_t^n, a_t^n; \boldsymbol{\theta}^n).$$

    6. Send $\boldsymbol{\theta}_g$ back to each AP: $\boldsymbol{\theta}^n \leftarrow \boldsymbol{\theta}_g$.

  end

  $t = t + 1$;

end

4.4.2 Gradient descent in Deep Q-Learning

The traditional Q-learning algorithm can be applied to solve for the optimal policy. However, traditional Q-learning is impractical if the dimension of action-state space is large, i.e., the curse of dimensionality [90]; thus, the well-known DQN is proposed in [81] to approximate the action-state Q-value function and the neural network used to achieve the approximation is called Q neural network (QNN).

Each AP is equipped with a QNN which outputs the approximated Q-value $\{Q(\mathbf{s}_t^n, a_t^n; \boldsymbol{\theta}^n) | a_t^n \in \mathcal{A}\}$ given the input state $\mathbf{s}_t^n$ and action a_t^n. The optimal policy is to choose the action with the largest Q-value. Unlike the tabular update for Q-learning in Eq (4.9), the QNN in deep Q-learning can be trained by minimizing prediction errors of $Q(\mathbf{s}_t^n, a_t^n; \boldsymbol{\theta}^n)$ at each AP and time slot, where $\boldsymbol{\theta}^n$ denotes the trainable QNN weights on AP n. After the reward is obtained, the state transfers to $\mathbf{s}_{t+1}^n$. The pair $(\mathbf{s}_t^n, a_t^n, r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n), \mathbf{s}_{t+1}^n)$ then forms a single training sample for QNN and is stored in training set $\mathcal{D}^n$. Please note that we will sample training data d_s^n from the training set $\mathcal{D}^n$ for each update in the training process. Next, we define the prediction loss function of QNN as

$$L(\boldsymbol{\theta}^n) = (v - Q(\mathbf{s}_t^n, a_t^n; \boldsymbol{\theta}^n))^2, \tag{4.11}$$

where $Q(\mathbf{s}_t^n, a_t^n; \boldsymbol{\theta}^n)$ is the output of QNN at time slot t and the approximate value function is defined as

$$v = r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) + \gamma \max_{a_{t+1}^n} Q(s_{t+1}^n, a_{t+1}^n; \boldsymbol{\theta}^n), \tag{4.12}$$

in which the second term $\gamma \max_{a_{t+1}^n \in \mathcal{A}} Q(s_{t+1}^n, a_{t+1}^n; \boldsymbol{\theta})$ is obtained by searching the maximum output of QNN with respect to the selection of action a_{t+1}^n given s_{t+1}^n. Then, we can update the trainable QNN weights using the semi-gradient algorithm [91] as below:

$$\boldsymbol{\theta}^n \leftarrow \boldsymbol{\theta}^n + \rho \left[v - Q(s_t^n, a_t^n; \boldsymbol{\theta}^n) \right] \nabla Q(s_t^n, a_t^n; \boldsymbol{\theta}^n), \tag{4.13}$$

where ∇ is the gradient with respect to $\boldsymbol{\theta}^n$. Moreover, each AP is employed with deep Q-learning to search for the optimal QNN. However, this inevitably results in a performance

loss, for some APs cannot avoid learning aggressive policies to maximize the contention benefits for themselves, and some APs learn conservative policies to avoid collision, especially when AP networks are densely overlapping. Hence, in our proposed protocol, each APs sends its QNN weights to APC that takes the average of the weights of all QNNs as follows:

$$\boldsymbol{\theta}_g = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\theta}^n, \tag{4.14}$$

where $\boldsymbol{\theta}_g$ is denoted as global weight. Minimizing the above equation is equivalent to minimize the summation of all loss functions of QNNs from all APs, i.e.,

$$\sum_{n=1}^{N} L(\boldsymbol{\theta}^n) = \sum_{n=1}^{N}(v - Q(\mathbf{s}_t^n, a_t^n; \boldsymbol{\theta}^n))^2. \tag{4.15}$$

In the meantime, minimizing the above equation implicates the following gradient descent for the summed loss function:

$$\boldsymbol{\theta}_g \leftarrow \boldsymbol{\theta}_g + \frac{\rho}{N}\sum_{n=1}^{N}\left[v - Q(s_t^n, a_t^n; \boldsymbol{\theta}^n)\right]\nabla Q(s_t^n, a_t^n; \boldsymbol{\theta}^n). \tag{4.16}$$

Hence, the global QNN weight $\boldsymbol{\theta}_g$ is equivalently obtained by iterations over the sampled data batch d_s^g collected from all APs' local data set $\{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$, which enables a faster convergence rate and lower loss function value. It is noteworthy that $\boldsymbol{\theta}_g$ is sent back to all APs from APC after global gradient descent completes according to Eq (4.16). Then, $\boldsymbol{\theta}_g$ replaces the previous QNN weights for future training and inference. This method is called First-Order Model-Agnostic Meta-Learning (FOMAML), which can further enhance all APs' models with non-IID local data [92].

Remark 3 *In Algorithm 4, samples are collected from the local data set in step 1. Then, in step 2, target value is calculated. Gradient descent method is implemented in step 3 based on the collected samples and the calculated target value. Step 4, 5, and 6 represent the FOMAML method and is triggered once every T local training loops. It is noteworthy that the Q-value in step 2 can be obtained by looping the action corresponding to the largest Q-value with time complexity of $\mathcal{O}(F)$. The gradient in machine learning is normally computed using*

the back-propagation method [93] as a numerical solution with time complexity of $\mathcal{O}(FM)$. Algorithm 4 is typically executed in batch mode - such that QNN update occurs once per batch to reduce computation load. FOMAML is only triggered every period of T to reduce the communication overhead between the APC and AP. The global QNN $\boldsymbol{\theta}_g$ is trained on APC using global information gathered by APC, i.e., $\{d_s^1, \ldots, d_s^N\}$.

4.4.3 AP Coordination: Greedy Algorithm

In the above section, each AP runs with a deep Q-learning algorithm independently. However, the channel on which each AP should run is not described. In this section, the APC policy that allocates channels to all APs considering PF is proposed.

Denote $\phi_t^n(f)$ as the instantaneous proportional achievable data rate for AP n at time slot t at channel f. We assume the block fading channel condition to explore the convergence property of our proposed algorithm, i.e., $\mathbf{C}_t^n(f) = \mathbf{C}^n(f)$ is assumed to be constant over multiple TXOP slots. Then we have $\phi^n(f) = \frac{\mathbf{C}^n(f)}{\mathbf{n}(f)}$. Denote $\mathbf{x}_t^n(f)$ as the actual data rate of AP n at time slot t at channel f, the allocation scheme for AP n can then be expressed as follows:

$$f^* = \underset{f}{\operatorname{argmax}}\, P_t^n(f), \tag{4.17}$$

where

$$P_t^n(f) = \frac{\phi^n(f)}{\tilde{D}_t^n} \tag{4.18}$$

in which

$$\tilde{D}_t^n = \left(1 - \frac{1}{t}\right) \tilde{D}_{t-1}^n + \frac{\mathbf{1}^T \mathbf{x}_t^n}{t}, \tag{4.19}$$

and $\tilde{D}_t^n$ represents the average throughput of AP n up to time slot t. Note that only one element in $\mathbf{x}_t^n \in \mathcal{R}^F$ is non-zero. After allocating one channel to an AP, the scheduler updates the ratio $P_t^n(f)$ for the next AP's allocation. The proposed greedy algorithm considering PF [62] on APC is specified in Algorithm 2. In Algorithm 2, each AP is allocated with a channel in the while-loop. The average throughput of AP n up to time slot t is calculated.

