Doctoral Thesis

Towards More Efficient 5G Networks via Dynamic Traffic Scheduling

Sung-tae Hong

Department of Electrical Engineering

Graduate School of UNIST

2020

Towards More Efficient 5G Networks via Dynamic Traffic Scheduling

Sung-tae Hong

Department of Electrical Engineering

Graduate School of UNIST

Towards more efficient 5G networks via dynamic traffic scheduling

A dissertation
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Sung-tae Hong

01/03/2020

Approved by

Advisor
Hyoil Kim

Towards more efficient 5G networks via dynamic traffic scheduling

Sung-tae Hong

This certifies that the dissertation of Sung-tae Hong is approved.

01/03/2020

signature

Advisor: Hyoil Kim

signature

Kyunghan Lee

signature

Jin Ho Chung

signature

Hyun Jong Yang

signature

Youngbin Im

Abstract

The 5G communications adopt various advanced technologies such as mobile edge computing and unlicensed band operations, to meet the goal of 5G services such as enhanced Mobile Broadband (eMBB) and Ultra Reliable Low Latency Communications (URLLC). Specifically, by placing the cloud resources at the edge of the radio access network, so-called mobile edge cloud, mobile devices can be served with lower latency compared to traditional remote-cloud based services. In addition, by utilizing unlicensed spectrum, 5G can mitigate the scarce spectrum resources problem thus leading to realize higher throughput services.

To enhance user-experienced service quality, however, aforementioned approaches should be more fine-tuned by considering various network performance metrics altogether. For instance, the mechanisms for mobile edge computing, e.g., computation offloading to the edge cloud, should not be optimized in a specific metric's perspective like latency, since actual user satisfaction comes from multi-domain factors including latency, throughput, monetary cost, etc. Moreover, blindly combining unlicensed spectrum resources with licensed ones does not always guarantee the performance enhancement, since it is crucial for unlicensed band operations to achieve peaceful but efficient coexistence with other competing technologies (e.g., Wi-Fi).

This dissertation proposes a focused resource management framework for more efficient 5G network operations as follows. First, Quality-of-Experience is adopted to quantify user satisfaction in mobile edge computing, and the optimal transmission scheduling algorithm is derived to maximize user QoE in computation offloading scenarios. Next, regarding unlicensed band operations, two efficient mechanisms are introduced to improve the coexistence performance between LTE-LAA and Wi-Fi networks. In particular, we develop a dynamic energy-detection thresholding algorithm for LTE-LAA so that LTE-LAA devices can detect Wi-Fi frames in a lightweight way. In addition, we propose AI-based network configuration for an LTE-LAA network with which an LTE-LAA operator can fine-tune its coexistence parameters (e.g., CAA threshold) to better protect coexisting Wi-Fi while achieving enhanced performance than the legacy LTE-LAA in the standards. Via extensive evaluations using computer simulations and a USRP-based testbed, we have verified that the proposed framework can enhance the efficiency of 5G.

Contents

I Introduction 1

1.1 Main Contributions 1

1.2 Organization of the Dissertation 2

II QoE-aware Computation Offloading to Capture Energy-Latency-Pricing Tradeoff in Mobile Clouds 3

2.1 Introduction 3

2.2 System Model 4

2.3 MECOS Framework and Computation Offloading Procedure 8

2.4 QoE-aware Transmission Scheduling of Offloaded Task's Data 11

2.5 QoE-aware Service Class Selection 21

2.6 Performance Evaluation 24

2.7 Conclusion and Future Work 35

III Lightweight Wi-Fi Frame Detection for Licensed Assisted Access LTE 36

3.1 Introduction 36

3.2 Preliminaries 38

3.3 SCD-based Wi-Fi Preamble Detector Design for LAA-LTE 38

3.4 Spatial Reuse Enhancement by the Proposal 48

3.5 Conclusion and Future Work 57

IV AI-powered Unlicensed Bands Communication for Enhanced Coexistence Performance . . . 58

4.1 Introduction . . . 58

4.2 Preliminaries . . . 59

4.3 System Model . . . 59

4.4 Proposed Algorithm . . . 61

4.5 Experiment . . . 63

4.6 Conclusion . . . 65

V Conclusion . . . 68

References . . . 69

Acknowledgements . . . 77

List of Figures

1 Computation Offloading Architecture 5

2 Finite-state Markov chain (FSMC) 6

3 QoE factors considered in the chapter 7

4 Two QoE scenarios 7

5 A procedure for computation offloading 9

6 Discrete-time model with slot time τ 10

7 Two types of matrices in the database-DP algorithm 14

8 Time slot aggregation for approximate dynamic programming 16

9 Exact values of $\zeta_{t;\widetilde{T}}$ and an approximate line of $\breve{\zeta}_{t;\widetilde{T}}$ 20

10 Performance loss of ADP and ADPe compared to optimal DP 26

11 An optimal schedule: DP vs. ADPe (when $B = 0.5$) 27

12 APP1: class selection and offloading cost 28

13 APP2: class selection and offloading cost 29

14 APP2*: class selection and offloading cost 30

15 QoE enhancement ratio with varying α and B 31

16 Energy, completion time, and monetary cost with varying (α, B) 32

17 QoE-aware scheduling vs. energy-only scheduling 34

18 Structure of a Wi-Fi frame 38

19 Proposed lightweight Wi-Fi frame detection mechanism for LAA devices 40

20 LAA and Wi-Fi samples for the first 4.264 μs of L-STF 42

21 (left) Implemented Wi-Fi and LAA transmitters based on USRP N210 and (right) experiments scenario, where ① is the Wi-Fi transmitter transmitting Wi-Fi signals and ② is the LAA transmitter or another Wi-Fi transmitter attempting to detect the Wi-Fi frame while they are performing CCA-based backoff. 43

22 SCD performance of Wi-Fi and LAA under S_1 (NLOS) and S_2 (LOS). (a) C_W in S_1. (b) C_L in S_1. (c) C_W in S_2. (d) C_L in S_2. 44

23 P_{M} and P_{FA} of the proposed SCD mechanism, with $N_{\mathrm{L}} = 98$ and various C_{th} . . 45

24 P_{D} and P_{FA} of the proposed SCD mechanism, with varying SNR. (a) P_{D}. (b) P_{FA}. 46

25 Concept of IoU and IoA 46

26 IoU, and IoA of frame detection scheme. (a) IoU. (b) IoA. 47

27 Hexagonal deployment of the two coexisting networks 50

28 Throughput of the center LAA BS 52

29 Delay of the center LAA BS 54

30 Throughput of the center Wi-Fi AP 55

31 Delay of the center Wi-Fi AP 56

32 Variation of E_{max} with varying P_{max} 61

33 Overview of RLA-CU 62

34 Simulator System for RLA-CU 63

35 Performance variation of an LAA operator 66

36 Performance variation of a Wi-Fi operator 67

I Introduction

The 5G network has been recently launched in some countries starting with South Korea, and is going to prevail in the world by 2020. Moreover, the number of 5G connections is expected to take 15 percent of the global connections [1]. Compared to its predecessors, 5G is designed to meet the requirements of various services, which are divided into three main categories; massive Machine Type Communications (mMTC), enhanced Mobile Broadband (eMBB), and Ultra Reliable Low Latency Communications (URLLC) [2]. In addition, 5G will more aggressively apply various AI techniques to its architecture and network management to realize more enhanced efficiency.

MEC, one of the key enablers of URLLC, provides computing resources at the edge of the radio access network to the mobile users via virtualization. Traditionally, cloud computing services have been provided by remote clouds in the Internet, and thus it was hard to provide more customized and low-latency services. On the contrary, MEC can realize new business opportunities such as location-aware services, augmented reality, and computation offloading [3], thanks to its proximity to the mobile devices. Among them, computation offloading is a promising application to promote the lifetime and performance of mobile devices by migrating a mobile device's tasks to the edge cloud over the 5G network. One of the challenges, however, is that multiple devices should share spectrum and cloud resources, thus necessitating two kinds of resource management, computation resource management and spectrum resource management.

The 5G network tries to aggressively use all available spectrum bands, including not only its own licensed spectrum bands but also unlicensed spectrum bands, to cope with eMBB. In the unlicensed spectrum, multiple technologies can coexist and hence a peaceful coexistence mechanism with them is essential for 5G. To address the concern, Licensed-Assisted Access (LAA) was first introduced in 3GPP Release 13 [4] based on the listen-before-talk mechanism to utilize the unlicensed spectrum as a supplement to the licensed spectrum, and later on LTE based LAA, called LTE-LAA, was successfully standardized. As the successor to LTE-LAA, 5G NR based unlicensed spectrum access, called NR-U, is under development and is expected to be published with 3GPP Release 16. According to the 3GPP document [5], the specification of NR-U operation considers LTE-LAA as the baseline in terms of the coexistence method. Therefore, the LTE-LAA centric research is still compelling and can be further extended to the future 5G NR-U operations.

1.1 Main Contributions

This dissertation aims to provide the resource management framework for more efficient 5G network, where the main contributions are:

- **Quality-of-Experience aware Computation Offloading Strategy to Capture Energy-Latency-Pricing Tradeoff.** A mobile device makes a computation offloading strategy based on the time-varying wireless channel state and the performance of the mobile edge cloud. The most crucial part of the strategy is considering the tradeoff relationship between energy consumption, latency, and monetary cost. For resolving this issue, we have proposed Quality-of-Experience based optimization algorithms in [6, 7], to decide whether to perform computation offloading, which edge cloud resources are used, and how to efficiently transmit the offloading data.

- **Improving the spatial efficiency of LTE-LAA network through the lightweight Wi-Fi frame detection method.** LTE-LAA adopts the listen-before-talk mechanism to avoid a collision in unlicensed spectrum bands. The criterion of LTE-LAA to determine the channel state varies with a coexistence situation. For example, the threshold of -52 dBm is applied at the absence of other technologies on a long-term basis, and otherwise -72 dBm is used. Because Wi-Fi is already pervasive, LTE-LAA sacrifices its spatial efficiency for peaceful coexistence, even though LTE-LAA is designed to operate at a high level of interference. To improve spatial efficiency, we have proposed the adaptive thresholding algorithm for clear channel assessment (CCA) with a lightweight Wi-Fi frame detection method in [8], which improves the performance of both LTE-LAA and Wi-Fi networks.

- **Enhancing the coexistence in unlicensed spectrum bands using AI.** The goal of an LTE-LAA network is peacefully coexisting with Wi-Fi networks in the unlicensed spectrum. However, with a static configuration of LTE-LAA's network parameters, the goal is hard to be achieved in various environments. We, therefore, propose an AI-based mechanism to cope with the issue. Specifically, reinforcement learning is adopted to adjust the LAA's control parameters dynamically, and being integrated with a discriminator neural network, the proposed algorithm shows that the performance of both LTE-LAA and Wi-Fi can be improved.

1.2 Organization of the Dissertation

The rest of the dissertation is organized as follows. Chapter 2 proposes a computation offloading strategy that decides how the mobile device transmits offloading data to a mobile edge cloud. Chapter 3 presents a dynamic but lightweight Wi-Fi frame detection algorithm to improve the spatial efficiency of the LTE-LAA network. Chapter 4 introduces an adaptive control of coexistence parameters of an LTE-LAA network to improve its own performance while protecting coexisting Wi-Fi networks. Finally, the dissertation is concluded in Chapter 5.

II QoE-aware Computation Offloading to Capture Energy-Latency-Pricing Tradeoff in Mobile Clouds

2.1 Introduction

Mobile devices have become the center of our daily lives, running various applications. The number of mobile devices is expected to reach 11.6 billion by 2021 due to the emerging mobile technologies like wearable devices and Internet of Things (IoT) [9]. Nevertheless, the usability of mobile devices is still limited by the slow evolution of battery capacity, e.g., the energy density of Li-ion batteries has increased three times in 23 years [10] while the mobile CPU performance has improved tenfold in 7 years [11]. Accordingly, it has been reported that customer satisfaction in mobile usage relies heavily on the battery life [12].

Computation offloading is a promising technology to promote the lifetime and performance of mobile devices, by migrating local computational tasks to the cloud over wireless channels [13–17]. One of its key challenges is optimal mobile-to-cloud transmission scheduling of the task's data (e.g., task's code to run remotely), for which an offloading scheduler optimizes the per-slot number of bits to transmit. Among various approaches to tackle the problem, one possible direction is to focus on minimizing the transmit energy consumption while satisfying some delay constraints, e.g., application deadline, as found in [18, 19]. Such schemes, however, tend to penalize the latency by deferring transmission until 'good' time slots [18] or by extending the transmission period with lowered transmit power [19].

Considering that mobile users value not only battery life but also the processing speed and service pricing [20], it is desirable to properly model the tradeoff between them. In such a vein, [21] proposed uplink-scheduling multiple applications in bundles at high-power states while considering how much they are postponed, and [22] proposed scheduling for video streaming that optimizes when to download based on the predicted user location and channel quality. In addition, [23, 24] tried to capture the tradeoff via Lyapunov optimization. Nevertheless, [21, 22, 24] ignored the impact of time-varying wireless channel quality, while [23] only considered delay-tolerant applications and used the transmit queue size as an indirect measure of the application delay. Moreover, none of them considered the issue of pricing in offloading service provisioning. Although [25] jointly considered energy, delay, and pricing in deciding which tasks to offload, it did not consider transmission scheduling nor the time-varying nature of a wireless channel.

In addition, it is important to consider local-cloud interactions in computation offloading to come up with practical offloading solutions. In this sense, [26, 27] proposed offloading decision schemes for an application consisting of multiple dependent tasks, to make per-task offloading decisions. [28] tried to pick the best cloud-network service provider pair to improve the quality of service, based on the cloud-specific and network-specific characteristics. [29] considered multi-broker based mobile cloud computing networks where the brokers pay for reserving cloud resources and assign them to mobile users, and proposed a game-theoretic approach to minimize the total incurred price. However, [26, 27, 29] considered the latency cost only as a constraint and ignored its impact on user satisfaction, and [27–29] overlooked the influence of time-varying channel quality.

This chapter addresses the tradeoff between energy consumption, application latency, and offloading service pricing using the concept of quality of experience (QoE), which is a blueprint of subjective and objective human needs and experiences in a given context [30]. For instance, at a high remaining battery level, a mobile user may put more weight on latency than energy, and thus such a control knob should be designed to be aware of users' QoE. In particular, we propose a computation offloading service framework, based on which three offloading problems are formulated: decision making of local vs. offloaded computing, optimal selection of the offloading service class, and optimal transmission scheduling of the offloaded task's data.

The optimal scheduling problem is formulated as dynamic programming (DP) considering a time-varying wireless channel, based on which we propose the database-assisted optimal DP algorithm and two suboptimal but computationally-efficient approximate DP (ADP) algorithms (called ADP and ADPe) by adopting the limited lookahead technique [31]. In addition, the service class selection problem is formulated such that it can determine the best service class (or service plan) from the ones offered by the cloud service provider where each class provides distinct computing power and service charge.

Via an extensive numerical analysis, we evaluated the performance of the proposed schemes in various QoE scenarios. Especially, ADPe is shown to achieve near-optimal performance with only 2.1% deviation from the optimal cost on average and enhance the QoE-aware cost by up to 2.38 times compared to the energy-minimal scheduling. Moreover, the analysis confirms that the class selection algorithm dynamically selects the service class according to user preference and the remaining battery level under the given monetary constraint.

The rest of this chapter is organized as follows. Section 2.2 introduces the system model and assumptions, and Section 2.3 elaborates the proposed computation offloading framework. Section 2.4 presents the DP, ADP and ADPe formulations of the optimal transmission scheduling problem, and Section 2.5 presents the optimal service class selection problem and the decision making problem. Section 2.6 evaluates the performance of the proposed schemes, and the chapter concludes with Section 2.7.

2.2 System Model

We consider a mobile edge system (MES) [32] located at the edge of a mobile network (e.g., eNB in LTE-A), through which mobile edge computing (MEC) [33] capabilities are provided to the mobile users. Specifically, a mobile user can offload its task (which is a part of a mobile application) to the MES as follows: the user transmits the task's data (e.g., code, inputs, etc.) to the MES over the wireless channel, and then the MES assigns a virtual machine (VM) running on its resources (e.g., CPU) to the task for offloaded processing [34, 35]. We also assume that the MES provides multiple service classes with varying computing power and service charge,

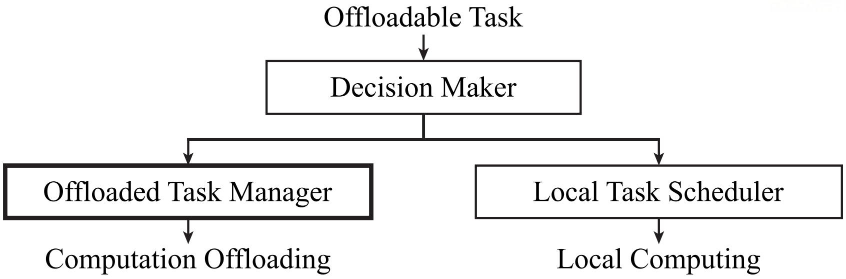


Figure 1: Computation Offloading Architecture

so that a mobile user can choose the most suitable one. Therefore, the MES and its resources can be considered as ‘the cloud’ in this chapter. Note that Section 2.3 will further describe our proposed computation offloading framework.

In a mobile device, computation offloading is performed by three major components: a decision maker, a local task scheduler, and an offloaded task manager, as shown in Fig. 1. To process a task, the decision maker chooses the most efficient method between local computing and computation offloading by comparing their incurred costs. When local computing is chosen, the local task scheduler optimizes the operation of local processors using various scheduling techniques [36,37]. When computation offloading is chosen, the offloaded task manager performs two functions, service class selection and transmission scheduling. The former is to choose the optimal service class to achieve the minimal QoE-aware cost, and the latter is to determine when and at which rate the task’s data is transferred from the mobile device to the cloud. Therefore, the decision maker interacts with the two other components to collect the costs of local computing and offloaded computing, so that it can make a decision.

Note that this chapter first focuses on the offloaded task manager and its functionalities (e.g., transmission scheduling) in Section 2.4, assuming that the decision maker has decided to offload a given task. Next, in Section 2.5, the chapter will discuss the service class selection problem based on the results in Section 2.4, and then provide further details on the mechanism of the decision maker and local task scheduling in Section 2.5 via comparison of the local computing cost with that of the offloading cost derived in Section 2.5.

Wireless Channel Model

We assume a Rayleigh fading channel with slow block fading, which is modeled as a finite-state Markov chain (FSMC) with K states as shown in Fig. 2 [38]. In the model, a state $S_k \in \{S_1, \ldots, S_K\}$ represents a distinct interval of nominal received SNR $\gamma^{nom} \in [\Gamma_{k-1}, \Gamma_k)$ according to the nominal transmit power P_{tx}^{nom}, with $\Gamma_0 = 0$ and $\Gamma_K = \infty$. This means that we have $\gamma^{nom} = P_{tx}^{nom} g/(N_0 W)$ where g is the channel gain, N_0 is the noise power spectral density, and W is the channel bandwidth. Therefore, the nominal SNR γ^{nom} is one-to-one mapped to the channel gain g. In addition, when the actual transmit power is P_{tx}, the actual received SNR is given as $(P_{tx}/P_{tx}^{nom})\gamma^{nom}$.

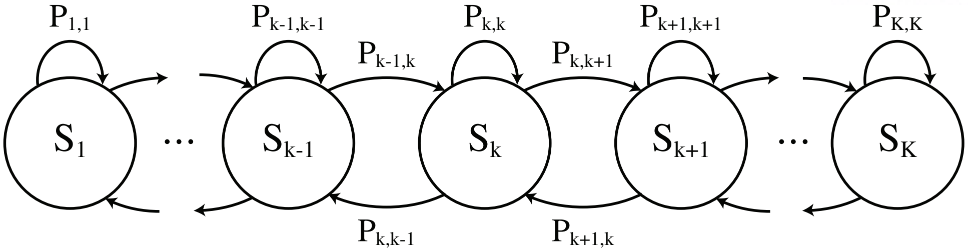


Figure 2: Finite-state Markov chain (FSMC)

For state S_k, we designate $\overline{\gamma}_k$ as the representative (nominal) SNR, which is the average SNR of the corresponding interval. For $k < K$, we have $\overline{\gamma}_k = (\Gamma_{k-1} + \Gamma_k)/2$. For $k = K$, we define $\overline{\gamma}_k = \Gamma_{K-1} + (\Gamma_{K-1} - \Gamma_{K-2})/2$.

The transition probabilities between adjacent states are approximated as [38, 39]

$$P_{k,k+1} \approx N(\Gamma_k) \cdot T_p/\pi_k, \quad k = 1, 2, \ldots, K-1, \tag{1}$$

$$P_{k,k-1} \approx N(\Gamma_{k-1}) \cdot T_p/\pi_k, \qquad k = 2, 3, \ldots, K, \tag{2}$$

where T_p is the packet duration, $N(\Gamma_k)$ is the level crossing rate of γ^{nom} over the SNR threshold Γ_k such as [38]

$$N(\Gamma_k) = \sqrt{\frac{2\pi\Gamma_k}{\gamma_0}} f_m \exp\left(-\frac{\Gamma_k}{\gamma_0}\right), \tag{3}$$

and π_k is the steady-state probability of S_k such as

$$\pi_k = \int_{\Gamma_{k-1}}^{\Gamma_k} p(\gamma^{nom}) d\gamma^{nom} = \exp\left(-\frac{\Gamma_{k-1}}{\gamma_0}\right) - \exp\left(-\frac{\Gamma_k}{\gamma_0}\right), \tag{4}$$

where γ_0 is the average (nominal) received SNR, f_m is the maximum Doppler frequency, and $p(\gamma^{nom})$ is the Rayleigh distribution.

Note that this chapter adopts the time-slotted architecture, using which a mobile user tries to determine how many bits to transmit at each slot until the whole necessary information for offloading is transmitted to the MES. Hence, the channel state of the FSMC model implies the state of a slot, assuming the coherence time of the channel is larger than the slot duration. Section 2.2 will provide further justification on the practicality of the time-slotted architecture.

QoE Factors

To capture the tradeoff between energy, latency, and pricing, we consider four QoE domains discussed in [30], human, contextual, technological, and business domains, as shown in Fig. 3. The human domain represents a human entity with various attributes like age, gender, role, and can be further categorized into subjective (e.g., feelings) and objective (e.g., biological parameters) QoE factors. The technological domain deals with every possible technological aspect during a service including network resources, devices, and quality-of-service (QoS). The business domain is related to commercial aspects of service provisioning such as pricing, which

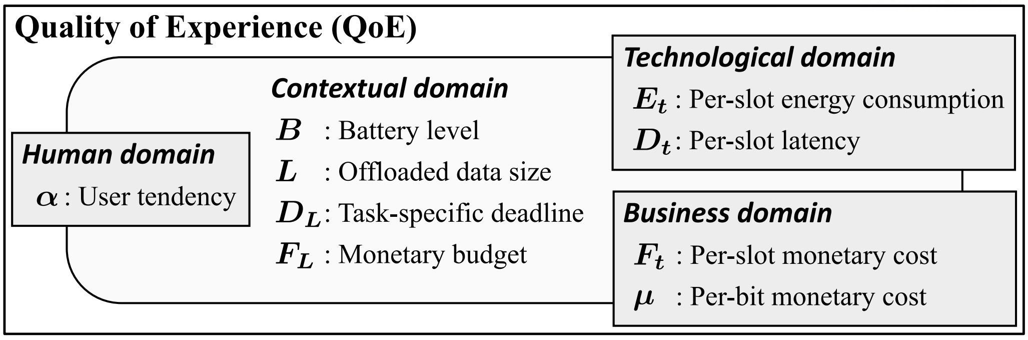
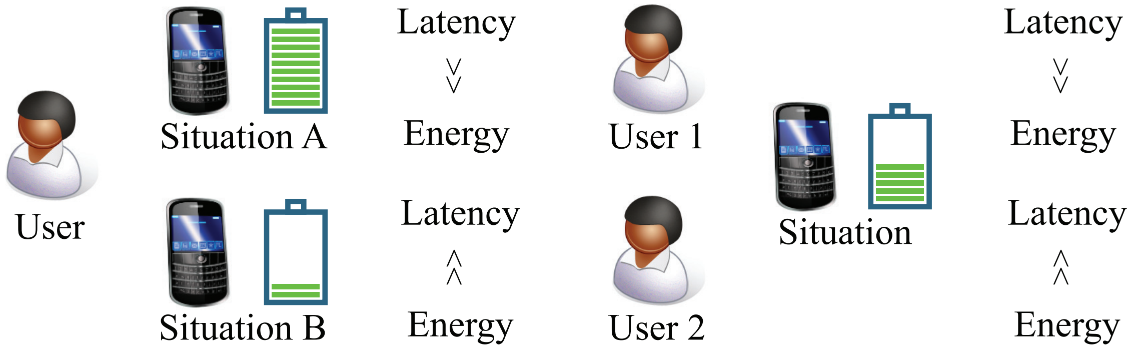

Quality of Experience (QoE)
Contextual domain
B : Battery level
L : Offloaded data size
D_L: Task-specific deadline
F_L : Monetary budget
Human domain
α : User tendency
Technological domain
E_t : Per-slot energy consumption
D_t: Per-slot latency
Business domain
F_t : Per-slot monetary cost
μ : Per-bit monetary cost

Figure 3: QoE factors considered in the chapter



(a) User tendency varies with the remaining battery level

(b) Sensitivity to QoS metrics varies with users

Figure 4: Two QoE scenarios

influence a customer's decision of choosing a service. Lastly, the contextual domain concerns with how all the domains interact with each other in given circumstances.

In this chapter, we consider nine QoE-related factors, each belonging to one of the four QoE domains as shown in Fig. 3. We first introduce a factor in the human domain, latency sensitivity $\alpha \in [0, 1]$, which reflects a user's preference to latency — a user with larger α prefers earlier completion of a task at the expense of energy; a user with small α desires to save more energy rather than minimizing latency. Regarding the contextual domain, we consider four factors, normalized battery level $B \in (0, 1]$, offloaded data size L (bits), task-specific deadline D_L (sec), and user's monetary budget F_L.[1] In terms of the technological domain, we consider two factors, per-slot energy consumption E_t and per-slot latency D_t. Lastly, regarding the business domain, we consider two factors, per-slot monetary cost F_t and per-bit monetary cost μ, where the former captures the time-based service charge by the offloading service provider and the latter models the charge by the network provider for data traffic transmitted over the wireless network.

To capture diverse user behaviors due to the tradeoff between energy and delay, we employ αB as a weighting factor using which the QoE-aware cost function is to be defined in Section 2.4. This approach can be better justified by two exemplifying scenarios in Fig. 4. In Fig. 4a, the

[1] D_L is assumed to be given by the offloaded task.

user's preference varies with the current battery level: the user desires high performance with a sufficient battery level, but wants to save energy when the battery is running out. In Fig. 4b, the two mobile users present different tendency in the same situation: User 1 prefers early completion of a task to extended battery lifetime, while User 2 has an opposite preference. Hence, when α is large (thus latency sensitive) but B is minimal (i.e., the battery is almost depleted), it is more reasonable to increase the sensitivity to energy.

Slotted Time Structure

This chapter adopts the time-slotted architecture (which will be further described in Section 2.3) due to its ability to model some practical scenarios. For instance, LTE has Physical Resource Block (PRB) with a duration of 0.5 ms as a basic unit of time-frequency resources, where such resources are allocated once every 1 ms, referred to as Transmission Time Interval (TTI). In TDD-LTE, per-UE uplink resource allocation could happen once every 10 ms (which is a frame duration), since 10 subframes in a frame are configured differently among D/U/S. Then, the slot duration in our model can be set as either 1 ms or 10 ms. In addition, in some applications like VoIP and machine type communications [40], an eNodeB employs semi-persistent scheduling in which PRBs can be allocated in advance over multiple TTIs to reduce the scheduling overhead. The whole duration of semi-persistent scheduling can then be set to D_L, so that a mobile device can periodically transmit data once every slot without interruption until the completion of offloading.

One remaining issue is that such a slot duration would be proper to be modeled as the FSMC. For $f_m = 5$ Hz, which is commonly assumed in LTE's performance evaluation scenarios [41], the channel's coherence time is determined as $0.423/f_m = 84.6$ ms [42] which is much larger than 1 ms and 10 ms. Moreover, in soon-to-come 5G, TTI will be further shortened to 0.1 ms, which will make the slot duration reduced by 10 times (i.e., 0.1 ms or 1 ms).

2.3 MECOS Framework and Computation Offloading Procedure

In this section, we propose the Mobile Edge Computation Offloading Service (MECOS) framework, a customized cloud service model, which has the following characteristics.

- MECOS employs an MES that possesses a variety of virtualized resources, and offers them through multiple service classes. Each class is mapped to a distinct type of resources (e.g., a specific CPU type like quad-core CPU) and has its own service charge. The service tariff is advertised to mobile users in advance, and not changing while an offloading process is in action.

- MECOS is ready to serve computation offloading whenever requested by a user, i.e., the VM startup time is unnecessary, by maintaining a pool of pre-booted VMs dedicated to the offloading service.

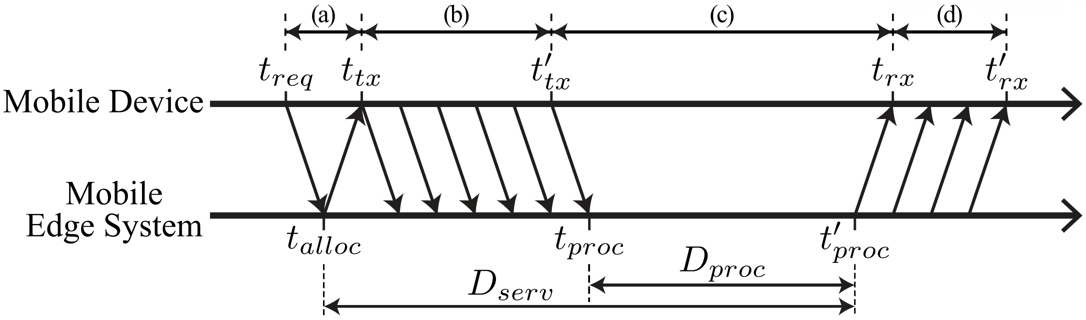


Figure 5: A procedure for computation offloading

- Once a service request is received from a mobile user, MECOS denies the service if there exist insufficient resources to allocate; otherwise, the requested resources are allocated to the user and secured until the offloaded computation is complete.

- The service charge is proportional to the total usage duration, where the charge starts at the allocation of the requested resources and ends at the completion of the requested computation.

Based on the MECOS framework, we introduce our computation offloading procedure, which is also illustrated in Fig. 5, as follows.

(a) **Initialization phase**: In this phase, a mobile user sends an initialization request to MECOS (at time t_{req}), specifying which class of the virtualized resources is required. On receipt of the request, MECOS allocates the requested resources (at time t_{alloc}), transmits an initialization response back to the device, and starts to charge the usage fee.

(b) **Data transmission phase**: After the request granted, the user starts to transmit (at time t_{tx}) L bits of offloading data according to its transmit schedule, and thus the transmission completion time t'_{tx} varies with how transmission scheduling is performed. Note that transmission scheduling will be discussed in Section 2.4.

(c) **Processing phase**: After full reception of the offloading data, MECOS processes the offloaded task from time t_{proc} till time $t'_{proc} = t_{proc} + D_{proc}$, where D_{proc} depends on the characteristics of the task and the computing power of the allocated resources and will be derived in Section 2.5. Afterwards, the allocated resources are retrieved at time t'_{proc} at which the service charge finishes. Then, the total time of cloud service becomes $D_{serv} = t'_{proc} - t_{alloc}$.

(d) **Response phase**: The computation results are transmitted back to the user, with a duration of $D_{rx} = t'_{rx} - t_{rx}$.

In addition, we consider major properties that have a substantial effect on computation offloading as follows.

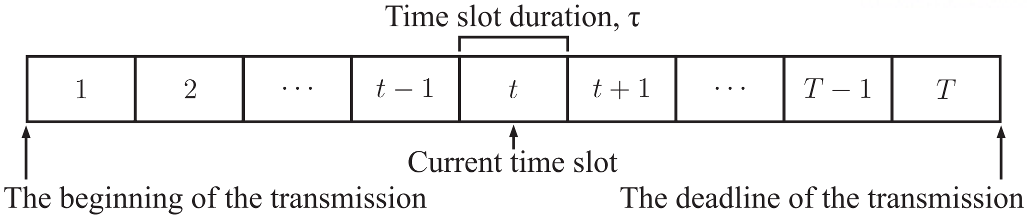


Figure 6: Discrete-time model with slot time τ

Application Properties

To characterize an offloaded task, we focus on following application properties. First, R^M_{app} denotes the number of required cycles on a mobile device per unit input data, e.g., 31,680 cycles/bit for a face recognition application [43] and 330 cycles/byte for a gzip application [44]. The number of required cycles, however, may vary with the type of a processor, such as x86 or ARM, even if the same application is executed [45]. Therefore, we also define the number of required cycles on the MECOS framework as $R_{app} = \theta \cdot R^M_{app}$, where θ $(0 < \theta \leq 1)$ is the weighting factor to reflect how fast the processor at MECOS is compared to the mobile processor.

Second, the data transmission phase should be completed before T (slots) as shown in Fig. 6, where each slot has a duration of τ (sec). The value of T is determined by two factors, the task-specific deadline D_L and the monetary budget of the user, which will be discussed later in Section 2.5. Third, during the data transmission phase, a mobile user transmits L (bits) of task-specific data to the cloud, whose amount varies from a few KB in the code partition case [14] to tens of MB in the virtual machine migration case [16]. Finally, we denote by L_{rx} (bits) the size of the offloaded computation's result to be delivered back to the user during the response phase.

MECOS Properties

We assume that MECOS has a set of Q service classes such as $\mathbb{Q} = \{1, \ldots, Q\}$, where each class $q \in \mathbb{Q}$ has a different usage charge and computing power. Such a multi-class model can be easily found at today's data centers which are equipped with heterogeneous hardware (e.g., various CPUs with different specifications). We also assume that the usage fee of MECOS is charged on a time-block basis, as found from Microsoft Azure [46] and Amazon EC2 [47].

Accordingly, we model that MECOS charges δ_q every ϵ seconds for the usage of class q, where $\epsilon = \lambda\tau, \lambda \in \mathbb{Z}^+$ and ϵ can be in the order of seconds or milliseconds to provide fine-grained pricing for real-time offloading service. Note that a duration of ϵ (sec) will be referred to as a *(charging) period* in the sequel. In addition, we denote the computing power of class q by H_q (cycles/period).[2]

All the properties are assumed known when a task is offloaded. Then, we focus only on data transmission and processing phases due to the following reasons. First, the initialization

[2] 'cycle' means a computation cycle and 'period' means a charging period.

phase is negligible due to small propagation delay (since single-hop) and small packet sizes of the initialization request/response. Second, the cost incurred in the response phase is un-controllable by the mobile device since L_{rx} is transmitted by the mechanism of a base station.

2.4 QoE-aware Transmission Scheduling of Offloaded Task's Data

This section defines the QoE-aware cost function and proposes three scheduling algorithms. First, we present the DP algorithm to derive the optimal solution of QoE-aware scheduling, whose complexity is non-trivial. Then, we propose the ADP and the ADPe algorithms that produce suboptimal solutions with much reduced complexity, whose near-optimal performance will later be shown in Section 2.6.

Frequently-used notations are summarized in Table 1. Note that this section reserves t as the time slot index, and assumes that the mobile user has leased the MECOS class q without loss of generality.