Algorithm 5: Greedy Algorithm Considering PF on APC.

Data: x_0^n, $\tilde{D}_0^n$, $C^n(f)$, and $n = 0$.

$\mathbf{n}(f) = 0$ for all f;

while $n \leq N$ **do**

$$\tilde{D}_t^n \leftarrow \left(1 - \frac{1}{t}\right) \tilde{D}_{t-1}^n + \frac{\mathbf{1}^T \mathbf{x}_t^n}{t}.$$

$$f^* \leftarrow \arg\max \left\{ \frac{\phi^n(f)}{\tilde{D}_t^n} \right\}$$

where

$$\phi^n(f) = \frac{\mathbf{C}^n(f)}{\mathbf{n}(f)}.$$

$n \leftarrow n + 1$;

$\mathbf{n}(f) \leftarrow \mathbf{n}(f) + 1$;

Allocate AP n to f^{th} channel;

end

Then, the channel is chosen to maintain the current PF. The greedy algorithm can guarantee the asymptotic PF, and the corresponding proof is shown in Appendix 4.8.

Round Robin (RR) and PF have been developed as two common scheduling strategies. Among those, PF is widely considered in wireless networks. Different AP has different average throughput due to the number of previously gained TXOPs and various channel spectral efficiency. The greedy algorithm considering PF exploits these variations by allocating the primary channel to the AP with the best conditions for the upcoming TXOP slot. As a design approach, this approach is superior to RR. In the end, the proposed DLCA protocol is shown in Fig.4.5.

Remark 4 *In Algorithm 7, the computation complexity of the PF scheduling method is*

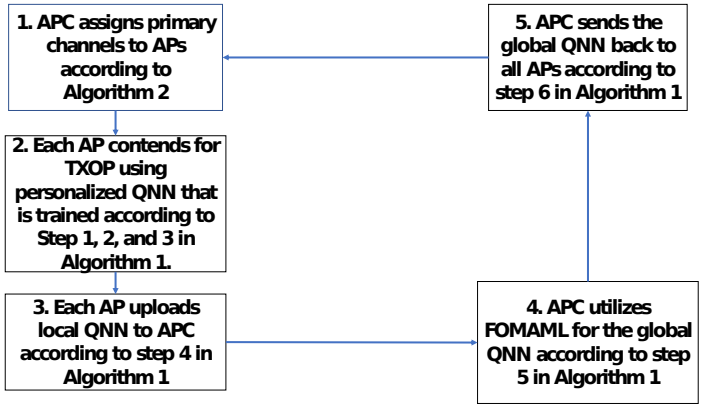


Figure 4.5: Flow Chart of DLCA Protocol: DLCA + Greedy Algorithm + FOMAML.

$\mathcal{O}(NF)$. *Similar to FOMAML, The greedy algorithm is only triggered every period of* T *on APC to reduce the communication burden between the APC and AP. The information of instantaneous data rate* $\mathbf{x}_t^n$*, average data rate* $\tilde{D}_t^n$*, and spectral efficiency* $\mathbf{C}^n(f)$ *can be exchanged between APs and APC through the wired connection such as Light Weight Access Point Protocol (LWAPP), wireless connection such as DCF. FOMAML and the greedy algorithm triggered with a suitable period has negligible impact on the network throughput as long as the QNN is of lightweight.*

4.5 Performance Evaluation

In this section, we present simulation results for a dense overlapping network that implements our DLCA protocol. The performance comparison between the DLCA protocol, SH-TXOP, and DCF RTS/CTS for the overlapping network characterizes the superiority of our DLCA protocol. In the end, we evaluate proportional fairness and stability achieved by the greedy algorithm in the DLCA protocol.

4.5.1 Throughput in Multi-AP and Multi-band Networks

With the FCC opening up the 6 GHz [94] band for unlicensed use for 5G wireless networks, joint operation in 5 and 6 GHz is feasible with orthogonal sub-channels with a bandwidth of 20 MHz. Our simulations consist of a fully overlapping multi-AP network using 5 GHz and 6 GHz bands- for a total of F sub-channels of 20 MHz. The TXOP slot is granted to the shared APs as a multiple of 32 μs, and the maximum amount of time granted is 8.16 ms. The TXOP is thus set as 8.16 ms. In the simulation, AP is assumed to be operating in the saturation mode, i.e., it is always necessary for AP to gain TXOPs because AP needs to communicate with its associated STAs in common Wi-Fi networks continuously. The average value of the spectral efficiency on each channel is 40 Mbps [87], and the spectral efficiency is assumed to be an uniform distribution, i.e., $\mathbf{C}_t^n(f) \sim \mathbf{U}[1,3]$ bit/s/Hz. FOMAML mechanism and the greedy algorithm are triggered with the period of $T = 100$ ms. We conduct Monte-Carlo simulations with 100 independent trials and then take the average of the results.

We firstly consider SH-TXOP without APC [75] as the baseline. In the SH-TXOP protocol, APs share the universal frequency band rather than operating on different sub-channels. After a certain AP wins the wide-band TXOP, it starts to share the TXOP on each 20 MHz sub-channel to other APs based on round robin (RR) method, indicated by Announcement Trigger frame (ATF) that contains the information such as the channel allocation scheme for shared APs. For example, if there are 16 APs and 4 channels and AP 1 wins the TXOP, then AP 1 allocates the first sub-channel as the primary channel for itself. Next, it allocates the second sub-channel to AP 2, and etc. Each AP contends for the wide-band TXOP using IEEE 802.11 DCF basic [74]. The Backoff Window size of DCF basic is $CW_{min} = 32$ and $m = 6$. Note that DCF basic method can be utilized in this scenario. Since the packet size in the process of gaining the sharing opportunity of TXOP is negligible, DCF basic is applicable in such a scenario. However, DCF basic is well known to perform worse than DCF RTS/CTS [74]. To futher enhance the DCF RTS/CTS, a model is proposed in [95] that increases the network throughput by optimizing the initial backoff window size for DCF

RTS/CTS. For this simulation, all APs are equally allocated to a fixed primary channel at the beginning. For example, if there are 16 APs and 4 channels, then AP $1-4$ are allocated to the first channel as their primary channel, and AP $5-8$ are allocated to the second channel as their primary channel and so on. Next, each AP contends for TXOP in its primary channel using DCF RTS/CTS with the optimized initial backoff window size [95]. This method in the simulation is called RTS/CTS. The aggregate network throughput x of the entire network with N APs for this method is expressed as follows:

$$x = \sum_{n=1}^{N} x^n = \sum_{n=1}^{N} \frac{z^n LU}{L+\delta}, \tag{4.20}$$

where L is the information bits in one packet, $\delta = \delta_0 U$ stands for the protocol overhead in the unit of bits. The channel bit rate U can be further written as the product of the sub-channel bandwidth and the link spectral efficiency. The above equation is further explained in Appendix 4.8.1. The system parameters are summarized in Table 4.2.