QoE-aware Cost Function

To determine the optimal amount of transmit data (in bits) at each slot, we define a QoE-aware cost function by combining energy consumption, communication latency, and monetary costs.

First, we define the energy cost function according to the model proposed in [48], which approximates the piecewise linear power-rate curve in practical systems. Specifically, denoting by s_t the amount of transmit data (in bits) at slot t, the energy consumption $E_t(s_t)$ at slot t is given as

$$E_t(s_t) = P_{tx} \cdot \tau = \frac{m \cdot {s_t}^n}{\tau^{n-1} g_t} \simeq u \frac{{s_t}^n}{Z_t}, \quad u = \frac{m P_{tx}^{nom}}{N_0 W \tau^{n-1}}, \tag{5}$$

where g_t is the channel gain at slot t, $n > 1$ and $m > 0$ are given design parameters to best approximate the power-rate curve [48], and P_{tx} is the actual transmit power introduced earlier in Section 2.2. The approximation at the third equality comes from replacing the nominal SNR corresponding to g_t with the representative nominal SNR $Z_t \in \{\overline{\gamma}_1, \ldots, \overline{\gamma}_K\}$. Physically, this model adapts P_{tx} to transmit s_t bits at slot t, for given Z_t. This chapter assumes that the current channel state is measured and known at the beginning of a slot, and the term 'channel state' will refer to either Z_t or S_k.

We define the latency cost function $D_t(l_t)$ at slot t as

$$D_t(l_t) = \tau \mathbb{1}_{\{l_t > 0\}}, \tag{6}$$

where l_t is the remaining bits to transmit, and $\mathbb{1}_{\{\cdot\}}$ is the indicator function which returns 1 if the statement inside the curly brackets is true and 0 otherwise. That is, as long as l_t is nonzero, the slot t should be utilized and thus an additional delay of τ is incurred. Note that $l_{t+1} = l_t - s_t$.

In addition, the monetary cost function $F_t(l_t)$ at slot t is defined similar to $D_t(l_t)$, such as

$$F_t(l_t) = \frac{\delta_q}{\lambda} \mathbb{1}_{\{l_t > 0\}}, \tag{7}$$

Table 1: Frequently-used notations in each scheduling algorithm

	DP (Sec. 2.4)	**ADP (Sec. 2.4)**	**ADPe (Sec. 2.4)**
Key concepts		Limited lookahead DP	Nonlinear regression of channel states
Cost function	J_t: the cost-to-go function	$\tilde{J}_t$: the approximated J_t	$\breve{J}_t$: the approximated $\tilde{J}_t$
Time slot index	T: the last time slot	$\widetilde{T}$: a virtual completion time $\widetilde{T}^*$: the optimal $\widetilde{T}$	$\breve{T}^*$: the approximated $\widetilde{T}^*$
Channel state	Z_t: the channel state	$1/\zeta_{t;\widetilde{T}}$: the virtual channel state	$1/\breve{\zeta}_{t;\widetilde{T}}$: the approximated $1/\zeta_{t;\widetilde{T}}$
Scheduling result	s_t: the number of bits to transmit s_t^*: the optimal s_t	$\hat{s}_t^*$: the suboptimal s_t provided $\widetilde{T}$ $\tilde{s}_t^*$: the suboptimal s_t provided $\widetilde{T}^*$	$\breve{s}_t^*$: the approximated $\tilde{s}_t^*$

which is caused by the service charge by MECOS during $[t_{alloc}, t_{proc}]$.

Finally, the QoE-aware cost function ξ_t at slot t is defined as a weighted combination of the three costs with weighting factors αB, $w_{QoE} \in (0, 1)$, and w_m, such as

$$\begin{aligned} \xi_t(l_t, s_t) &= (1 - \alpha B)\frac{E_t(s_t)}{E_L} + w_{QoE}\left\{\alpha B \frac{D_t(l_t)}{D_L} + w_m \frac{F_t(l_t)}{F_L}\right\} \\ &= a\frac{s_t^n}{Z_t} + b\mathbb{1}_{\{l_t>0\}}, \end{aligned} \tag{8}$$

where $E_L = B \cdot C_{Batt}$ (C_{Batt}: the battery capacity in Joules), and F_L is the maximum monetary budget within which a user is willing to pay for offloading. That is, E_L, D_L, and F_L are normalizing factors to achieve *unit-less* combination of energy, latency, and monetary cost, where $E_t(s_t)/E_L$ implies the ratio of per-slot energy consumption to the total energy budget, $D_t(l_t)/D_L$ indicates the ratio of per-slot delay to the total time budget, and $F_t(l_t)/F_L$ represents the ratio of the per-slot monetary cost to the maximum monetary budget. In addition, $a = (1 - \alpha B)u/(B \cdot C_{Batt})$ and $b = w_{QoE}\{\alpha B\tau/D_L + w_m\delta_q/(\lambda F_L)\}$.[3]

The weighting factors w_{QoE} and w_m are introduced to strike a balance between energy, delay, and monetary costs. At each slot, the energy cost is proportional to ${s_t}^n$ whereas the delay and monetary costs are either zero or constant. As a result, the gap between the costs gets wider as s_t grows, which is affected by L and T. Hence, w_{QoE} plays a role in maintaining a balance in the weighted sum, which is adjusted properly for a given pair of (L, T).[4] In addition, w_m means the relative weight of the monetary cost to the latency cost. For example, when $w_m = 0$, Eq. (8) represents the energy-latency tradeoff relationship.

Our criterion of determining w_{QoE} and w_m is to make all three costs (i.e., energy, latency, and monetary costs) equally contribute to the total cost in the case when the mobile user faces a value-neutral situation. More specifically, we set w_{QoE} and w_m such that the three costs coincide (in their values or in the order of their magnitudes) when the mobile user equally cares energy and delay (i.e., $\alpha = 0.5$) and has half of the battery life (i.e., $B = 0.5$).[5]

Optimal Scheduling via Dynamic Programming

Our objective is to find the optimal number of bits to transmit per each slot that minimizes the expected total cost $E\left[\sum_t \xi_t(l_t, s_t)\right]$. Since the choice of s_t determines l_{t+1} and thus affects the future decision s_{t+1}, the problem is suitable for being formulated as dynamic programming. Designating l_t as the system state, the cost-to-go function $J_t(l_t)$, which is the sum of the cost at

[3] Note that we assume B is constant during the scheduling process since T is usually in the order of seconds.

[4] In practice, w_{QoE} can be pre-adjusted offline for each possible (L, T) and stored in a table, which can be looked up at runtime.

[5] This implies that the three costs are treated to have the same level of 'importance', since Eq. (8) deals with the *normalized* costs.

$$\begin{bmatrix} & & & & & & J_1(L) \\ J_2(0) & J_2(1) & \cdots & J_2(j-1) & \cdots & & J_2(L) \\ \vdots & \vdots & & \vdots & & & \vdots \\ J_i(0) & J_i(1) & \cdots & J_i(j-1) & \cdots & & J_i(L) \\ \vdots & \vdots & & \vdots & & & \vdots \\ J_T(0) & J_T(1) & \cdots & J_T(j-1) & \cdots & & J_T(L) \end{bmatrix}$$

(a) Cost-to-go Matrix, $\mathbf{M}_1$

$$\begin{bmatrix} & & & & & & s_1^*(L) \\ s_2^*(0) & s_2^*(1) & \cdots & s_2^*(j-1) & \cdots & & s_2^*(L) \\ \vdots & \vdots & & \vdots & & & \vdots \\ s_i^*(0) & s_i^*(1) & \cdots & s_i^*(j-1) & \cdots & & s_i^*(L) \\ \vdots & \vdots & & \vdots & & & \vdots \\ 0 & 1 & \cdots & j-1 & \cdots & & L \end{bmatrix}$$

(b) Optimal-s_t Matrix, $\mathbf{M}_2$

Figure 7: Two types of matrices in the database-DP algorithm

t and the expected cost from $(t+1)$ to T, is expressed as

$$J_t(l_t) = \begin{cases} \min\limits_{0 \le s_t \le l_t} \left\{ a\frac{s_t{}^n}{Z_t} + b\mathbb{1}_{\{l_t>0\}} + \mathop{\mathbb{E}}\limits_{Z_{t+1}} \left[J_{t+1}\left(l_t - s_t\right)\right] \right\}, & \text{when } 1 \le t < T, \\ a\frac{l_t{}^n}{Z_t} + b\mathbb{1}_{\{l_t>0\}}, & \text{when } t = T. \end{cases} \tag{9}$$

Denoting by s_t^* the optimal number of bits to transmit at t,

$$s_t^*(l_t) = \begin{cases} \operatorname*{argmin}\limits_{0 \le s_t \le l_t} \left\{ J_t(l_t) \right\}, & \text{when } 1 \le t < T, \\ l_t, & \text{when } t = T. \end{cases} \tag{10}$$

Therefore, the optimal policy set, $\{s_1^*, \ldots, s_T^*\}$, is obtained by recursively solving Eq. (9) backwards. Then, the achieved total cost by the optimal policy set, J_{DP}^*, is determined as

$$J_{DP}^* = \sum_{t=1}^{T} \left\{ a\frac{(s_t^*)^n}{Z_t} + b\mathbb{1}_{\{l_t^*>0\}} \right\}, \tag{11}$$

$$l_t^* = l_{t-1}^* - s_{t-1}^*, \forall t > 1, \quad l_1^* = L. \tag{12}$$

Generally, a backward iterative method induces high complexity and is often NP-hard. To mitigate the complexity, we propose a database-driven algorithm, called database-DP, that removes the duplicate calculations in the backward iterations by pre-building all possible combinations in the form of a database. The database consists of two types of T-by-$(L+1)$ matrices, the cost-to-go matrix $\mathbf{M}_1$ and the optimal-s_t matrix $\mathbf{M}_2$, as shown in Fig. 7. For each type, there exist K distinct copies such as $\mathbf{M}_1^k$, $\mathbf{M}_2^k$, $k = 1, 2, \ldots, K$, where $\mathbf{M}_1^k$ and $\mathbf{M}_2^k$ assume $Z_i = S_k$, $\forall i$. In the i-th row and the j-th column, we store $\mathbf{M}_1^k(i,j) = J_i(j-1)$ and $\mathbf{M}_2^k(i,j) = s_i^*(j-1)$.

The database construction algorithm is shown in Fig. 1 and summarized as follows. Starting from $i = T$, we decrement i one by one until it reaches $i = 1$. At each i, we enumerate all possible S_k's, and for each chosen S_k we set $Z_i = S_k$ and generate the i-th row entries $\mathbf{M}_1^k(i,j)$ and $\mathbf{M}_2^k(i,j)$, $\forall j$, according to Eqs. (9) and (10). Since the database entries for the rows larger than i have been already created, the two equations can be solved via simple table lookup. Once the database is constructed, the transmission scheduler (of the offloaded task manager) can utilize it online by simply looking up the corresponding s_t^* in the right matrix according to the given set of t, l_t, and Z_t.

Algorithm 1 A pseudocode for database construction

Database L, T

```
procedure DATABASE($L, T$)
    for $i \leftarrow T, 2$ do                                   ▷ For $1 < t \leq T$
        for $k \leftarrow 1, K$ do
            $Z_i \leftarrow S_k$
            for $j \leftarrow 1, L+1$ do
                $l_i \leftarrow j - 1$
                $\mathbf{M}_1^k(i, j) \leftarrow J_i(l_i)$
                if $i = T$ then
                    $\mathbf{M}_2^k(i, j) \leftarrow l_T$
                else
                    $\mathbf{M}_2^k(i, j) \leftarrow s_i$ that minimizes $J_i(l_i)$
                end if
            end for
        end for
    end for
    for $k \leftarrow 1, K$ do                                   ▷ For $t = 1$
        $Z_1 \leftarrow S_k$
        $\mathbf{M}_1^k(1, L) \leftarrow J_1(L)$
        $\mathbf{M}_2^k(1, L) \leftarrow s_1$ that minimizes $J_1(L)$
    end for
end procedure
```

Complexity Analysis of Original DP and Database-DP We compare the computational complexity of the original DP and the proposed database-DP. The number of iterations required for DP, denoted by Ψ_{DP}, is determined as

$$\Psi_{DP} = K(L+1)^T \in O(L^T), \tag{13}$$

since each slot has at most $(L+1)$ possible choices of s_t, i.e., $0, 1, \ldots, L$. On the other hand, the number of iterations required for the database-DP, denoted by Ψ_{DB}, is given as

$$\Psi_{DB} = K\left(2(L+1) + (T-2)\sum_{j=1}^{L+1} j\right) \in O(L^2 T), \tag{14}$$

In Eq. (14), the first term is related to the slot $t = T$ which has $(L+1)$ cases to consider since $0 \leq l_T \leq L$ and $s_T^* = l_T$, and the slot $t = 1$ which has $l_1 = L$ thus leading to $(L+1)$ possible

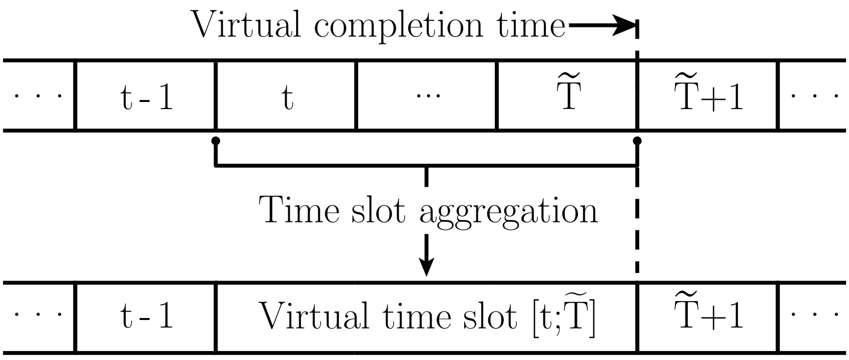


Figure 8: Time slot aggregation for approximate dynamic programming

choices of s_1. The second term concerns with $1 < t < T$, i.e., a total of $(T-2)$ slots, where the sum implies that when $l_t = j-1$ (corresponding to the j's column of a matrix), there are j ways of choosing s_t such that $s_t = 0, 1, \ldots, (j-1)$.

Eqs. (13) and (14) show that database-DP much enhances the complexity of the original DP from $O(L^T)$ to $O(L^2T)$.[6] The computational requirement, however, still increases fast as L and T grow. To further reduce the complexity while maintaining reasonably good performance, we propose an approximate DP algorithm in the next subsection.

ADP: One-step Lookahead Approximation of DP

The one-step lookahead approach is one of the approximation methods to mitigate the complexity of DP computation, by making a decision based on lookahead of a limited number of stages with approximate cost-to-go functions [31]. In Eq. (9), the cost-to-go function $J_t(l_t)$ incorporates the time slots from t to T, and hence it should consider all possible combinations according to $l_{t'}$'s and $Z_{t'}$'s for $t' \geq t$, whose number grows rapidly as t gets smaller. The one-step lookahead scheme, however, replaces the expectation of future unknown behaviors, i.e., $\underset{Z_{t+1}}{\mathbb{E}}\left[J_{t+1}\left(l_t - s_t\right)\right]$ in Eq. (9), with an approximate cost-to-go function, thus eliminating the need for enumerating all of the combinatorial cases.

To describe the proposed one-step lookahead approximate DP (ADP) algorithm, we define three concepts – *virtual completion time*, *virtual time slot*, and *virtual channel state*. The virtual completion time, denoted by $\widetilde{T}$, is the time slot at which the offloading process is believed to be completed, i.e., $l_{\widetilde{T}} > 0$ and $l_{\widetilde{T}+1} = 0$. The virtual time slot, denoted by $[t;\widetilde{T}]$, aggregates $(\widetilde{T}-t+1)$ time slots $t, t+1, \ldots, \widetilde{T}$ into one, as shown in Fig. 8. Then, the virtual channel state, denoted by $1/\zeta_{t;\widetilde{T}}$, is the channel state of the virtual time slot, reflecting the combined effect of the channel states from slots $t, \ldots, \widetilde{T}$.

[6] Note that $O(L^2T) < O(L^T)$ as long as $T > 2$ and $L > 3$, which is mostly likely in practical scenarios.