Table 4.2: System Parameters for Multi-AP Networks

Parameters	**Value**
slot time (μs)	50
SIFS (μs)	28
DIFS (μs)	128
PHY Header (μs)	20
TXOP (μs)	640
CTS_Timeout (μs)	300
ACK_Timeout (μs)	300
Headers (Bytes)	36
ACK (Bytes)	14 + PHY Header
RTS (Bytes)	20 + PHY Header
CTS (Bytes)	14 + PHY Header
ATF (Bytes)	16 + PHY Header

We utilize PyTorch [96] to train QNN for DLCA. The simulations are conducted on a server with a CPU (Intel Core i7-9700k) and a GPU (NVIDIA GeForce GTX 2080Ti) in Python language. QNN is constructed by $h = 5$ fully connected layers with 64 neurons

in each layer, which is illustrated in Fig.4.6. The operation of the QNN starts by taking the state vector as the input. Then, it outputs two Q values corresponding to action - transmission and action - wait respectively. AP decides to transmit if the corresponding Q value is larger and wait otherwise. Table 4.3 lists the hyper-parameter of the deep Q-learning. ReLU (Rectified Linear Unit) defined as

$$f(x) = x^+ = \max(0, x) \tag{4.21}$$

is utilized as the activation function to the input of each neuron in the QNN. Unlike other activation functions such as the sigmoid function, ReLu can help the QNN avoid the vanishing gradient issue because the gradient of $f(x)$ when $x > 0$ is always a constant. Therefore, choosing ReLu can prompt faster learning process and better performance.

Table 4.3: Hyper-Parameters of QNN

Parameters	**Value**
State size	40
Batch size	32
Learning rate ρ	0.001
γ in Eq (4.12)	0.9
ϵ in Eq (4.10)	0.05
η in Eq (4.3)	0.5
Step size of λ	0.1
Gradient descent step size u	0.25

The simulation result of the network throughput is shown in Fig.4.7. We validate the precision of the RTS/CTS model by showing that the simulation results are close to their corresponding theoretical results. Three curves related to the DLCA protocol are plotted respectively. The curve labeled with DLCA reflects the simulation of running distributed deep reinforcement learning on each AP without primary channel allocation from the APC. In this case, each AP only has a fixed primary channel allocation similar to the RTS/CTS simulation. On the other hand, DLCA + greedy method represents the simulation of running distributed deep reinforcement learning on each AP with primary channel allocation decision

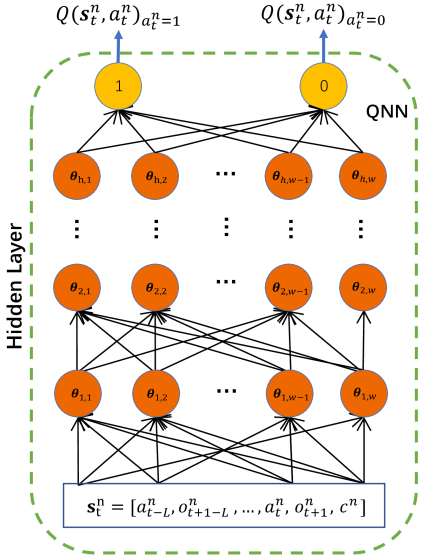
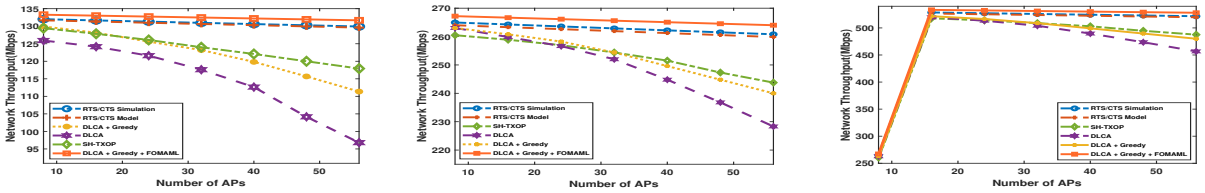


Figure 4.6: QNN: Fully-connected Neural Network.

from APC. Note that APC does not update QNN globally in DLCA + greedy method. In Fig.4.7(b), when the number of APs is small, the performance of DLCA and DLCA + greedy method is better than SH-TXOP. The reason is that when the number of APs is $N = 8$ for both DLCA methods, each AP has its own exclusive primary channel and no collision happens at all, which can be observed from Fig.4.8. However, 8 APs have to contend for the wide-band TXOP for SH-TXOP. The average slots wasted in the collision for DLCA and DLCA + greedy method are zero, not to mention that SH-TXOP has more average IDLE slots per TXOP than DLCA + greedy. As the number of APs increases, SH-TXOP outperforms DLCA and DLCA + greedy. This is because some APs can develop very aggressive TXOP contention policy without the supervision from FOMAML. Note that each AP only wants to maximize its own total reward. Hence, it is possible that APs assigned to one channel are all aggressive and we can view this training process as Prisoner's Dilemma; that is, if one AP does not develop aggressive TXOP contention policy, then it has no chance to get any TXOP forever. Therefore, it is likely that DLCA will lead to high collision probability. However, SH-TXOP and DCF RTS/CTS have backoff counter to avoid collision probability if collision



(a) The Number of Channel is F=4

(b) The Number of Channel is F=8

(c) The Number of Channel is F=16

Figure 4.7: Network Throughput vs Number of APs.

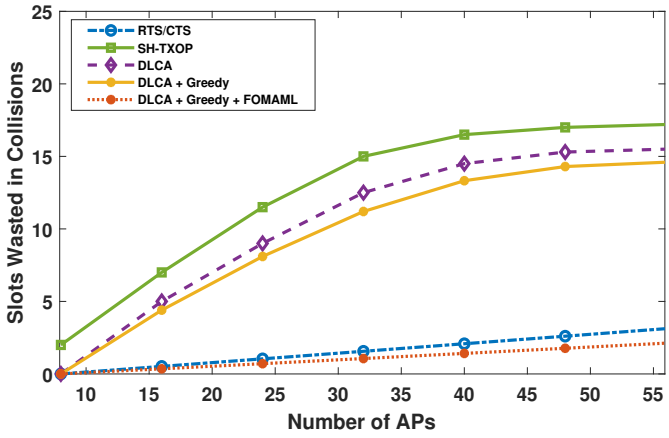


Figure 4.8: Average Collision vs Number of APs. The Number of Channel is $F = 8$.

happens. Hence, as the number of APs increases, average slots wasted in collisions and average IDLE slots per TXOP all increases for DLCA and DLCA + greedy in Fig.4.8 and 4.9, which leads to severe performance loss. Although both RTS/CTS method and DLCA + greedy + FOMAML perform well, DLCA + greedy + FOMAML provides higher network throughput than RTS/CTS method by 3% for the total number of AP ranging from 8 to 56. This is because the overhead remains in RTS/CTS and the backoff window takes up time slots without sending any data packet. One can observe the RTS/CTS method has more average collision slots and IDLE slots from Fig.4.8 and 4.9. In Fig.4.7(c), for the case of 16 channels with 8 APs, only half of the channel resources are utilized. Hence, the network

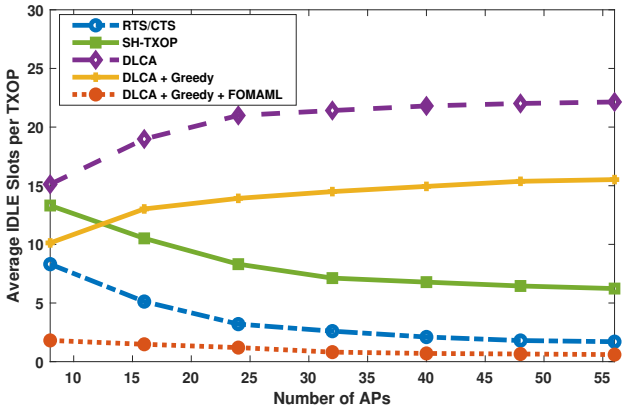


Figure 4.9: Average IDLE vs Number of APs. The Number of Channel is $F = 8$.

throughput grows linear with increasing number of APs at the beginning. In Fig.4.7, DLCA + Greedy + FOMAML outperforms SH-TXOP by 10% when the number of APs is $N = 56$ in average of three cases. The advantage of FOMAML can also be demonstrated in Fig.4.10, during the training process, the network throughput in both methods have large variance in the initial learning phase. However, DLCA + greedy + FOMAML has faster convergence rate and smaller variance. This can be attributed to the fact that each AP's self-training only reaches local optimality, emphasizing the important role of FOMAML as the global optimizer that achieves the necessary AP coordination for throughput maximization.

4.5.2 PF in Multi-AP and Multi-band Networks

In the above section, aggregate network throughput is simulated. However, aggregate network throughput does not reflect the throughput of each AP, leading to a potential issue that the maximum throughput can always be achieved by having the same AP holding the channel, and no fairness exists at all. Hence, fairness must be guaranteed so that each AP in the network can utilize the TXOP for UL/DL communication with associated STAs. The greedy algorithm has been proven to enable PF among APs asymptotically previously. This section implements simulations to study a network utility metric that describes PF and network

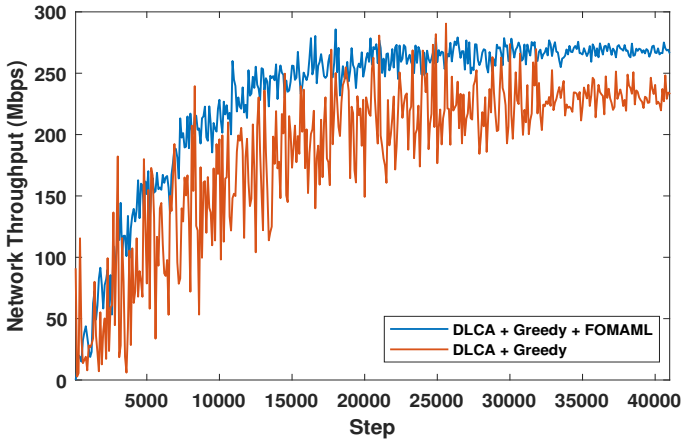


Figure 4.10: Convergence: Throughput vs Training Steps. The Number of Channel $F = 8$, The Number of APs $N = 16$. Each time step is one round of gradient descent in Eq. (4.13).

throughput. Meanwhile, the stability of our proposed algorithm is also investigated.

To be consistent with the notations in Appendix.4.8, the network utility is defined as $\sum_{n=1}^{N} \log(\bar{D}^n)$, where $\bar{D}^n$ is the average data rate of AP n. We simulate the network utility of DLCA + greedy + FOMAML, SH-TXOP. and RTS/CTS for comparison. In Fig.4.11, DLCA protocol performs better than RTS/CTS by 13.8% and SH-TXOP by 28.3% when the number of APs equals $N = 56$. In $\sum_{n=1}^{N} \log(\bar{D}^n)$, the *log* term punishes the AP that has low throughput. Therefore, the network utility demonstrates joint network throughput and PF. We next show the stability of DLCA + greedy + FOMAML in terms of AP's PF ratio, which is expressed as

$$b^n = \frac{\bar{D}^n}{\phi^n}. \tag{4.22}$$

According to Eq (4.28), the closer b^n of each AP is to each other, the better PF is achieved. In Fig.4.12, the PF ratios of 3 APs converge to 0.87, 0.84, and 0.81 respectively after 30000 steps. Then, we exchange the value of spectral efficiency $\mathbf{C}_t^n(f)$ between AP 1 and AP 3 at step 30000 to demonstrate the stability. After a sudden change at step 30000, three curves experience drastic oscillation. Then, we can observe that the curves of AP 1 and AP 3 converge again eventually. This result indicates that the fairness broken by a sudden

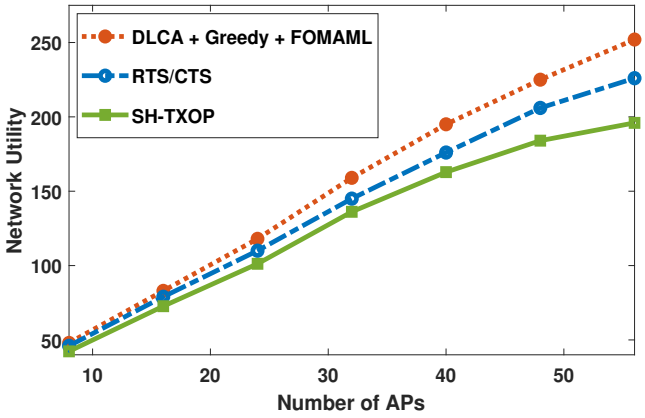


Figure 4.11: Network Utility ($\sum_{n=1}^{N} \log(\bar{D}^n)$) vs Number of APs. The number of channel is $F = 8$.

network change can be restored very quickly. Hence, our greedy algorithm is shown to be robust and efficient.

In the end, we conclude that the design of multi-AP network with APC has a higher upper limit than the multi-AP network without APC in terms of the network throughput. The choice of RTS/CTS or DLCA + greedy + FOMAML is constrained by the hardware and energy cost. For the AP with limited power constraint, one can choose RTS/CTS with lower power consumption but the performance loss, especially the lack of PF, might lead to an unsatisfied user experience. On the other hand, if the power budget is high enough and each AP is able to run light-weight QNN with suitable CPU or GPU, AP can reach higher network throughput, and proportional fairness among APs can be also guaranteed.