Theorem 1 *$\zeta_{t;\widetilde{T}}$ is derived as*[7]

$$\zeta_{t;\widetilde{T}} = \begin{cases} \left(Z_t^{\frac{1}{n-1}} + \mathbb{E}\left[\zeta_{t+1;\widetilde{T}}\right]^{-\frac{1}{n-1}} \right)^{1-n}, & \textit{when } t < \widetilde{T}, \\ Z_t^{-1}, & \textit{when } t = \widetilde{T}. \end{cases} \tag{15}$$

Proof 1 *We assume that the offloading process is completed exactly at slot $\widetilde{T}$, and define $\zeta_{t;\widetilde{T}}$ as a reciprocal of the virtual channel state combining slots $t, \ldots, \widetilde{T}$. For $t = \widetilde{T}$, it is certain that $\zeta_{t;\widetilde{T}} = 1/Z_{\widetilde{T}}$, and thus the approximate cost-to-go becomes the exact cost-to-go such as*

$$\tilde{J}_{\widetilde{T}}(l_{\widetilde{T}}) = a \cdot l_{\widetilde{T}}{}^n \Big/ \left(\zeta_{\widetilde{T};\widetilde{T}}\right)^{-1} + b. \tag{16}$$

When $t = \widetilde{T} - 1$, the approximate cost-to-go is given as[8]

$$\begin{aligned} \tilde{J}_{\widetilde{T}-1}(l_{\widetilde{T}-1}) &= \min_{0 \le s_{\widetilde{T}-1} < l_{\widetilde{T}-1}} \left\{ a \frac{s_{\widetilde{T}-1}{}^n}{Z_{\widetilde{T}-1}} + b + \mathop{\mathbb{E}}_{\zeta_{\widetilde{T};\widetilde{T}}} \left[\tilde{J}_{\widetilde{T}} \left(l_{\widetilde{T}-1} - s_{\widetilde{T}-1} \right) \right] \right\} \\ &= \min_{0 \le s_{\widetilde{T}-1} < l_{\widetilde{T}-1}} \left\{ a \frac{s_{\widetilde{T}-1}{}^n}{Z_{\widetilde{T}-1}} + a \frac{(l_{\widetilde{T}-1} - s_{\widetilde{T}-1})^n}{1 \Big/ \mathop{\mathbb{E}}_{\zeta_{\widetilde{T};\widetilde{T}}} \left[\zeta_{\widetilde{T};\widetilde{T}} \right]} + 2b \right\} \end{aligned} \tag{17}$$

where the function inside the curly brackets is minimized at

$$\tilde{s}_{\widetilde{T}-1} = l_{\widetilde{T}-1} \Big/ \left\{ 1 + \left(Z_{\widetilde{T}-1} \mathbb{E}_{\zeta_{\widetilde{T};\widetilde{T}}} \left[\zeta_{\widetilde{T};\widetilde{T}} \right] \right)^{-\frac{1}{n-1}} \right\} \tag{18}$$

which can be easily shown via differentiation. Note that Eq. (18) *satisfies the range $0 \le s_{\widetilde{T}-1} < l_{\widetilde{T}-1}$. By applying Eq.* (18) *to Eq.* (17)*, we obtain*

$$\tilde{J}_{\widetilde{T}-1}(l_{\widetilde{T}-1}) = a \cdot l_{\widetilde{T}-1}{}^n \Big/ \left(\zeta_{\widetilde{T}-1;\widetilde{T}}\right)^{-1} + 2b, \tag{19}$$

$$\zeta_{\widetilde{T}-1;\widetilde{T}} = \left(Z_{\widetilde{T}-1}{}^{\frac{1}{n-1}} + \mathbb{E}_{\zeta_{\widetilde{T};\widetilde{T}}} \left[\zeta_{\widetilde{T};\widetilde{T}} \right]^{-\frac{1}{n-1}} \right)^{-(n-1)}. \tag{20}$$

In addition, (18) *is rewritten using* (20) *as*

$$\tilde{s}_{\widetilde{T}-1} = l_{\widetilde{T}-1} (Z_{\widetilde{T}-1} \zeta_{\widetilde{T}-1;\widetilde{T}})^{\frac{1}{n-1}}. \tag{21}$$

By repeating the steps for $t = \widetilde{T} - 2$, we obtain

$$\tilde{J}_{\widetilde{T}-2}(l_{\widetilde{T}-2}) = a \cdot l_{\widetilde{T}-2}{}^n \Big/ \left(\zeta_{\widetilde{T}-2;\widetilde{T}}\right)^{-1} + 3b, \tag{22}$$

$$\zeta_{\widetilde{T}-2;\widetilde{T}} = \left(Z_{\widetilde{T}-2}{}^{\frac{1}{n-1}} + \mathbb{E}_{\zeta_{\widetilde{T}-1;\widetilde{T}}} \left[\zeta_{\widetilde{T}-1;\widetilde{T}} \right]^{-\frac{1}{n-1}} \right)^{-(n-1)}. \tag{23}$$

It can be shown by induction that the same structure in Eqs. (20) *and* (23) *appears for any slot index within $t, \ldots, \widetilde{T} - 1$. This implies that the virtual time slot $[t; \widetilde{T}]$ can be modeled to have a channel state given in Eq.* (15) *and the latency cost of $(\widetilde{T} - t + 1)b$, which completes the derivation.*

[7] The proof of the theorem is inspired by [18].

[8] Note that the range $0 \le s_{\widetilde{T}-1} < l_{\widetilde{T}-1}$ excludes $l_{\widetilde{T}-1}$ since offloading should complete at slot $\widetilde{T}$.

Using the FSMC model, $\mathbb{E}[\zeta_{t+1;\widetilde{T}}]$ in Eq. (15) is derived for $Z_t = \overline{\gamma}_k$ as

$$\mathbb{E}\left[\zeta_{t+1;\widetilde{T}}\right] = \begin{cases} \mathbb{1}_{\{k\neq 1\}} P_{k,k-1}\left(\overline{\gamma}_{k-1}{}^{\frac{1}{n-1}} + \mathbb{E}\left[\zeta_{t+2;\widetilde{T}}\right]^{-\frac{1}{n-1}}\right)^{1-n} & \\ \quad + P_{k,k}\left(\overline{\gamma}_{k}{}^{\frac{1}{n-1}} + \mathbb{E}\left[\zeta_{t+2;\widetilde{T}}\right]^{-\frac{1}{n-1}}\right)^{1-n} & \\ \quad + \mathbb{1}_{\{k\neq K\}} P_{k,k+1}\left(\overline{\gamma}_{k+1}{}^{\frac{1}{n-1}} + \mathbb{E}\left[\zeta_{t+2;\widetilde{T}}\right]^{-\frac{1}{n-1}}\right)^{1-n}, & \text{when } t < \widetilde{T}-1 \\ \mathbb{1}_{\{k\neq 1\}} P_{k,k-1}\frac{1}{\overline{\gamma}_{k-1}} + P_{k,k}\frac{1}{\overline{\gamma}_{k}} + \mathbb{1}_{\{k\neq K\}} P_{k,k+1}\frac{1}{\overline{\gamma}_{k+1}}, & \text{when } t = \widetilde{T}-1 \end{cases} \tag{24}$$

Note that we determine $\mathbb{E}[\zeta_{t+1;\widetilde{T}}]$ in Eq. (15) only by Eq. (24), not by recursively applying Eq. (15) to itself.

Based on the aforementioned concepts, we now sketch the steps in the proposed ADP algorithm.

- At slot t, we consider $(T-t+1)$ possible cases: the transmission of the remaining l_t bits would be completed at slot $\widetilde{T}$, where $\widetilde{T} = t, \ldots, T$.
- For each possible $\widetilde{T}$, we aggregate slots $t+1, \ldots, \widetilde{T}$ into a virtual time slot $[t+1;\widetilde{T}]$. We treat the virtual slot as a single slot with a channel state $1/\zeta_{t+1;\widetilde{T}}$ incurring the latency cost of $(\widetilde{T}-t)b$, which has been proven in Theorem 1. Then, the problem reduces to the system consisting of two slots, t and $[t+1;\widetilde{T}]$, eliminating the need for the recursive iterations in the original DP.
- We find the optimal s_t for each possible $\widetilde{T}$, and compare thus-derived s_t's to find the one with the minimal (approximate) cost-to-go. The resulting s_t becomes the (suboptimal) solution of the ADP for slot t.

The algorithm is now formally described as follows.

Deriving the $\widetilde{T}$-dependent solution $\hat{s}_t^*$ The first step is to find the suboptimal value for a fixed virtual completion time. Let's define $\tilde{J}_{t;\widetilde{T}}(l_t)$ as the approximate cost-to-go function for a fixed $\widetilde{T}$, such as

$$\tilde{J}_{t;\widetilde{T}}(l_t) = \begin{cases} \min\limits_{0\leq s_t < l_t} \left\{ a\frac{s_t{}^n}{Z_t} + b + \mathbb{E}_{\zeta_{t+1;\widetilde{T}}}\left[\tilde{J}_{t+1}(l_t - s_t)\right]\right\}, & \text{when } t < \widetilde{T} \\ a\frac{l_t{}^n}{Z_t} + b, & \text{when } t = \widetilde{T} \end{cases} \tag{25}$$

where $\tilde{J}_{t+1}(l_t - s_t)$ assumes that the transmission of the remaining $(l_t - s_t)$ bits is completed at slot $\widetilde{T}$. In Theorem 1, $\tilde{J}_{t+1}(l_t - s_t)$ has been derived as

$$\tilde{J}_{t+1}(l_t - s_t) = a \cdot (l_t - s_t)^n / \left(\zeta_{t+1;\widetilde{T}}\right)^{-1} + (\widetilde{T} - t)b. \tag{26}$$

By applying Eq. (26) to Eq. (25) and taking derivative with s_t, we obtain s_t that minimizes the approximate cost-to-go, denoted by $\hat{s}^*_{t;\widetilde{T}}$, such as

$$\hat{s}^*_{t;\widetilde{T}} \;=\; \begin{cases} l_t \cdot \left(Z_t \cdot \zeta_{t;\widetilde{T}}\right)^{\frac{1}{n-1}}, & \text{when } 1 \le t < \widetilde{T}, \\ l_t, & \text{when } t = \widetilde{T}. \end{cases} \tag{27}$$

By applying Eq. (27) to Eq. (25), we finally obtain

$$\tilde{J}_{t;\widetilde{T}}(l_t) \;=\; a \cdot {l_t}^n / \big(\zeta_{t;\widetilde{T}}\big)^{-1} + (\widetilde{T} - t + 1)b. \tag{28}$$

Deriving the suboptimal solution $\tilde{s}^*_t$ The second step is to compare $\tilde{J}_{t;\widetilde{T}}(l_t)$, $\forall \widetilde{T}$ to find the optimal $\widetilde{T}$ minimizing the approximate cost-to-go. Denoting such $\widetilde{T}$ by $\widetilde{T}^*$, we have

$$\widetilde{T}^* = \underset{t \le \widetilde{T} \le T}{\operatorname{argmin}} \left\{ \tilde{J}_{t;\widetilde{T}}(l_t) \right\}. \tag{29}$$

Then, the (sub)optimal amount of bits to transmit according to ADP at slot t, denoted by $\tilde{s}^*_t$, is determined as

$$\tilde{s}^*_t = \hat{s}^*_{t;\widetilde{T}^*} \tag{30}$$

Finally, the achieved total cost by the (sub)optimal policy set, J^*_{ADP}, is determined as

$$J^*_{ADP} \;=\; \sum_{t=1}^{T} \left\{ a \frac{(\tilde{s}^*_t)^n}{Z_t} + b \mathbb{1}_{\{\tilde{l}^*_t > 0\}} \right\}, \tag{31}$$

$$\tilde{l}^*_t \;=\; \tilde{l}^*_{t-1} - \tilde{s}^*_{t-1}, \forall t > 1, \quad \tilde{l}^*_1 = L. \tag{32}$$

Complexity Analysis of Approximate-DP The complexity of ADP consists of two parts: offline complexity in building the lookup table of all possible $\zeta_{t;T}$'s by Eqs. (15) and (24), and online complexity at time t to determine $\tilde{s}^*_t$ by Eq. (29). Specifically, the number of iterations required for ADP is

$$\Psi_{ADP} = 2KT + \sum_{t=1}^{T} (T - t + 1) \in O(T^2), \tag{33}$$

where the first term is the offline complexity in calculating $\zeta_{t;T}$ in the descending order of t, with K possible channel states in each of Eqs. (15) and (24).[9] The second term is the total online complexity combining each t's complexity in solving Eq. (29). That is, the ADP algorithm runs fast in real-time since at each t only $(T - t + 1)$ iterations are necessary to run.

The ADP's complexity does not depend on L and thus large tasks won't degrade the performance. Considering that $L^2 \gg T, K$ in practice, e.g., $L = 81,920$ bits in code offloading [14], the ADP's offline complexity in building the lookup table ($2KT$) is much less than the database-DP ($O(L^2T)$). In addition, the lookup table is lightweight to store in a mobile device since $2KT$ is also related to the size of the table. Moreover, although the lookup table is built for $\widetilde{T} = T$, it can be reused by any other $\widetilde{T}$ because $\zeta_{t;\widetilde{T}}$ depends only on the number of time slots between t and $\widetilde{T}$. For example, $\zeta_{t;T} = \zeta_{t';\widetilde{T}}$ when $T - t = \widetilde{T} - t'$.

[9] $\mathbb{E}[\zeta_{t+1;\widetilde{T}}]$ is determined by Eq. (24) via table lookup of pre-built entries for larger t.

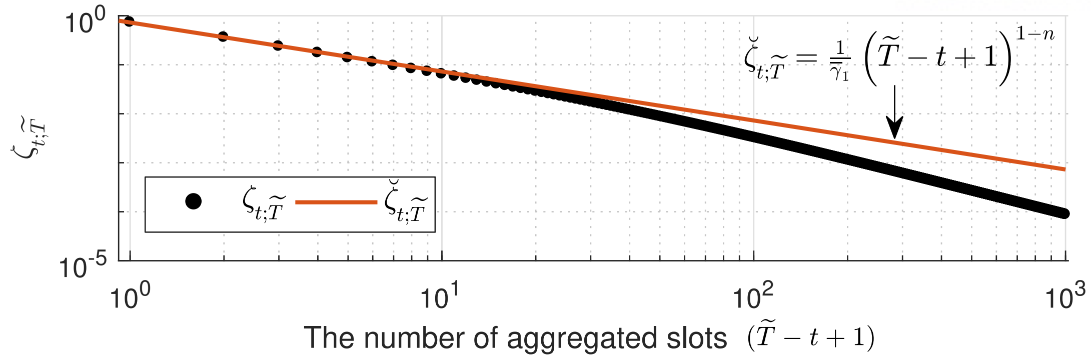


Figure 9: Exact values of $\zeta_{t;\widetilde{T}}$ and an approximate line of $\breve{\zeta}_{t;\widetilde{T}}$

ADPe: Further Enhancement of ADP

Even though ADP alleviates complexity from $O(L^2T)$ to $O(T^2)$, it is still desirable to further reduce the complexity for real-time operations. Since ADP's complexity mostly arises when obtaining $\widetilde{T}^*$, we present an enhanced approximate DP (ADPe) algorithm that can reduce the complexity in Eq. (29) in case of slow-fading where the channel state tends to remain the same with high probability. The slow-fading assumption becomes reasonable in some practical cases, such as pedestrians walking along the street. In LTE, for example, one of the recommended performance evaluation scenarios is to set the maximum Doppler frequency to 5 Hz [41], which corresponds to the situation where a person takes a walk at 0.8 m/s while using a Band 3 network that is the most popular band in the world [49].

The ADPe algorithm consists of three steps. The first step is to build a table of $\zeta_{t;\widetilde{T}}$ for all channel states by Eqs. (15) and (24), while approximating $\zeta_{t;\widetilde{T}}$ as follows. Assuming $P_{k,k} \approx 1, \forall k$, Eq. (24) can be approximated for $Z_t = \overline{\gamma}_k$ as

$$\mathbb{E}\left[\zeta_{t+1;\widetilde{T}}\right] \approx \begin{cases} \left(\overline{\gamma}_k{}^{\frac{1}{n-1}} + \mathbb{E}\left[\zeta_{t+2;\widetilde{T}}\right]^{-\frac{1}{n-1}}\right)^{1-n}, & \text{when } t < \widetilde{T}-1 \\ \overline{\gamma}_k{}^{-1}, & \text{when } t = \widetilde{T}-1 \end{cases} \tag{34}$$

and hence $\zeta_{t;\widetilde{T}}$ can be approximated to a power function as

$$\zeta_{t;\widetilde{T}} \approx \breve{\zeta}_{t;\widetilde{T}} \triangleq \frac{1}{Z_t}\left(\widetilde{T}-t+1\right)^{1-n}. \tag{35}$$

Fig. 9 presents an illustrative example of such an approximation which confirms that $\breve{\zeta}_{t;\widetilde{T}}$ shows a similar tendency to $\zeta_{t;\widetilde{T}}$ with a reasonable degree of accuracy. Please note that the gap between the two cases for large $(\widetilde{T}-t+1)$ is in fact insignificant since the figure is drawn in log scale.

The second step is to derive $\widetilde{T}^*$ in Eq. (29) using $\breve{\zeta}_{t;\widetilde{T}}$ such as

$$\widetilde{T}^* \approx \breve{T}^* \triangleq \underset{t\leq\widetilde{T}\leq T}{\operatorname{argmin}} \left\{\breve{J}_{t;\widetilde{T}}(l_t)\right\} \tag{36}$$

where $\breve{J}_{t;\widetilde{T}}$ is the approximate cost-to-go function by replacing $\zeta_{t;\widetilde{T}}$ with $\breve{\zeta}_{t;\widetilde{T}}$ in Eq. (28), such as

$$\tilde{J}_{t;\widetilde{T}}(l_t) \approx \breve{J}_{t;\widetilde{T}} \triangleq a \cdot l_t{}^n/\left(\breve{\zeta}_{t;\widetilde{T}}\right)^{-1} + (\widetilde{T}-t+1)b. \tag{37}$$

Then, $\breve{T}^*$ is obtained as follows by differentiating Eq. (37):

$$\breve{T}^* = \begin{cases} t, & \text{when } \dot{T} < t \\ \dot{T}, & \text{when } t \le \dot{T} \le T \\ T, & \text{when } \dot{T} > T \end{cases} \tag{38}$$

where

$$\dot{T} = t - 1 + \left(\frac{b \cdot Z_t}{a(n-1) \cdot {l_t}^n} \right)^{-\frac{1}{n}}. \tag{39}$$

Lastly, based on Eq. (30), the (sub)optimal value of $\tilde{s}_t^*$ is approximately obtained as

$$\tilde{s}_t^* \approx \breve{s}_t^* \triangleq \hat{s}^*_{t;[\breve{T}^*]} \tag{40}$$

where $[\breve{T}^*]$ is round-half-to-even of $\breve{T}^*$ that is necessary since the table of $\zeta_{t;\widetilde{T}}$ allows only integers as its indices. By the same token, we define the approximate cost-to-go function provided $\breve{T}^*$ as

$$\breve{J}_{t;\breve{T}^*}(l_t) = a \cdot {l_t}^n / \big(\zeta_{t;[\breve{T}^*]}\big)^{-1} + (\breve{T}^* - t + 1)b. \tag{41}$$

Then, the achieved total cost by the (sub)optimal policy set, J^*_{ADPe}, is determined as

$$\begin{aligned} J^*_{ADPe} &= \sum_{t=1}^{T} \left\{ a \frac{(\breve{s}_t^*)^n}{Z_t} + b \mathbb{1}_{\{\breve{l}_t^* > 0\}} \right\}, && (42) \\ \breve{l}_t^* &= \breve{l}_{t-1}^* - \breve{s}_{t-1}^*, \forall t > 1, \quad \breve{l}_1^* = L. && (43) \end{aligned}$$

Complexity Analysis of Enhanced Approximate-DP The ADPe algorithm incurs low complexity because all the parameters except $\zeta_{t;\widetilde{T}}$ are derived in their closed forms. Therefore, its computational complexity originates only from building the tables, and we obtain

$$\Psi_{ADPe} = 2KT \in O(T). \tag{44}$$

In other words, unlike in ADP, ADPe does not incur any computation in the middle of offloading.