4.6 Summary

In this chapter, we propose enhancements to the RRM architecture for dense overlapping Wi-Fi networks that align with the proposed coordinated AP operation in Wi-Fi 7 (802.11be). Specifically, we develop a novel multi-AP coordination system architecture with DLCA protocol. The proposed protocol considers not only the network throughput maximization but

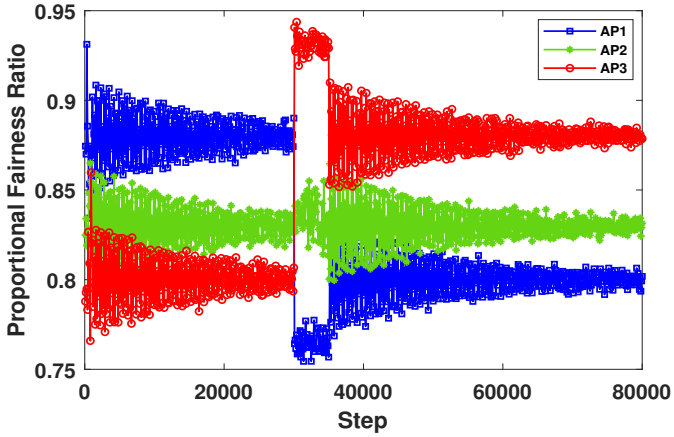


Figure 4.12: PF ratio b^n vs Time Step with Greedy Algorithm. PF ratios of 3 APs out of $N = 18$ APs on $F = 8$ channels are depicted. Each time step is one slot time in Tab. 4.2.

also the proportional fairness among APs. The performance of DLCA related algorithms is then evaluated via simulations and compared with SH-TXOP protocol and RTS/CTS as benchmarks. The numerical results show that DLCA outperforms SH-TXOP and state-of-the-art RTS/CTS with an optimized initial back-off window in terms of network throughput and network utility. Moreover, convergence rate and stability are also demonstrated in the simulation.

This chapter studies the fully overlapping dense Wi-Fi networks. In our future work, we will investigate partially overlapping dense Wi-Fi networks. In such a case, optimization for dynamic AP coordination set, frequency reuse in different coordination set for 802.11be, and coexistence with other protocols will be considered.

4.7 Appendix: Reward Estimation Method

Algorithm 6 is a modified version of Monte Carlo method in [91] that aims to estimate reward and help Q-learning converges faster. For AP n operating in f^{th} frequency channel, we initialize the total reward to be zero in step 1. Then, if the current action $a_t^n(f) = 1$, we use while loop to find all feedback of the transmission action (action value is equal to

1) in the state vector. Suppose one of the feedback is ACK, we add one to the weighted total reward value in step 2 since it is a successful transmission. Otherwise, we minus one to punish the weighted total reward value in step 3. Suppose the current action $a_t^n(f) = 0$, then the total reward value is 1 if busy channel is sensed in step 4 since the AP successfully avoids a potential collision. The total reward value is set to -1 as a punishment if idle channel is sensed in step 5.

Algorithm 6: Reward estimation method.

Data: f, $\mathbf{s}_t^n$, feedback for all actions in $\mathbf{s}_t^n$.
Result: $r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n)$

1. $r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) \leftarrow \mathbf{0}$;
if $a_t^n(f) == \{1\}$ **then**
  $l \leftarrow L$;
  while $l \geq 0$ **do**
    if *the feedback of* $a_{t-l}^n(f) = 1$ *is ACK* **then**
      2. $r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) \leftarrow \eta \times r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) + 1$
    else
      3. $r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) \leftarrow \eta \times r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) - 1$
    end
    $l \leftarrow l - 1$;
  end
else
  if $\mathbf{o}_{t+1}^n(f) == \{1\}$ **then**
    4. $r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) \leftarrow 1$
  else
    5. $r(\mathbf{s}_t^n, a_t^n, \mathbf{s}_{t+1}^n) \leftarrow -1$
  end
end

4.8 Appendix: Proof of Asymptotic Proportional Fairness

We show that the greedy algorithm considering PF maximizes the aggregate throughput while guaranteeing the asymptotic PF on one channel. Denote $p_t^n(f)$ as the probability of the AP n at time slot t being assigned with channel f. Assume the spectral efficiency does not change within t slots, then we have

$$\bar{D}_t^n = \sum_f \sum_t p_t^n(f)\phi^n(f). \tag{4.23}$$

Therefore, the network utility maximization problem [62] is

$$\max \ \sum_n \log(\sum_f \sum_t p_t^n(f)\phi^n(f)) \tag{4.24}$$

$$\text{s.t.} \ \sum_n \sum_f p_t^n(f) \leq 1, p_t^n(f) \geq 0, \ \forall n, f, t. \tag{4.25}$$

Note that the above problem is a convex problem since log function with composition of an affine function still preserves concavity. Applying Lagrange multipliers, we obtain the following:

$$\sum_n \log(\sum_t \sum_f p_t^n(f)\phi^n(f)) - \sum_t \lambda_t(\sum_n \sum_f p_t^n(f) - 1) \tag{4.26}$$

Taking the derivative w.r.t. $p_t^n(f)$, we obtain the optimal solution as

$$\frac{\phi^n}{\bar{D}_t^n} - \lambda^* = 0 \quad \text{if} \quad p_t^n(f) > 0, \tag{4.27}$$

Asymptotically, the PF algorithm helps the AP network to reach the PF, i.e.,

$$\lim_{t \to \infty} \frac{\bar{D}_t^1}{\phi^1} = \cdots = \lim_{t \to \infty} \frac{\bar{D}_t^n}{\phi^n}. \tag{4.28}$$

Hence, allocation method shown in Eq. (4.17) follows the optimal condition of the 11be network utility maximization problem with PF.

4.8.1 Performance of Multi-AP IEEE 802.11 RTS/CTS Networks

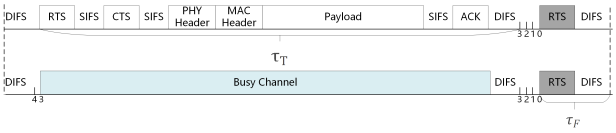


Figure 4.13: Graphic illustration of successful transmission and collision in DCF networks with the RTS/CTS access mechanism.

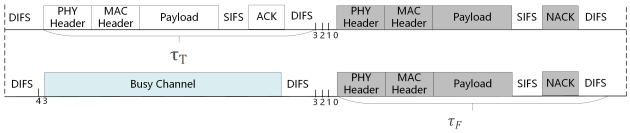


Figure 4.14: Graphic illustration of successful transmission and collision with DCF basic mechanism.

The aggregate network data rate x is the average number of information bits successfully transmitted per second, which is the sum of the average number of information bits that AP n successfully transmits per second x^n

$$x = \sum_{n=1}^{N} x^n, \tag{4.29}$$

The duration to transmit a packet consisting of L information bits is given by $\frac{L}{U} + \delta_0$, where U is the channel bit rate, and δ_0 is the protocol overhead in seconds. With the RTS/CTS mechanism and DCF basic illustrated in Fig. 4.13 and 4.14 respectively, δ_0 is given by

$$\begin{aligned} \delta_0^{RTS} &= \frac{\text{RTS} + \text{CTS} + \text{ACK}}{U_b} + \text{Header} + \text{DIFS} + 3 \times \text{SIFS} \\ \delta_0^{basic} &= \frac{\text{ACK}}{U_b} + \text{Header} + \text{DIFS} + \text{SIFS}, \end{aligned} \tag{4.30}$$

where U_b denotes the basic rate. In the end, the data rate x^n of each BSS i can be expressed as follows:

$$x^n = \frac{Lz^n}{\frac{L}{U} + \delta_0} = \frac{z^n LU}{L + \delta},$$

where $\delta = \delta_0 U$ stands for the protocol overhead in the unit of bits. The channel bit rate U can be further written as the product of the channel bandwidth of each AP and the link spectral efficiency. The details of derivation of z^n is related to parameters of τ_T and τ_F [95]. As for the throughput of Multi-AP IEEE 802.11 DCF basic network without APC, $\tau_T^{DCF} = \tau_T - (RTS + CTS + 2 \times SIFS)$. τ_F^{DCF} is equal to τ_T^{DCF} in basic DCF since there is no RTS/CTS and the collision leads to a waste of whole packet time instead of RTS/CTS with short duration, which is the reason why RTS/CTS can enhance the system throughput.

Chapter 5

CONCLUSION

In this dissertation, we have investigated optimized resource allocation policies for wireless networks, with the unifying theme of enhancing the performance and efficiency of wireless networks by emphasizing various communication protocols and resources across three distinct chapters. By addressing different network scenarios and developing novel approaches and algorithms tailored to these specific cases, we provide a comprehensive framework for optimizing resource allocation policies in wireless networks.

In Chapter 2, we explore joint caching and computing policy optimization for a bidirectional computation task model, focusing on both homogeneous and heterogeneous scenarios. The optimal policy is derived for the homogeneous case, while LR-based methods are proposed for the heterogeneous case. Chapter 3 develops a novel downlink scheduler for URLLC, transforming reliability and latency requirements into an optimal policy formulation and proposing CCCP and greedy algorithms for resource allocation. The scheduler is designed to maintain proportional fairness among eMBB users while maximizing the aggregate data rate. In Chapter 4, enhancements to RRM architecture for dense overlapping Wi-Fi networks were introduced in line with Wi-Fi 7 (802.11be) coordinated AP operation, developing a multi-AP coordination system architecture with the DLCA protocol. This protocol aims to maximize network throughput and maintain proportional fairness among APs.

The results and contributions of this thesis demonstrate the effectiveness of the proposed resource allocation policies and novel algorithms in improving network performance and efficiency across a range of wireless network scenarios. By building upon the unifying theme of optimizing wireless networks with different emphasis on various communication protocols and resources, this work offers valuable insights and solutions for the field of wireless engineering.

Future research includes exploring partially overlapping dense Wi-Fi networks, focusing on optimization for dynamic AP coordination sets, frequency reuse in different coordination sets for 802.11be, and coexistence with other protocols. Additionally, incorporating machine learning techniques and emerging wireless technologies such as massive MIMO, mmWave communication could lead to further advancements in wireless network performance and the development of innovative solutions for next-generation communication systems.

BIBLIOGRAPHY

[1] J. Hecht, "The bandwidth bottleneck that is throttling the internet," *Nature News*, vol. 536, no. 7615, p. 139, Aug. 2016.

[2] E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, June 2017.

[3] 3GPP, "Study on physical layer enhancements for NR ultra-reliable and low latency case (URLLC)," no. TR38.824, Mar 2019.

[4] T. Adame, M. Carrascosa-Zamacois, and B. Bellalta, "Time-sensitive networking in ieee 802.11 be: On the way to low-latency wifi 7," *Sensors*, vol. 21, no. 15, p. 4954, 2021.

[5] 3GPP, "Physical channels and modulation," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.211, Jan 2020.

[6] I. . W. Group *et al.*, "Part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications: Higher-speed physical layer in the 5ghz band," *IEEE Std 802.11*, 1999.

[7] L. Zhang, H. Yin, Z. Zhou, S. Roy, and Y. Sun, "Enhancing WiFi multiple access performance with federated deep reinforcement learning," in *IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*.

[8] H. Yin, L. Zhang, and S. Roy, "Multiplexing urllc traffic within embb services in 5g nr: Fair scheduling," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 1080–1093, 2021.

[9] L. Zhang, Y. Sun, Z. Chen, and S. Roy, "Communications-caching-computing resource allocation for bidirectional data computation in mobile edge networks," *IEEE Transactions on Communications*, vol. 69, no. 3, pp. 1496–1509, 2021.

[10] L. Zhang, H. Yin, S. Roy, and L. Cao, "Multiaccess point coordination for next-gen wi-fi networks aided by deep reinforcement learning," *IEEE Systems Journal*, vol. 17, no. 1, pp. 904–915, 2023.

[11] H. Liu, Z. Chen, and L. Qian, "The three primary colors of mobile systems," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 15–21, Sep. 2016.

[12] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[13] Y. Sun, Y. Cui, and H. Liu, "Joint pushing and caching for bandwidth utilization maximization in wireless networks," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 391–404, Dec. 2018.

[14] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[15] M. Patel, Y. Hu, P. Hede, J. Joubert, C. Thornton, B. Naughton, J. Roldan Ramos, C. Chan, V. Young, S. Jin Tan, D. Lynch, N. Sprecher, T. Musiol, C. Manzanares, U. Rauschenbach, S. Abeta, L. Chen, K. Shimizu, A. Neal, P. Cosimini, A. Pollard, and G. Klas, "Mobile edge computing-introductory technical white paper," *Mobile-Edge Computing (MEC) ISG, European Telecommunications Standards Institute (ETSI)*, Sep. 2014.

[16] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, Mar. 2017.

[17] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017.

[18] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[19] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[20] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[21] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *arXiv preprint arXiv:1908.06849*, 2019.

[22] X. Yang, Z. Chen, K. Li, Y. Sun, N. Liu, W. Xie, and Y. Zhao, "Communication-constrained mobile edge computing systems for wireless virtual reality: Scheduling and tradeoff," *IEEE Access*, vol. 6, pp. 16 665–16 677, Mar. 2018.

[23] X. Yang, Z. Fei, J. Zheng, N. Zhang, and A. Anpalagan, "Joint multi-user computation offloading and data caching for hybrid mobile cloud/edge computing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 018–11 030, Nov. 2019.

[24] C. Liang, Y. He, F. R. Yu, and N. Zhao, "Enhancing video rate adaptation with mobile edge computing and caching in software-defined mobile networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 7013–7026, Oct. 2018.

[25] Y. Hao, M. Chen, L. Hu, M. S. Hossain, and A. Ghoneim, "Energy efficient task caching and offloading for mobile edge computing," *IEEE Access*, vol. 6, pp. 11 365–11 373, Mar. 2018.

[26] M. Chen, Y. Hao, L. Hu, M. S. Hossain, and A. Ghoneim, "Edge-cocaco: Toward joint optimization of computation, caching, and communication on edge cloud," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 21–27, July 2018.

[27] S. Yu, R. Langar, X. Fu, L. Wang, and Z. Han, "Computation offloading with data caching enhancement for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 098–11 112, Sep. 2018.

[28] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Bandwidth gain from mobile edge computing and caching in wireless multicast systems," *submitted to IEEE Trans. Wireless Commun.*, arxiv.org/abs/1901.09738, 2019.

[29] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Feb. 2016.

[30] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communications, caching, and computing for mobile virtual reality: Modeling and tradeoff," *IEEE Trans. on Commun.*, vol. 67, no. 11, pp. 7573–7586, Nov. 2019.

[31] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management science*, vol. 50, no. 12, pp. 1861–1871, Dec. 2004.