2.5 QoE-aware Service Class Selection

This section focuses on which MECOS service class should be selected for computation offloading to achieve best QoE. In particular, the offloaded task manager (introduced earlier in Section 2.2) needs to choose the MECOS class to incur the minimum cost during the overall computation offloading procedure. To do so, Sections 2.5 and 2.5 derive the QoE-aware cost at the processing and data transmission phases, and then Section 2.5 formulates the optimal strategy of choosing the best service class for computation offloading.

Offloading Cost at Processing Phase

We hereby derive each of energy, latency, and monetary costs in the processing phase. First, the latency $D_{proc}(q)$ and energy consumption $E_{proc}(q)$ are determined as

$$\begin{aligned} D_{proc}(q) &= \frac{R_{app} \cdot L}{H_q} \times \epsilon, & (45) \\ E_{proc}(q) &= D_{proc}(q) \cdot P_{proc}, & (46) \end{aligned}$$

where P_{proc} is the consumed power by the mobile device during the processing phase. In what follows, we assume $P_{proc} = 0$ because the local device may consume energy during the processing of an offloaded task at the cloud not for the task's application but for some other applications like background tasks, for which the offloaded task is not responsible.

The processing phase may start in the middle of an ϵ period, and thus the monetary cost $F_{proc}(q)$ due to the service charge of MECOS is derived as

$$\begin{aligned} F_{proc}(q) &= \left(u(\epsilon) + \left\lceil \frac{R_{app} \cdot L}{H_q} - u(\epsilon) \right\rceil \right) \times \delta_q, & (47) \\ u(\epsilon) &= \frac{\lambda - (\breve{T}^* \bmod \lambda)}{\lambda}, & (48) \end{aligned}$$

where $u(\epsilon)$ is the ratio of the remaining portion of the period to ϵ, and $\bmod$ indicates the modulo operation.

Then, similar to the definition of Eq. (8), the total cost at the processing phase is determined as

$$\phi_{proc}(q) = w_1 \frac{E_{proc}(q)}{E_L} + w_2 \frac{D_{proc}(q)}{D_L} + w_3 \frac{F_{proc}(q)}{F_L} \tag{49}$$

where $w_1 = 1 - \alpha B$, $w_2 = w_{QoE} \cdot \alpha B$, and $w_3 = w_{QoE} \cdot w_m$.

Offloading Cost at Data Transmission Phase

The mobile user has to pay the usage charge for two types of services: one provided by the cloud service provider, and the other by the wireless carrier for the network usage to exchange data with the cloud service provider. When the charging rate of the wireless carrier is μ per bit, the monetary cost is determined as

$$F_{tx} = L \cdot \mu. \tag{50}$$

Hence, the total cost at the data transmission phase becomes

$$\phi_{tx}(q) = \breve{J}_{1;\breve{T}^*}(L) + w_3 \frac{F_{tx}}{F_L}. \tag{51}$$

Note that $\breve{J}_{1;\breve{T}^*}(L)$ is a function of q, because the monetary cost function in Eq. (7) depends on q, and the deadline for the transmission phase (i.e., T) is also a function of q as shown later in Eq. (54) of Section 2.5.

Optimal Service Class Selection Problem

Now we consider an optimal selection problem where a mobile user wants to determine the best of Q MECOS classes to lease so as to minimize the overall cost incurred by computation offloading. We first determine the deadline for the data transmission phase, and then find the best strategy based on the computation offloading cost of the data transmission and processing phases.

For given class q, ϕ_{proc} becomes constant while ϕ_{tx} varies with transmission scheduling. In addition, the transmission deadline T depends on q, as shown as follows. First, the transmission deadline due to the task-specific time constraint D_L, denoted by $T_1(q)$, is given as

$$T_1(q) = \left\lfloor \frac{D_L - (D_{proc}(q) + D_{rx})}{\tau} \right\rfloor . \tag{52}$$

Next, the transmission deadline due to the monetary constraint F_L, denoted by $T_2(q)$, is given as

$$T_2(q) = \left\lfloor \{F_L - (F_{tx} + F_{proc}(q) + F_{rx})\} \frac{\lambda}{\delta_q} \right\rfloor \tag{53}$$

where $F_{rx} = L_{rx} \cdot \mu$. Finally, the deadline of the transmission phase is determined as

$$T(q) = \min \{T_1(q), T_2(q)\} . \tag{54}$$

For given $T(q)$, the best class q^* is determined as

$$q^* = \underset{q \in \mathbb{Q}}{\operatorname{argmin}} \ (\phi_{tx}(q) + \phi_{proc}(q)) . \tag{55}$$

where $T(q)$ is applied to $\phi_{tx}(q)$.

Finally, the overall QoE-aware cost by computation offloading performed with class q^*, denoted by $\Phi_C(q^*)$, is given as

$$\Phi_C(q^*) = \phi_{tx}(q^*) + \phi_{proc}(q^*). \tag{56}$$

Local Computing vs. Offloaded Computing

The decision maker, introduced earlier in Section 2.2, determines whether a given task should be offloaded or not by comparing the overall cost of local computing with that of offloading, i.e., $\Phi_C(q^*)$ in Eq. (56). This section completes the discussion on the decision maker by deriving the local computing cost and by describing how to decide between local and offloaded computing.

In fact, deriving the local computing cost is not trivial due to the heterogeneous characteristics of mobile processors and various existing algorithms to assign tasks to the processors. A mobile processor usually possesses multiple heterogeneous cores such as high-performance cores for computation-intensive tasks and energy-efficient cores for other types of tasks [50], so that it can carry out numerous background/user applications simultaneously. Moreover, there exist various algorithms on how to utilize such cores, such as dynamic voltage and frequency scaling (DVFS) and dynamic power management (DPM) [36,37,51].

Considering aforementioned aspects, we assume that the offloadable task is assigned a high-performance core which processes the task exclusively until the completion of the task (*not* until the completion of the task's application, though). We believe the assumption is reasonable to some extent because the offloadable task is usually latency-sensitive (i.e., with a specific time deadline) and thus the mobile device needs to process it with higher priority, e.g., by assigning a dedicated core with high performance. In this case, the processing delay is derived as

$$D_M = (R_{app}^M \times L)/H_M, \tag{57}$$

where H_M is the frequency of a core in Hertz. In addition, the energy consumption during the processing is given as

$$E_M = D_M \times P_M, \tag{58}$$

where P_M is the power consumption when the core is fully utilized to process the offloadable task. Then, the QoE-aware local computing cost is defined similar to Φ_C as

$$\Phi_M = (1 - \alpha B)\frac{E_M}{E_L} + w_{QoE}\alpha B\frac{D_M}{D_L}, \tag{59}$$

where the monetary cost is excluded since local computing does not incur any MECOS-related charges.

Finally, the decision maker decides whether computation offloading is performed or not, according to the following criterion:

$$\begin{cases} \text{local computing}, & \text{when } \Phi_M \le \Phi_C(q^*), \\ \text{offloading to } q^*, & \text{when } \Phi_M > \Phi_C(q^*). \end{cases} \tag{60}$$

In Section 2.6, we will consider the proposed decision making in Eq. (60) for a couple of real mobile applications.

2.6 Performance Evaluation

In this section, we evaluate the performance of the proposed DP, ADP, and ADPe algorithms and the service class selection algorithm via extensive simulations. Our simulation environment is presented as follows.

- **Wireless Channel**: We consider a wireless channel with the bandwidth of 20 MHz, the maximum Doppler frequency f_m of 5 Hz, and $N_0W = -96$ dBm (according to the Johnson-Nyquist noise with 5 dB noise figure [41,52]). We also consider $\gamma_0 = 16$ dB, $\tau = 1$ ms, and $P_{tx}^{nom} = 100$ mW. Then, the channel is modeled as an FSMC with 15 states.

- **Mobile Device**: We assume a mobile device has $C_{Batt} = 11.55$ Wh [53], $H_M = 1.8$ GHz [54], and $P_M = 750$ mW [55].

Table 2: $T(q)$ of APP1 and APP2

	APP1	APP2
$q = 1$	623	1719
$q = 2$	699	1936
$q = 3$	754	2091

Table 3: Transmission deadline of APP2*

	$T_1(q)$	$T_2(q)$	$T(q)$
$q = 1$	1719	2037	1719
$q = 2$	1936	806	806
$q = 3$	2091	276	276

- **MECOS Service Classes**: Based on a commercial cloud service [56], we consider three service classes ($Q = 3$) where $q = 1$ is the cheapest but slowest whereas $q = 3$ is the most expensive but fastest, such as $H_q = 2.5, 3, 3.5$ GHz, respectively. We set $\delta_q = 4.37 \times 10^{-32} \times {H_q}^3$ by adopting the Cobb-Douglas demand model [57] which provides the relationship between the consumer demand for a service, price, and delivery lead time while assuming the same demand for all of the Q classes.[10] Note that the equation of δ_q has been set up according to the price introduced in [56], and we assume $\theta = 1/1.7$ [45] and $\epsilon = 100$ ms.

- **Application Characteristics**: Based on [43] and [58], we consider two applications, a face recognition application with $L = 7.5$ KB and $R_{app} = 31,680$ (cycles/bit) and a video game application with $L = 256$ KB and $R_{app} = 2,640$ (cycles/bit), each of which represents a resource intensive application and a latency sensitive application, respectively [14]. Regarding L_{rx}, we set $L_{rx} = 1$ KB for the face recognition application since its result includes lightweight information like the face's location in the image and the person's name, and $L_{rx} = 256$ KB for the video game application (which is the same as L) considering that the application's state takes a major portion of both input and output data [14]. Also, we assume that the task-specific deadline D_L equals to the processing delay of local computing D_M.

- **Misc.**: Based on [48] and [59], we adopt $n = 2.67$ and $m = 1.054 \times 10^{-33}$ to calculate the transmission power. To calculate D_{rx} of each application, we consider the downlink data rate of 39 Mbps which corresponds to one of the achievable data rates at $\gamma_0 = 16$ dB [60]. We also consider $F_L = 0.01$ which represents a situation where a user pays $9/month for computation offloading service and performs offloading 30 times per day on average. In addition, $\mu = 10$ $/GB [61].

Based on this environment, we consider three experimental scenarios, APP1, APP2, and APP2*, in Section 2.6 and thereafter. APP1 indicates the face recognition application with $F_L = 0.01$, whereas APP2 and APP2* represent the video game application with $F_L = 0.01$ and

[10] The equal-demand assumption becomes realistic when the cloud service provider wants to evenly utilize its resources.

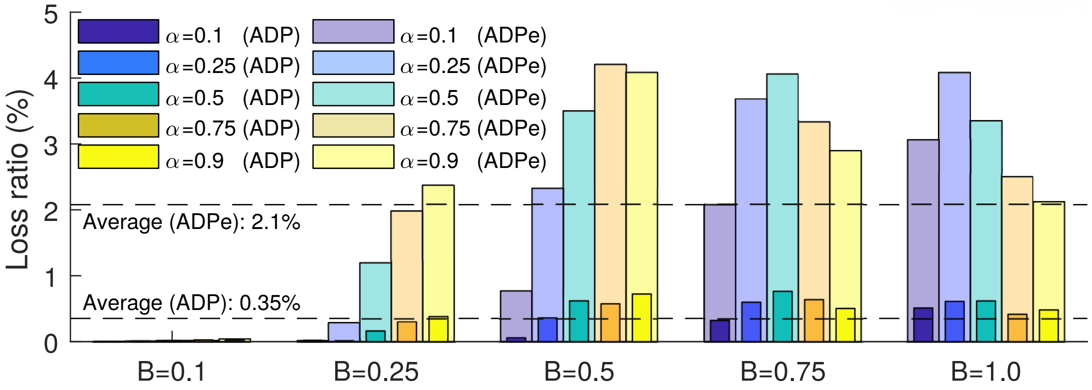


Figure 10: Performance loss of ADP and ADPe compared to optimal DP

$F_L = 0.004906$, respectively. Note that APP2* is considered only in Section 2.6 to show the impact of the monetary budget on class selection, where $F_L = 0.004906$ is a fine-tuned condition at which APP2* starts to behave differently from APP2. According to Eq. (54), the transmission phase deadlines of APP1 and APP2 are always decided by the task-specific time constraint D_L (i.e., by $T_1(q)$) as presented in Table 2, whereas the transmission phase deadline of APP2* is decided by either $T_1(q)$ or $T_2(q)$ as presented in Table 3. As a result, in APP1 and APP2, a faster service class (i.e., larger q) assures a more extended time budget. On the contrary, in APP2*, a faster class more restricts the time budget.

For each scenario, we generated 1,000 realizations of the described channel and observed the average behavior. To properly capture the tradeoff relationship, we considered a user tendency range of $0.1 \leq \alpha \leq 0.9$ excluding extreme cases like $\alpha = 0$ and $\alpha = 1$, and a battery state range of $0.1 \leq B \leq 1$. For any combination of α and B, it turns out that the decision maker always decides to perform computation offloading by applying Eq. (60), regardless of APP1, APP2, and APP2*. This is primarily because the chosen γ_0 helps computation offloading achieve smaller energy and latency costs than local computing; local computing, however, would be chosen when the average channel condition deteriorates (i.e. small γ_0). Therefore, in the sequel, we concentrate on the performance of the offloaded task manager, service class selection, and transmission scheduling.

Goodness in Performance of ADP and ADPe

To evaluate how much deviated the performance of the proposed suboptimal transmission algorithms is from the optimal performance, we compare the database-DP with ADP and ADPe algorithms with a sample application[11] with $L = 1$ KB, $T = 1{,}000$ slots, $D_L = 2$ seconds, and $w_{QoE} = 7.77 \times 10^{-16}$. The performance loss in total achieved cost is measured by the *loss ratio*,

[11] Due to the complexity of the database-DP, APP1, APP2, and APP2* are inadequate to consider for this type of evaluation.

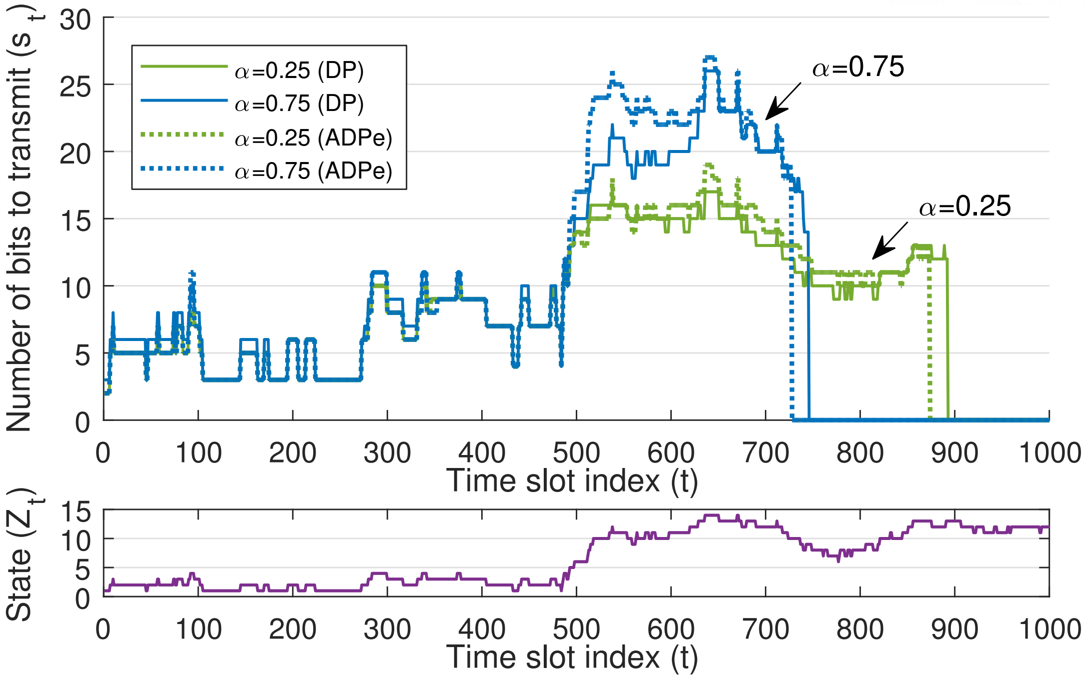


Figure 11: An optimal schedule: DP vs. ADPe (when $B = 0.5$)

which is defined as

$$\text{loss ratio (\%)} = \left| \frac{J^*_{sub} - J^*_{DP}}{J^*_{DP}} \right| \times 100 \tag{61}$$