[32] Y. Sun, L. Zhang, Z. Chen, and S. Roy, "Communications-caching-computing tradeoff analysis for bidirectional data computation in mobile edge networks," in *IEEE Vehicular Technology Conference (VTC) 2020-Fall, to appear*, 2020.

[33] M. Hifi and L. Wu, "An equivalent model for exactly solving the multiple-choice multi-dimensional knapsack problem," *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 3, no. 3, pp. 43–58, Dec. 2012.

[34] H. A. Le Thi, T. P. Dinh, and H. Van Ngai, "Exact penalty and error bounds in dc programming," *Journal of Global Optimization*, vol. 52, no. 3, pp. 509–535, Sep. 2012.

[35] A. Le Thi Hoai and P. D. Tao, "Solving a class of linearly constrained indefinite quadratic problems by DC algorithms," *Journal of Global Optimization*, vol. 11, no. 3, pp. 253–285, Oct. 1997.

[36] P. V. Amadori and C. Masouros, "Low RF-complexity millimeter-wave beamspace-MIMO systems by beam selection," *IEEE Transactions on Communications*, vol. 63, no. 6, pp. 2212–2223, May 2015.

[37] X. Zhang, L. Chen, J. Qiu, and J. Abdoli, "On the waveform for 5G," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 74–80, Nov. 2016.

[38] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The next generation wireless access technology*. Academic Press, 2020.

[39] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 166–174, 2017.

[40] A. Anand, G. De Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," in *Proc. IEEE INFOCOM*, April 2018.

[41] R. Basukala, H. A. M. Ramli, and K. Sandrasegaran, "Performance analysis of EXP/PF and M-LWDF in downlink 3GPP LTE system," in *2009 1st Asian Himalayas Int. Conf. on Internet*, 2009, pp. 1–5.

[42] M. Sybis, K. Wesolowski, K. Jayasinghe, V. Venkatasubramanian, and V. Vukadinovic, "Channel Coding for Ultra-Reliable Low-Latency Communication in 5G Systems," in *Proc. IEEE 84th Veh. Technol. Conf.(VTC)*, Sept. 2016.

[43] 3GPP, "Study on new radio (NR) access technology," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.912, Jun 2018.

[44] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 124–130, 2018.

[45] V. Kumar and N. B. Mehta, "Modeling and analysis of differential CQI feedback in 4G/5G OFDM cellular systems," *IEEE Trans. Wireless Commun.*, Apr. 2019.

[46] M. Angjelichinoski, K. F. Trillingsgaard, and P. Popovski, "A statistical learning approach to ultra-reliable low latency communication," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 5153–5166, 2019.

[47] 3GPP, "Physical layer procedures for data," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.214, Mar 2019.

[48] A. A. Esswie and K. I. Pedersen, "Opportunistic spatial preemptive scheduling for URLLC and eMBB coexistence in multi-user 5G networks," *IEEE Access*, vol. 6, pp. 38 451–38 463, 2018.

[49] R. Kassab, O. Simeone, P. Popovski, and T. Islam, "Non-orthogonal multiplexing of ultra-reliable and broadband services in fog-radio architectures," *IEEE Access*, vol. 7, pp. 13 035–13 049, 2019.

[50] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.

[51] A. Anand and G. De Veciana, "Resource allocation and HARQ optimization for URLLC traffic in 5G wireless networks," *IEEE IEEE J. Sel. Areas Commun.*, Nov Oct. 2018.

[52] C. She, C. Yang, and T. Q. S. Quek, "Joint uplink and downlink resource configuration for ultra-reliable and low-latency communications," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2266–2280, 2018.

[53] S. E. Elayoubi, P. Brown, M. Deghel, and A. Galindo-Serrano, "Radio resource allocation and retransmission schemes for URLLC over 5G networks," *IEEE J. Sel. Areas Commun.*, Apr. 2019.

[54] P. Korrai, E. Lagunas, S. K. Sharma, S. Chatzinotas, A. Bandi, and B. Ottersten, "A RAN resource slicing mechanism for multiplexing of eMBB and URLLC services in OFDMA based 5G wireless networks," *IEEE Access*, vol. 8, pp. 45 674–45 688, 2020.

[55] P. K. Korrai, E. Lagunas, S. K. Sharma, S. Chatzinotas, and B. Ottersten, "Slicing based resource allocation for multiplexing of eMBB and URLLC services in 5G wireless networks," in *Proc. IEEE 24th Int. Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2019, pp. 1–5.

[56] A. K. Bairagi, M. S. Munir, M. Alsenwi, N. H. Tran, and C. S. Hong, "A matching based coexistence mechanism between eMBB and URLLC in 5G wireless networks," in *Proc. 34th ACM/SIGAPP Symposium on Applied Computing*, ser. SAC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2377–2384.

[57] A. Anand, G. de Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 477–490, 2020.

[58] S. R. Pandey, M. Alsenwi, Y. K. Tun, and C. S. Hong, "A Downlink Resource Scheduling Strategy for URLLC Traffic," in *Proc. IEEE Int. Conf. on Big Data and Smart Computing (BigComp)*, Feb. 2019.

[59] M. Alsenwi, S. R. Pandey, Y. K. Tun, K. T. Kim, and C. S. Hong, "A chance constrained based formulation for dynamic multiplexing of eMBB-URLLC traffics in 5G new radio," in *Proc. Int. Conf. on Information Networking (ICOIN)*, Jan. 2019.

[60] C. Wengerter, J. Ohlhorst, and A. G. E. von Elbwart, "Fairness and throughput analysis for generalized proportional fair frequency scheduling in OFDMA," in *Proc. IEEE 61st Veh. Technol. Conf.(VTC)*, vol. 3, 2005, pp. 1903–1907 Vol. 3.

[61] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[62] R. Srikant and L. Ying, *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

[63] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[64] S. Lien, S. Shieh, Y. Huang, B. Su, Y. Hsu, and H. Wei, "5G New Radio: Waveform, frame structure, multiple access, and initial access," *IEEE Comm. Magazine*, vol. 55, pp. 64–71, June Jun. 2017.

[65] Samsung, "4G-5G Interworking White Paper," Tech. Rep., Jul 2017.

[66] T. Fehrenbach, R. Datta, B. Göktepe, T. Wirth, and C. Hellge, "URLLC services in 5G low latency enhancements for LTE," in *Proc. IEEE 88th Veh. Technol. Conf.(VTC)*, 2018, pp. 1–6.

[67] 3GPP, "Feasibility study for further advancements for E-UTRA (LTE-Advanced)," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR36.912, Jul 2020.

[68] B. K. Sriperumbudur and G. R. Lanckriet, "On the convergence of the concave-convex procedure," in *Proc. 22nd Int. Conf. Neural Information Processing Systems*. Curran Associates Inc., 2009, pp. 1759–1767.

[69] M. B. Cohen, Y. T. Lee, and Z. Song, "Solving linear programs in the current matrix multiplication time," in *Proc. 51st annual ACM SIGACT symposium on theory of computing*, 2019, pp. 938–942.

[70] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A Tutorial on IEEE 802.11ax High Efficiency WLANs," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 197–216, Sep. 2019.

[71] J. Liu, T. Pare, Y. Seok, J. Wang, F. Hsu, and J. Yee, "Multi-AP Enhancement and Multi-Band Operations," Mediatek Inc., Tech. Rep. IEEE 802.11-18/1155r1, Jun. 2018.

[72] S. Naribole, W. B. Lee, K. Srinivas, R. Duan, and A. Ranganath, "Shared TXOP Protocol," Samsung Inc., Tech. Rep. IEEE 802.11-20/0277r1, Mar. 2020.

[73] B. Kwak, N. Song, and L. Miller, "Performance Analysis of Exponential Backoff," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 343–355, Apr. 2005.

[74] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[75] W. Ahn, "Novel Multi-AP Coordinated Transmission Scheme for 7th Generation WLAN 802.11 be," *Entropy*, vol. 22, no. 12, p. 1426, 2020.

[76] L. Sequeira, J. L. de la Cruz, J. Ruiz-Mas, J. Saldana, J. Fernandez-Navajas, and J. Almodovar, "Building an SDN Enterprise WLAN Based on Virtual APs," *IEEE Communications Letters*, vol. 21, no. 2, pp. 374–377, 2017.

[77] Y. Wang, X. Li, P. Wan, and R. Shao, "Intelligent Dynamic Spectrum Access Using Deep Reinforcement Learning for VANETs," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 15 554–15 563, Feb. 2021.

[78] H. Song, L. Liu, J. Ashdown, and Y. Yi, "A Deep Reinforcement Learning Framework for Spectrum Management in Dynamic Spectrum Access," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 208–11 218, Jan. 2021.

[79] W. Ahsan, W. Yi, Z. Qin, Y. Liu, and A. Nallanathan, "Resource Allocation in Uplink NOMA-IoT Networks: A Reinforcement-Learning Approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 8, pp. 5083–5098, Mar. 2021.

[80] H. Ding, F. Zhao, J. Tian, D. Li, and H. Zhang, "A Deep Reinforcement Learning for User Association and Power Control in Heterogeneous Networks," *Ad Hoc Networks*, vol. 102, p. 102069, May 2020.

[81] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level Control through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[82] Y. Kihira, Y. Koda, K. Yamamoto, T. Nishio, and M. Morikura, "Adversarial Reinforcement Learning-based Robust Access Point Coordination against Uncoordinated Interference," *arXiv preprint arXiv:2004.00835*, 2020.

[83] Y. Yu, T. Wang, and S. Liew, "Deep-reinforcement Learning Multiple Access for Heterogeneous Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1277–1290, Mar. 2019.

[84] Y. Yu, S. C. Liew, and T. Wang, "Non-uniform Time-step Deep Q-network for Carrier-sense Multiple Access in Heterogeneous Wireless Networks," *IEEE Transactions on Mobile Computing*, Apr. 2020.

[85] ——, "Multi-Agent Deep Reinforcement Learning Multiple Access for Heterogeneous Wireless Networks with Imperfect Channels," *IEEE Transactions on Mobile Computing*, pp. 1–1, Feb. 2021.

[86] X. Ye, Y. Yu, and L. Fu, "Multi-channel Opportunistic Access for Heterogeneous Networks Based on Deep Reinforcement Learning," *IEEE Transactions on Wireless Communications*, pp. 1–1, July 2021.

[87] D. Lopez-Perez, A. Garcia-Rodriguez, L. Galati-Giordano, M. Kasslin, and K. Doppler, "IEEE 802.11be Extremely High Throughput: The Next Generation of Wi-Fi Technology Beyond 802.11ax," *IEEE Communications Magazine*, vol. 57, no. 9, pp. 113–119, Sept. 2019.

[88] M. G. Bellemare, G. Ostrovski, A. Guez, P. Thomas, and R. Munos, "Increasing the action gap: New operators for reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[89] C. Watkins and P. Dayan, "Technical Note: Q-Learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, May 1992.

[90] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[91] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[92] A. Fallah, A. Mokhtari, and A. Ozdaglar, "On the convergence theory of gradient-based model-agnostic meta-learning algorithms," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 1082–1092.

[93] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT press, 2016.

[94] Federal Communications Commission (FCC), "Unlicensed Use of the 6 GHz Band," *Docket No. 17-183*, Apr. 2020.

[95] Y. Gao, L. Dai, and X. Hei, "Throughput Optimization of Multi-BSS IEEE 802.11 Networks With Universal Frequency Reuse," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3399–3414, May 2017.

[96] 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," in *Advances in Neural Information Processing Systems*, 2019, pp. 8026–8037.

VITA

Journal Papers:

Lyutianyang Zhang, and et.al: "Communications-Caching-Computing ResourceAllocation for Bidirectional Data Computationin Mobile Edge Networks", IEEE Transactions on Communications, 69(3):1496–1509, 2021.

Hao Yin, **Lyutianyang Zhang**, Sumit Roy: "Multiplexing URLLC Traffic within eMBB Services in 5G NR: Fair Scheduling", IEEE Transactions on Communications, 69(2):1080–1093, 2021.

Lyutianyang Zhang, and et.al: "Multi-Access Point Coordination for Next-GenWi-Fi Networks Aided by Deep Reinforcement Learning", sent to IEEE System Journal, under major revision.

Conference Papers:

Lyutianyang Zhang, H. Yin, Z. Zhou, S. Roy and Y. Sun, "Enhancing WiFi Multiple Access Performance with Federated Deep Reinforcement Learning," 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), Victoria, BC, Canada, 2020, pp. 1-6, doi: 10.1109/VTC2020-Fall49728.2020.9348485.

Lyutianyang Zhang, Liu Cao, Sumit Roy, "Efficient PHY Layer Abstraction for 5G NR Sidelink in ns-3" (Accepted by WNS3).

V. Sathya, **Lyutianyang Zhang** and M. Yavuz, "Towards Private 5G O-RAN Implementation: Performance and Business Validation," 2022 IEEE Future Networks World Forum (FNWF), Montreal, QC, Canada, 2022, pp. 676-681, doi: 10.1109/FNWF55208.2022.00124.

L. Cao, **Lyutianyang Zhang**, S. Jin and S. Roy, "Efficient MIMO PHY Abstraction With Imperfect CSI for Fast Simulations," in IEEE Wireless Communications Letters, vol. 12, no. 3, pp. 530-534, March 2023, doi: 10.1109/LWC.2022.3233542.

Peng Liu, Chen Shen, Chunmei Liu, Fernando J. Cintrón, **Lyutianyang Zhang**, Liu Cao, Richard Rouil, and Sumit Roy. 2022. 5G New Radio Sidelink Link-Level Simulator and Performance Analysis. In Proceedings of the 25th International ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM '22). Association for Computing Machinery, New York, NY, USA, 75–84. https://doi.org/10.1145/3551659.3559049

Teaching Experience:

1. Teaching Assistant, ECE Department, UW, Fall 2018, EE 505 (Probability and Random Process).

2. Teaching Assistant, ECE Department, UW, Fall 2018, EE 461 (Computer Networking).

Industry Experience:

1. Spectrum Research, Nokia Bell Lab, Naperville, Chicago, IL, Summer 2019.

2. 5G ORAN Engineer, Celona Inc., Cupertino, CA, Summer 2022.