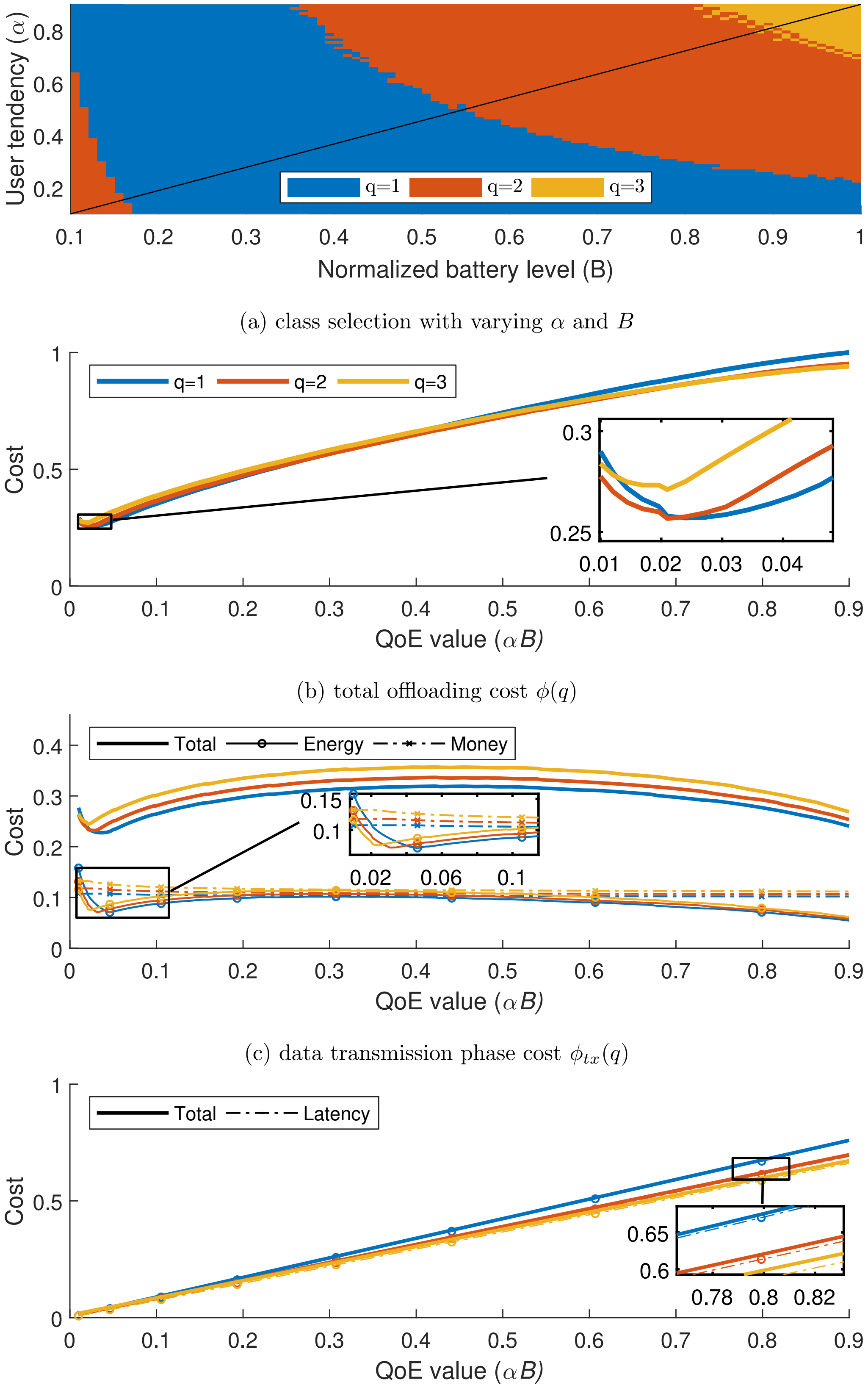
where J^*_{sub} is either J^*_{ADP} or J^*_{ADPe}.

Fig. 10 illustrates the measured loss ratio, by varying α and B such that $\alpha \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$ and $B \in \{0.1, 0.25, 0.5, 0.75, 1.0\}$. As observed, the performance deviations of ADP and ADPe due to approximation are only 0.35% and 2.1% on average and well bounded below 0.77% and 4.21% respectively, confirming their near-optimal performance. The ADP and ADPe are shown to perform worse with larger αB, which is because in the delay-sensitive case (i.e., large α), the two algorithms try to transmit more data at "good" slots than the DP does, aiming at shortening the completion time — which is well illustrated in Fig. 11. From now on, we use the ADPe algorithm to evaluate the impact of QoE factors on real applications.

MECOS Class Selection

In this subsection, we evaluate the impact of energy, latency, and monetary costs on the MECOS class selection problem for the three aforementioned experimental scenarios. First, Fig. 12 presents APP1's results, where sub-figure (a) shows the selected service class with various combinations of $0.1 \leq \alpha \leq 0.9$ and $0.1 \leq B \leq 1$, and sub-figures (b), (c), (d) show the related offloading costs according to varying αB along the solid straight line in Fig. 12a.[12] In particular,

[12] Note that the straight line implies the one between ($\alpha = 0.1, B = 0.1$) and ($\alpha = 0.9, B = 1$). In addition, the same color convention is applied in Figs. 12, 13, and 14 as follows: blue for $q = 1$, red for $q = 2$, yellow for $q = 3$.



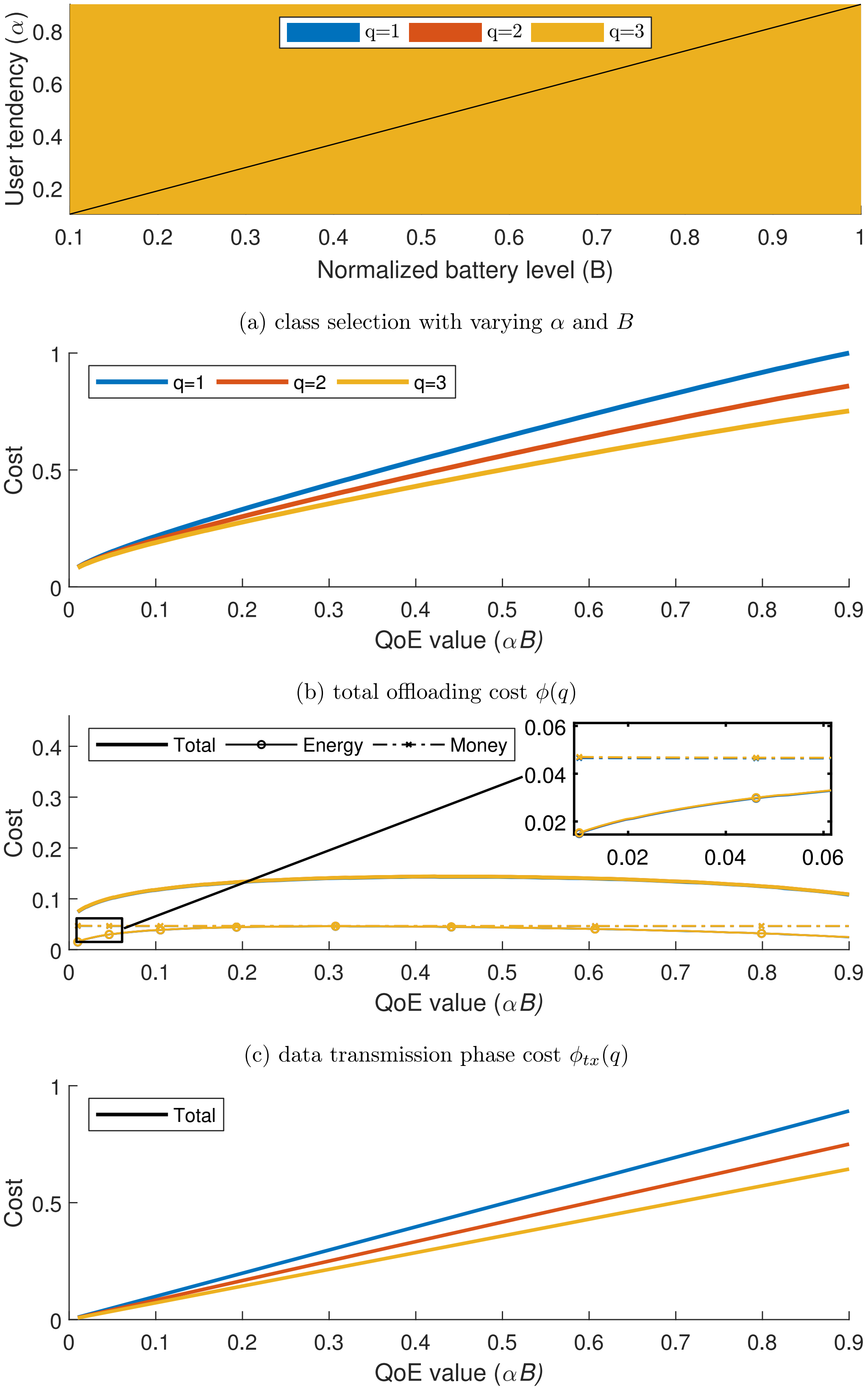
(a) class selection with varying α and B

(b) total offloading cost $\phi(q)$

(c) data transmission phase cost $\phi_{tx}(q)$

(d) processing phase cost $\phi_{proc}(q)$

Figure 12: APP1: class selection and offloading cost



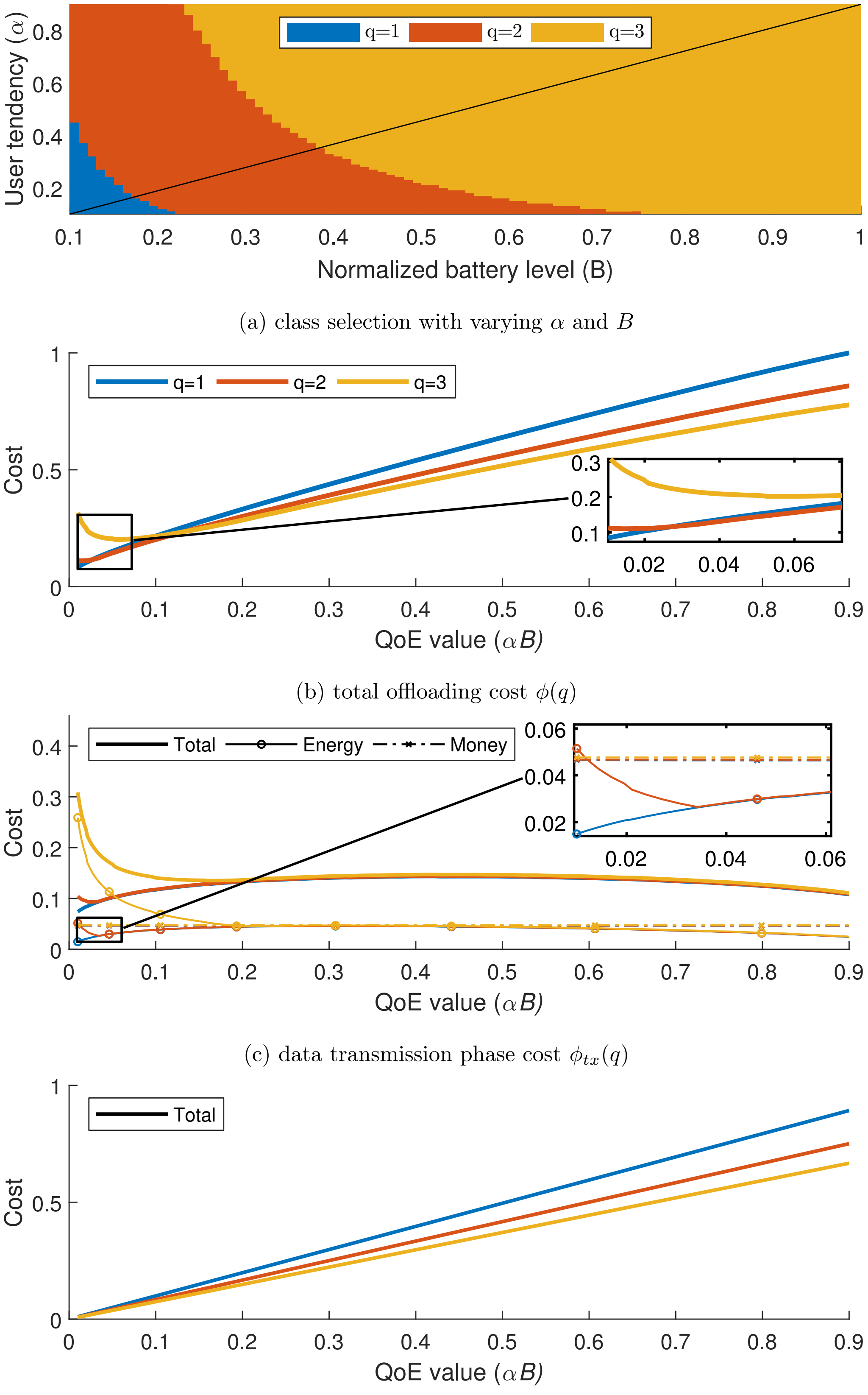
(a) class selection with varying α and B

(b) total offloading cost $\phi(q)$

(c) data transmission phase cost $\phi_{tx}(q)$

(d) processing phase cost $\phi_{proc}(q)$

Figure 13: APP2: class selection and offloading cost



(a) class selection with varying α and B

(b) total offloading cost $\phi(q)$

(c) data transmission phase cost $\phi_{tx}(q)$

(d) processing phase cost $\phi_{proc}(q)$

Figure 14: APP2*: class selection and offloading cost

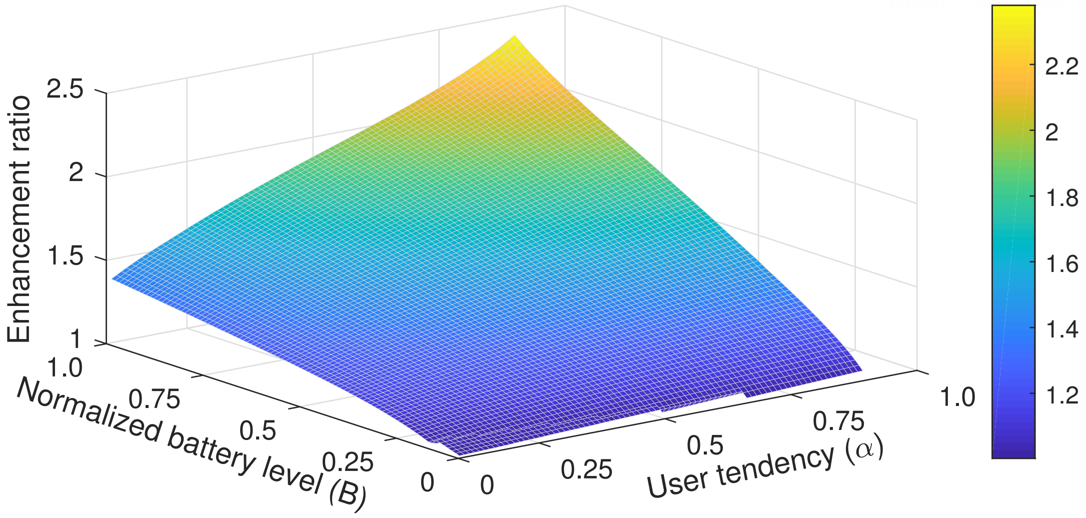


Figure 15: QoE enhancement ratio with varying α and B

Fig. 12b presents the overall offloading cost $\phi(q)$, Fig. 12c presents the total data transmission phase cost $\phi_{tx}(q)$ and its two most significant contributors, energy cost[13] and monetary cost[14], and Fig. 12d presents the total processing phase cost $\phi_{proc}(q)$ and its major contributor, latency cost.[15] It can be seen from Fig. 12a that except when αB is extremely small, the larger αB is, the more expensive class is chosen because latency becomes more important than energy, where $q = 1$ is chosen by 54.62% of the cases, followed by $q = 2$ (42.17%) and $q = 3$ (3.21%). When αB is very small, however, each MECOS service class incurs the almost same amount of $\phi_{proc}(q)$ as presented in Fig. 12d; therefore $\phi_{tx}(q)$ is critical to making a decision. In the data transmission phase, a more expensive class provides more room to save energy (e.g., $q = 3$ consumes the least energy) but incurs more charges, as presented in Fig. 12c. Due to this trade-off, $q = 2$ is accordingly selected, which consumes moderate energy and requires a moderate cost.

Next, Figs. 13 and 14 present the results of APP2 and APP2* in a similar way to Fig. 12. We can first notice that APP2 always selects $q = 3$ as shown in Fig. 13a. On the contrary, APP2* diversifies its choice as shown in Fig. 14a, which is caused by the difference in $\phi(q)$ at small αB as seen from Figs. 13b and 14b. The main reason for this change is that APP2* consumes more energy than APP2 at small αB, as seen from Figs. 13c and 14c, due to its smaller monetary budget and in turn tighter deadline $T(q)$.

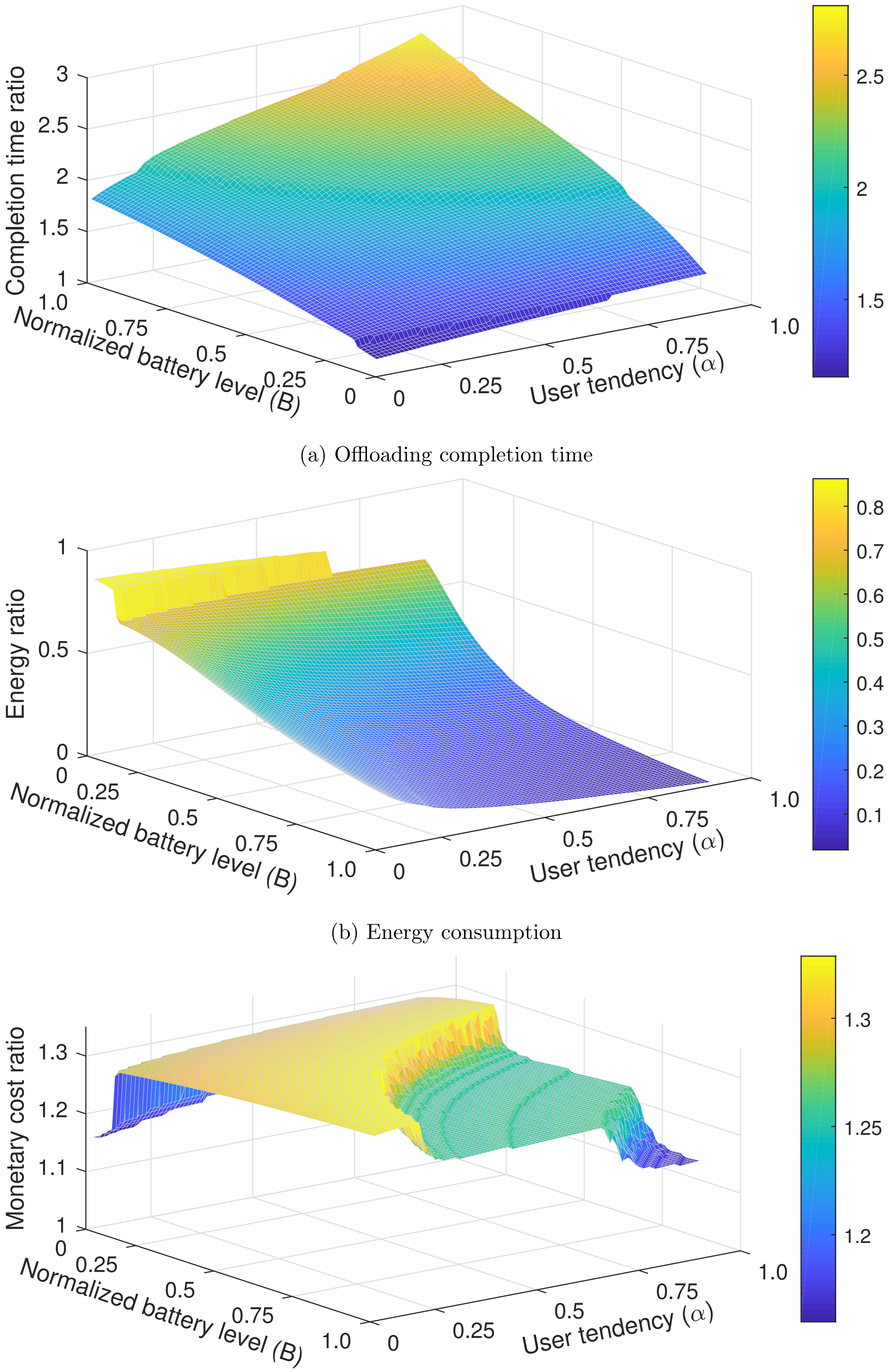
Impact of QoE Factors

To evaluate the impact of QoE factors on the total achieved cost, we ran two versions of the ADPe algorithm: $w_{QoE} = 0$ (i.e., energy-only case) vs. $w_{QoE} > 0$ (i.e., QoE case), for the

[13] More specifically, the energy-related part in the first term of Eq. (51).

[14] More specifically, the monetary-related part in the first term of Eq. (51) plus the second term of Eq. (51).

[15] Note that we later omit the latency cost plot in Figs. 13 and 14 because the gap between $\phi_{proc}(q)$ and the latency cost gets even narrower.



(a) Offloading completion time

(b) Energy consumption

(c) Monetary Cost

Figure 16: Energy, completion time, and monetary cost with varying (α, B)

APP1 scenario. Then, we measured how much times the QoE-based scheme can enhance the cost compared to the energy-only counterpart via the *enhancement ratio*, which is defined as

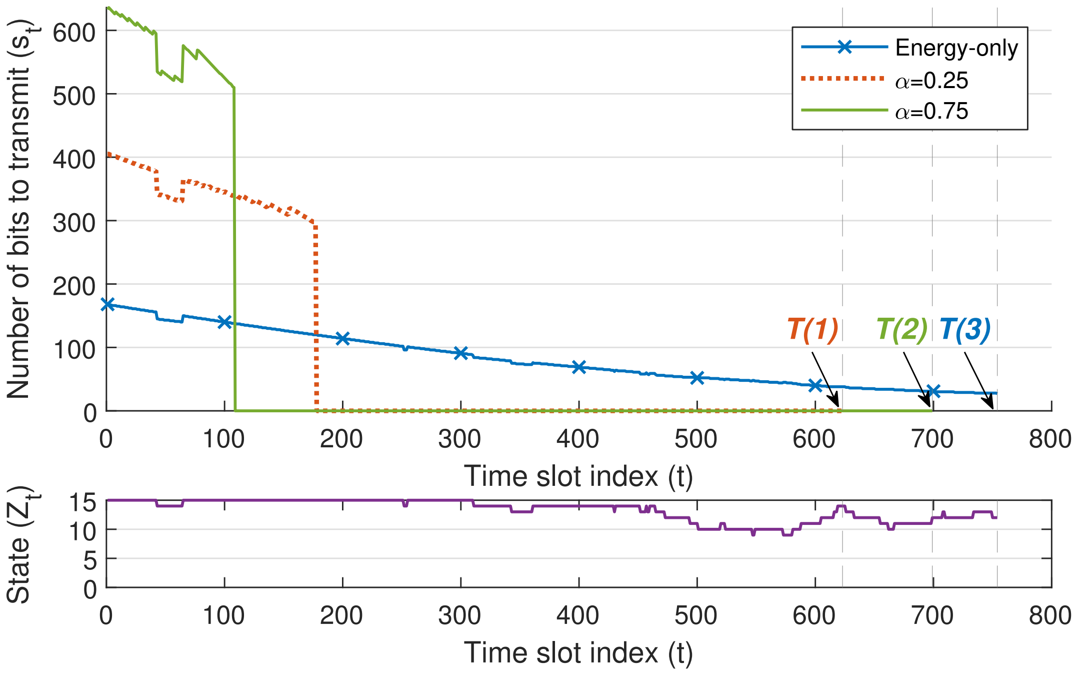
$$\text{enhancement ratio} = \frac{J^*_{ADPe} \text{ when } w_{QoE} = 0}{J^*_{ADPe} \text{ when } w_{QoE} > 0}. \tag{62}$$

Fig. 15 illustrates the measured enhancement ratio with $0.1 \leq \alpha \leq 0.9$ and $0.1 \leq B \leq 1$. As shown, the QoE-based scheme achieves up to 2.38 times smaller (thus better) cost when both α and B are large, i.e., the user is more delay-sensitive and has enough energy, since there is more room in balancing energy with latency. Such enhancement gets less prominent as either α or B decreases eventually converging to the ratio equal to 1, which is expected since energy saving becomes most important in such a situation.

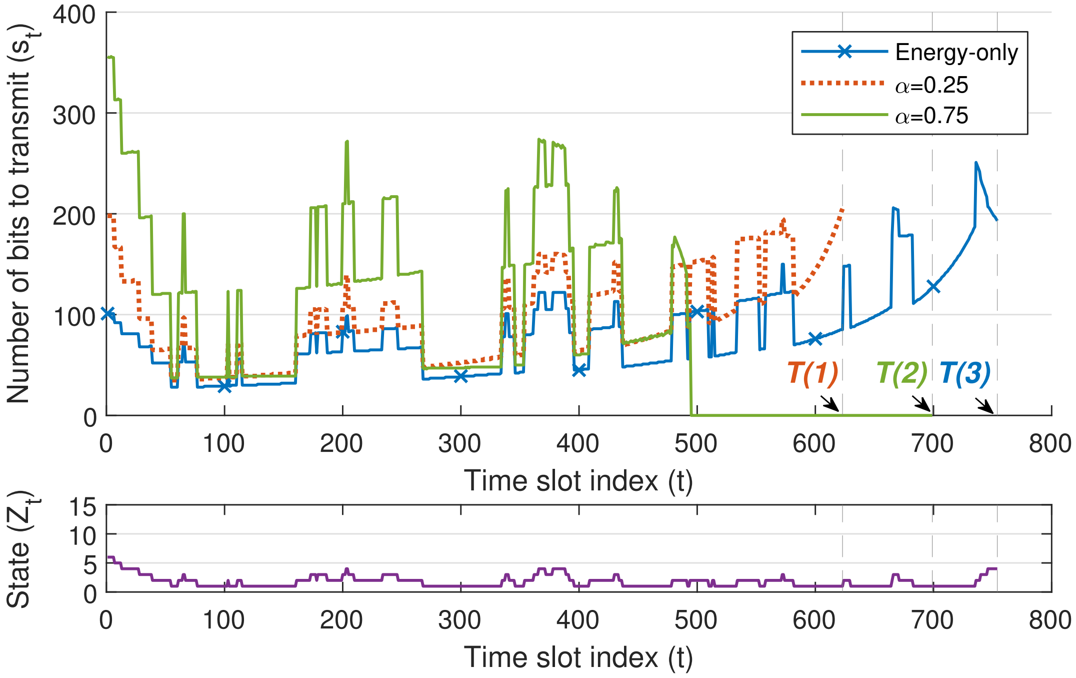
Fig. 16 also presents ADPe's performance with varying α and B. Figs. 16a, 16b, and 16c plot the ratio between $w_{QoE} = 0$ and $w_{QoE} > 0$ in terms of the completion time slot, energy cost, and monetary cost, respectively.[16] As seen in Fig. 16a, as α and B gets larger, the completion time gap between the two cases gets wider resulting in up to 2.81 times faster completion in the QoE case. In addition, Fig. 16b shows that the QoE case always consumes more energy and the gap between the two cases gets larger as α and/or B grows. This is because, for the sake of minimizing the energy-latency combined cost, the QoE scheme is willing to sacrifice more energy as the device has larger remaining battery and/or the user's delay-sensitivity is stronger. Finally, Fig. 16c reveals that the ratio of monetary cost is always larger than 1, implying that the QoE scheme spends less money than the energy-only scheme while achieving QoE-aware scheduling. Note that several dramatic drops in the cost can be observed, which originate from the change in the optimal service class decision.

Fig. 17 compares the QoE-aware scheme ($\alpha = 0.25, 0.75$) with the energy-only scheme when $B = 0.5$, in terms of the optimized transmission scheduling. Because each scheme selects a different service class, such as $q = 1$ for $\alpha = 0.25$, $q = 2$ for $\alpha = 0.75$, and $q = 3$ for the energy-only scheme, their transmission time budgets are also different. Fig. 17a presents a good channel condition, where the two QoE scheme completes the offloaded task as early as at slot 178 out of 623 slots (the $\alpha = 0.25$ case) and at slot 109 out of 699 slots (the $\alpha = 0.75$ case) by exploiting the favorable environment, whereas the energy-only scheme consumes all the slots resulting in the worst delay performance. Fig. 17b, on the other hand, presents a bad channel condition, where the QoE scheme with $\alpha = 0.75$ completes the task earlier than others (at slot 495) to cope with the delay-sensitivity even though it has more room for transmission than small α, while the QoE scheme with $\alpha = 0.25$ consumes all the slots reflecting the preference to energy saving. In fact, such behavior of $\alpha = 0.25$ shows the same tendency with the energy-only scheme, in the sense that they tend to defer transmission until the last moments resulting in rapidly rising s_t towards the end.

[16] That is, J^*_{ADPe} in Eq. (62) is replaced by the metric in consideration.



(a) Under a good channel condition



(b) Under a bad channel condition

Figure 17: QoE-aware scheduling vs. energy-only scheduling

Summary of Findings

We hereby summarize major findings in Section 2.6. Section 2.6 confirmed that the proposed ADPe algorithm achieves near-optimal performance while much reducing computational complexity. Section 2.6 presented that the decision maker may choose the MECOS service class quite differently according to the nature of the given task, user tendency α, the remaining battery level, and the monetary budget. Section 2.6 further showed that QoE-aware scheduling always achieves lower total cost than energy-saving-only scheduling, which gets more prominent as α and B becomes larger. It has also revealed that QoE-aware scheduling well captures the user tendency by completing offloading faster for larger α, and adapts to the time-varying channel conditions.

As a result, we validate that energy, latency, and monetary costs are all crucial factors in computation offloading, where each can affect user's QoE significantly.

2.7 Conclusion and Future Work

In this chapter, we proposed optimal transmission scheduling and optimal service class selection of computation offloading while capturing the tradeoff between energy, latency, and pricing. The transmission scheduling problem has been formulated as DP, and its optimal scheduling and two suboptimal scheduling algorithms have been derived. The service class selection problem has been also introduced for a mobile user to choose the best service to use for offloading. Through extensive simulations, the proposed transmission scheduling has been shown to effectively reduce the latency while enhancing the performance against the existing energy-only scheme. In addition, the proposed service class selection method has been shown to select the optimal class according to user tendency and the remaining battery level while satisfying the monetary constraint.

In the future, we would like to consider more diverse applications, e.g., the ones that allow parallel processing of the offloaded task at the cloud while its data is still being transmitted. In addition, we would like to investigate how to fine-tune the QoE-related weighting factors to reflect actual human behaviors and experience, based on the feedback provided by mobile users like Mean Opinion Score (MOS).

III Lightweight Wi-Fi Frame Detection for Licensed Assisted Access LTE

3.1 Introduction

Licensed Assisted Access (LAA) LTE (LAA-LTE) is a 3GPP standard that combines unlicensed bands with LTE's licensed bands via carrier aggregation [62]. LAA-LTE (henceforth referred to as LAA) adopts energy detection (ED) based Clear Channel Assessment (CCA) to peacefully coexist with legacy devices in the unlicensed spectrum (e.g., Wi-Fi). Using CCA, an LAA device senses a channel busy when other co-channel devices are transmitting, and postpones its transmission if the energy of the received signal is stronger than the ED threshold.

According to 3GPP Release 15 [62], LAA's maximum ED threshold should be −52 dBm given that there is a *long-term* guarantee of the absence of other legacy technologies, or −72 dBm otherwise.[17] In reality, however, Wi-Fi is already pervasive everywhere, and hence LAA devices will mostly have to consider −72 dBm. This will in turn lead to severely degraded spatial reuse due to too conservative transmission by LAA devices, since their ED thresholds can no longer be set beyond −72 dBm *even when* the Wi-Fi devices in the vicinity temporarily stay inactive.

Spatial efficiency of LAA could be much enhanced if LAA is allowed to set the ED threshold dynamically, e.g., −72 dBm when nearby Wi-Fi stations are transmitting but −52 dBm during when only LAA devices are active in the area temporarily. Unfortunately, however, with ED alone, LAA cannot distinguish whether the received signal is originated from LAA or Wi-Fi, since ED does not capture the unique feature of Wi-Fi's transmission. Although one might argue that an LAA device can recognize an LAA frame once it decodes the first OFDM symbol of the received signal, Wi-Fi frames cannot be protected at the moment they appear since the duration of one OFDM symbol (66.7 μs [63]) is much larger than LAA's CCA delay requirement (4 μs [62]), and we cannot still apply a Wi-Fi specific ED threshold by only identifying LAA frames. Therefore, we need a more lightweight Wi-Fi frame detection mechanism so that LAA devices can instantly determine whether the received signal is from Wi-Fi or not.

There have been several approaches to enable LAA to differentiate Wi-Fi frames from LTE frames. [64–68] proposed making LAA devices equipped with Wi-Fi interface, using which Wi-Fi frames can be decoded. Such approaches, however, incur a non-trivial extra cost due to the increased complexity in combining two technologies seamlessly. Another possible approach is to exchange control packets among LAA nodes to indicate their transmission intention so that any unannounced received signals can be considered as non-LAA-compliant, at the expense of huge signaling overhead.

To detect Wi-Fi frames without Wi-Fi interfaces, inter-technology detection algorithms have been also proposed. First, [69] adopts the artificial neural network, with which a Wi-Fi device can

[17] Provided that the channel bandwidth is 20 MHz and the transmit power is 23 dBm.

detect the presence of an LTE-U signal by examining the error pattern of a received Wi-Fi signal. The proposed method, however, works only after a collision occurs. In addition, [70] presents an LTE-U algorithm to detect the number of coexisting Wi-Fi APs based on the distribution of the received signal power. The work, however, requires prior knowledge on the distribution and one second measurement period for detection.

Without implementing Wi-Fi frame detection, [71] shows that the performance of LAA can still be improved by lowering Wi-Fi's energy detection threshold, but at the expense of the degradation of Wi-Fi throughput. On the other hand, [72] proposed an integrated network of LTE-A and WLAN, and considered centralized scheduling of underlay transmissions to maximize their sum throughput while guaranteeing the minimum per-network throughput. Although such an approach might avoid listen-before-talk based channel access by allowing inter-network interference, it requires centralized coordination between the two networks.

This chapter tries to fill the gap by proposing a lightweight but effective method, using which LAA can detect the existence of a Wi-Fi frame and recognize the duration of the frame. The proposed method is built upon a popular Wi-Fi preamble detection method called Schmidl-Cox detection (SCD), which has been carefully modified to overcome the critical difference between LAA and Wi-Fi. Specifically, we have fine-tuned the SCD parameters via extensive experiments using USRP, and then shown that the developed method satisfies the CCA time requirement of the 3GPP standard [62] and the IEEE 802.11 standard [73]. Our contribution is four-fold:

- The proposed Wi-Fi preamble detection is built upon a correlator utilizing LAA's own time domain samples, thus incurring low complexity and requiring no extra Wi-Fi interface. Moreover, the proposed scheme can be performed at LAA devices in a distributed way without necessitating any coordination between them, and also satisfies the required time constraint of 4 μs.

- The proposed energy tracking algorithm can capture the start and the end of a Wi-Fi frame, by tracking the energy pattern of the signal identified by the proposed preamble detection.

- Combining the two mechanisms, an LAA device can differentiate Wi-Fi frames from LAA frames, thus allowing LAA to dynamically vary the ED threshold according to the type of a coexisting signal. Moreover, we can better protect Wi-Fi stations from LAA by applying even smaller ED threshold to them than the current LAA standard specifies. By doing so, LAA can significantly enhance its spatial reuse efficiency.

- We have confirmed the accuracy of the proposed methods via MATLAB simulations and USRP-based experiments, and further evaluated via NS-3 based simulations how much spatial reuse can be enhanced by our methods. In addition, we also revealed that the proposed mechanism can protect Wi-Fi better than the legacy LAA.

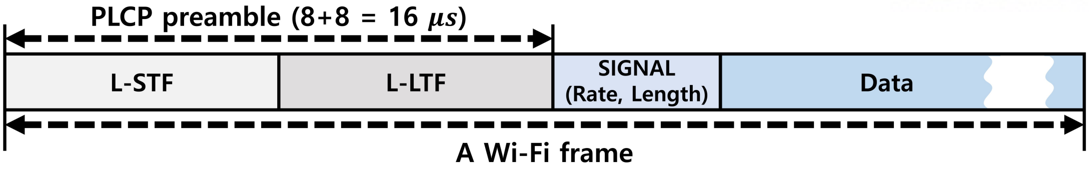


Figure 18: Structure of a Wi-Fi frame

The remainder of the chapter is organized as follows. Section 3.2 introduces preliminaries including the Schmidl-Cox detection, and Section 3.3 elaborates our proposed LAA mechanism. Section 3.4 evaluates the performance of the proposal in a multi-cell LAA-WLAN coexistence scenario, and finally Section 3.5 concludes the chapter.

3.2 Preliminaries

Wi-Fi Preamble Structure

Fig. 18 illustrates the structure of a Wi-Fi frame, where its PLCP preamble consists of L-STF and L-LTF [73]. L-STF further consists of 10 identical short symbols, each comprised of 16 time domain samples, whereas L-LTF consists of two long symbols. In particular, L-STF is designed to help a Wi-Fi station detect the start of a Wi-Fi frame, for which L-STF's 160 time domain samples can be utilized to capture its periodic pattern repeated every 16 samples.

Schmidl-Cox Detection for Wi-Fi Preamble Detection

The Schmidl-Cox detection (SCD) is one of the most common methods for Wi-Fi preamble detection in WLAN [74], which is described as follows. Let $s_{recv}(k)$, $1 \leq k \leq N_{\mathrm{W}}$, be the samples taken from the L-STF part of a received Wi-Fi frame, where N_{W} is referred to as the SCD window size and is assumed an even number. The SCD detector splits the group of N_{W} samples into two disjoint subgroups, each with $N_{\mathrm{W}}/2$ samples, and measures the correlation between them such as

$$C_{\mathrm{W}} = \left| \sum_{k=1}^{N_{\mathrm{W}}/2} \frac{s_{recv}(k) \cdot s^{*}_{recv}(k + \frac{N_{\mathrm{W}}}{2})}{\sum_{k=1}^{N_{\mathrm{W}}/2} |s_{recv}(k)|^2} \right|^2, \tag{63}$$

where $\sum_{k=1}^{N_{\mathrm{W}}/2} |s_{recv}(k)|^2$ is a normalization factor and $s^{*}_{recv}(\cdot)$ is the complex conjugate of $s_{recv}(\cdot)$. For a given threshold C_{th}, $C_{\mathrm{W}} > C_{\mathrm{th}}$ implies that the samples are periodic with the period of $N_{\mathrm{W}}/2$, thus suggesting the existence of a Wi-Fi preamble (more specifically, its L-STF). Considering L-STF's repeating pattern with the period of 16, we have $N_{\mathrm{W}} \in \{32, 64, 96, 128, 160\}$.

3.3 SCD-based Wi-Fi Preamble Detector Design for LAA-LTE

Implementing SCD using off-the-shelf LAA devices confronts a unique challenge due to the difference in the sampling frequency between Wi-Fi and LAA. When utilizing a 20 MHz channel, the sampling frequency of Wi-Fi is set to $f_{\mathrm{W}} = 20$ MHz [73], whereas that of LAA is set to

$f_{\mathrm{L}} = 30.72$ MHz [63].[18] Hence, when an LAA device receives a Wi-Fi frame, the Wi-Fi preamble is upsampled. Each symbol of L-STF in the Wi-Fi preamble is designed to have the same 16 samples when sampled at 20 MHz by Wi-Fi for periodicity. If sampled at 30.72 MHz by LAA, however, each symbol of L-STF has 24 or 25 samples, which are in general different from those of other symbols in L-STF. Thus, the periodicity is not guaranteed in the sampled signals by LAA.

Fortunately, since f_{L} is much larger than f_{W}, an LAA device extracts more samples from the received signal than a Wi-Fi station does. Therefore, the samples obtained by LAA include more details of the signal structure, from which we deduce that the LAA device could preserve the periodicity of a Wi-Fi preamble despite the incompatibility in the PHY design. We therefore revisit the SCD algorithm to redesign its parameters for LAA, and then obtain the best parameter values via USRP-based experiments.

Proposed Framework for Wi-Fi Preamble Detection by LAA

The Schmidl-Cox detection (SCD) is employed as for Wi-Fi preamble detection by LAA. After being upsampled by LAA, the original 160 samples of L-STF become $N_{\mathrm{STF}} = \lceil 160 \times 30.72/20 \rceil = 246$ samples. Immediate conversion of the correlation window size would be $N_{\mathrm{L}} = \lceil N_{\mathrm{W}} \times 30.72/20 \rceil$, which, however, is not always an even number. In pursuit of making the two blocks to be correlated have the same number of samples, we choose to make N_{L} be the closest even number of $N_{\mathrm{W}} \times 30.72/20$ such as[19]

$$N_{\mathrm{L}} = \mathrm{round}\left(\frac{N_{\mathrm{W}} \times 30.72/20}{2} \right) \times 2, \tag{64}$$

which results in $N_{\mathrm{L}} \in \{50, 98, 148, 196, 246\}$ considering $N_{\mathrm{W}} \in \{32, 64, 96, 128, 160\}$. A proper choice of N_{L} will be determined in Section 3.3.

To check the periodicity of the upsampled L-STF, the correlation is computed with the window size N_{L} starting from the first sample to the last (i.e., N_{STF}-th) sample of the L-STF. Specifically, the correlation metric at the k-th window, $k \in [0, N_{\mathrm{STF}} - N_{\mathrm{L}}]$, is defined by

$$C_{\mathrm{L}}^{(k)} := \left| \sum_{k_1=1+k}^{N_{\mathrm{L}}/2+k} \frac{s_{\mathrm{L}}(k_1) \cdot s_{\mathrm{L}}^*(k_1 + N_{\mathrm{L}}/2)}{\sum_{k_1=1+k}^{N_{\mathrm{L}}/2+k} |s_{\mathrm{L}}(k_1)|^2} \right|^2, \tag{65}$$

where $s_{\mathrm{L}}(u)$ denotes the u-th sample of the L-STF sampled at 30.72 MHz. Then, an LAA device acknowledges the presence of a Wi-Fi signal if

$$C_{\mathrm{L}} = \max_k C_{\mathrm{L}}^{(k)} > C_{\mathrm{th}}, \tag{66}$$

where C_{th} is the threshold value to be determined in Section 3.3.

Using the aforementioned procedure, the start of a Wi-Fi frame can be detected. Nevertheless, SCD alone cannot determine until when the Wi-Fi's transmission must be protected. Once

[18] The sampling frequency implies the sampling rate of a received signal at the baseband.

[19] In fact, we have confirmed this choice is the best via computer simulations.

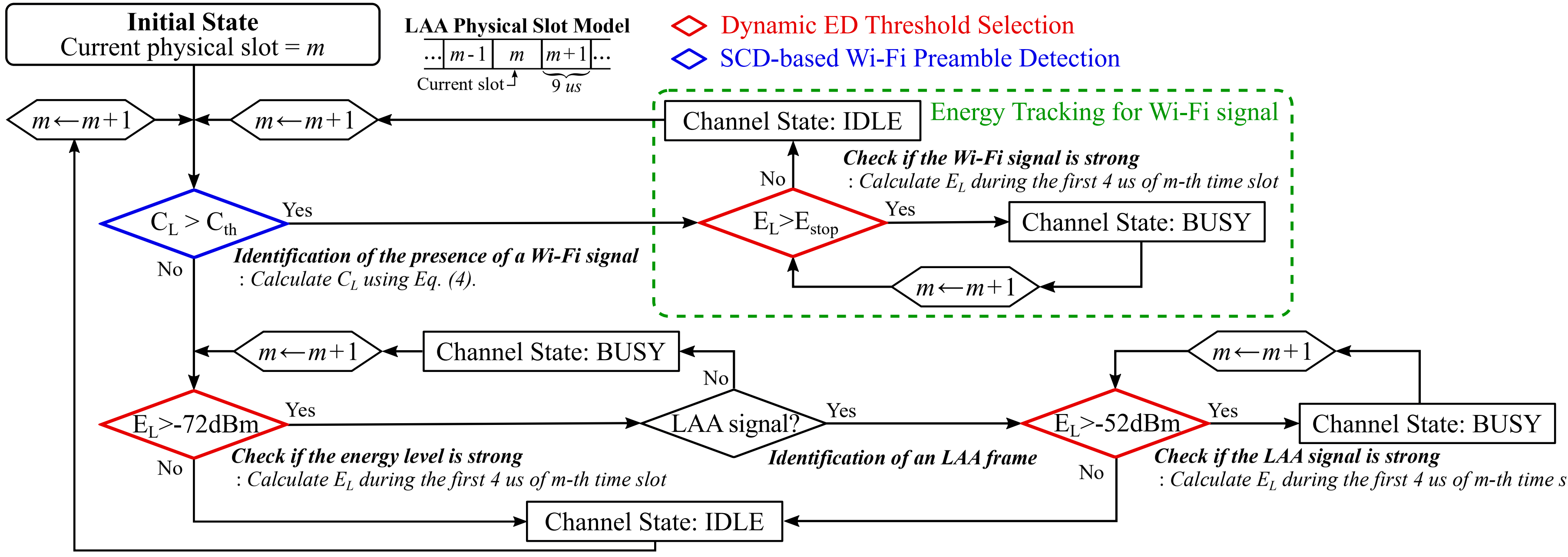

Initial State
Current physical slot = m
LAA Physical Slot Model
... m-1 m m+1 ...
Current slot
9 us
Dynamic ED Threshold Selection
SCD-based Wi-Fi Preamble Detection
Energy Tracking for Wi-Fi signal
Channel State: IDLE
Check if the Wi-Fi signal is strong
: Calculate E_L during the first 4 us of m-th time slot
No
Yes
$E_L > E_{stop}$
Channel State: BUSY
$m \leftarrow m+1$
$C_L > C_{th}$
Yes
No
Identification of the presence of a Wi-Fi signal
: Calculate C_L using Eq. (4).
Channel State: BUSY
No
E_L>-72dBm
Yes
No
Check if the energy level is strong
: Calculate E_L during the first 4 us of m-th time slot
LAA signal?
Yes
Identification of an LAA frame
E_L>-52dBm
Yes
No
Channel State: BUSY
Check if the LAA signal is strong
: Calculate E_L during the first 4 us of m-th time slot
Channel State: IDLE

Figure 19: Proposed lightweight Wi-Fi frame detection mechanism for LAA devices

Table 4: $E[C_{\mathrm{L}}^{(k)}]$ and $Var(C_{\mathrm{L}}^{(k)})$ under the perfect channel

N_{L}	50	98	148	196	246
$E[C_{\mathrm{L}}^{(k)}]$	0.80	0.977	0.928	0.911	0.980
$Var(C_{\mathrm{L}}^{(k)})$	10^{-5}	7×10^{-7}	5×10^{-7}	4×10^{-7}	0

a Wi-Fi signal is acknowledged, we propose employing the *Energy Tracking* algorithm, where the LAA device measures the energy E_{L} of the received signal at every slot, and if $E_{\mathrm{L}} \leq E_{\mathrm{stop}}$ occurs at a slot, the device treats the slot as the end of the Wi-Fi frame. A proper value of E_{stop} will be determined in Section 3.3.

Note that 3GPP requires that the energy detection be conducted for the first 4 μs of each 9 μs physical slot [62], which is set similar to Wi-Fi. Hence, we propose that an LAA transmitter performs SCD with the window size of N_{L} for the first 4 μs at every slot, and if $C_{\mathrm{L}} > C_{\mathrm{th}}$, the device concludes that the received signal has a Wi-Fi preamble. In the 4 μs period, the SCD window is sliding such that the first SCD is performed by using the first sample of the period combined with the previous $N_{\mathrm{L}} - 1$ samples, whereas the last SCD is performed by using the last N_{L} samples within the period. Accordingly, the energy detection by the Energy Tracking algorithm is also performed for the first 4 μs at every slot.

The proposed Wi-Fi frame detection mechanism is illustrated in Fig. 19. Assuming slot m is the current physical slot, an LAA device first tries to detect whether a Wi-Fi preamble exists via the proposed *SCD-based Wi-Fi preamble detection* method (shown as a blue-colored box in the figure). If detected, the proposed *Energy Tracking* method is performed (shown in the dotted green box). Otherwise (i.e., no Wi-Fi preamble), the algorithm tries to determine whether there exists an LAA signal (via LAA decoding) or a signal other than Wi-Fi and LAA, and the channel is determined idle if no signal exists. The red boxes constitute the proposed *Dynamic ED threshold selection* method, where the thresholds of E_{stop}, -72 dBm, and -52 dBm are applied selectively depending on the type of the detected signal. Note that $m \leftarrow m + 1$ implies the algorithm moves to the next slot. In addition, LAA's DCF-like channel access mechanism[20] operates in accordance with the state update by Fig. 19, e.g., channel access is deferred if the channel state is decided busy, and the backoff procedure is resumed if the channel state is decided idle.

Proposed LAA-WiFi coexistence scenario: An LAA device improves its spatial efficiency by relaxing the requirement of the 3GPP standard such that it applies (1) a higher ED threshold of -52 dBm for received LAA signals, and (2) a lower ED threshold of -82 dBm for received Wi-Fi signals. For any other signal, the standard CCA threshold of -72 dBm is applied. On the other hand, a Wi-Fi device works in accordance with the IEEE 802.11 specification. For example, if a Wi-Fi AP receives and detects a Wi-Fi signal (via its legacy preamble detection

[20] LAA performs channel access in a similar way to Wi-Fi's Distributed Coordination Function (DCF).

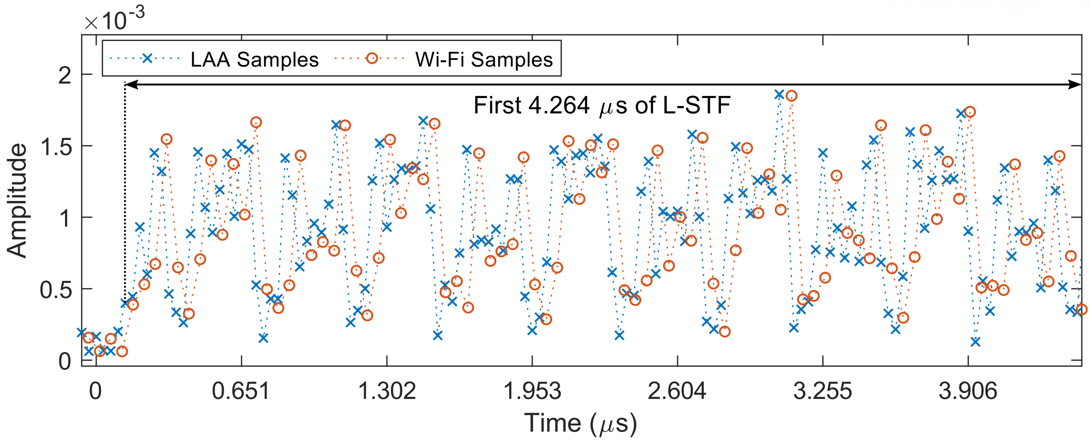


Figure 20: LAA and Wi-Fi samples for the first 4.264 μs of L-STF

mechanism), the preamble detection CCA threshold is applied (e.g., -82 dBm for 20 MHz channel). Otherwise, the energy detection CCA threshold (e.g., -62 dBm for 20 MHz channel) is applied. Therefore, a Wi-Fi device applies -62 dBm to a received signal originated from an LAA device.

Feasibility check: To ascertain whether the upsampled signal preserves the periodicity of L-STF, we performed a MATLAB simulation with a Wi-Fi transmitter and an LAA device. In doing so, we assumed the perfect channel between the two devices,[21] to focus on the impact of the sampling frequency mismatch. The Wi-Fi transmitter constructs a Wi-Fi frame and transmits it over the channel; then, the LAA device upsamples the received preamble at 30.72 MHz, and calculates $C_{\mathrm{L}}^{(k)}$ for various N_{L} and k. Table 4 presents the mean $E[C_{\mathrm{L}}^{(k)}]$ and the variance $Var(C_{\mathrm{L}}^{(k)})$ with respect to k, for each N_{L}, where it reveals that $E[C_{\mathrm{L}}^{(k)}]$ is very close to 1 (except when $N_{\mathrm{L}} = 50$) while $Var(C_{\mathrm{L}}^{(k)})$ is negligible.

In fact, the results suggest using $N_{\mathrm{L}} = 246$ and large C_{th} for achieving the best accuracy in the periodicity check. Nevertheless, wireless environments encounter various uncertainties and thus $(N_{\mathrm{L}}, C_{\mathrm{th}})$ should be determined via real experiments. Moreover, the chosen N_{L} should fit into the 4 μs duration. In the sequel, it will turn out a proper configuration is quite different from what the above simulation suggests.

Nevertheless, we want to provide experimental evidence on the periodicity of L-STF preserved by LAA's upsampling. We generated and transmitted a Wi-Fi signal with a USRP, and captured the signal by another USRP acting as either a Wi-Fi device or an LAA device.[22] Fig. 20 presents the signal amplitude of thus-obtained samples for the first 4.264 μs of L-STF, clearly showing that LAA samples closely follow the trend of Wi-Fi samples. That is, the periodicity of L-STF is indeed preserved in LAA samples even in a real noisy channel environment.

[21] A perfect channel is the channel without channel distortion or noise.

[22] The signal is sampled by the USRP at 25M samples per second and then is properly up/down-sampled to match with the sampling frequencies of 20 MHz and 30.72 MHz respectively.

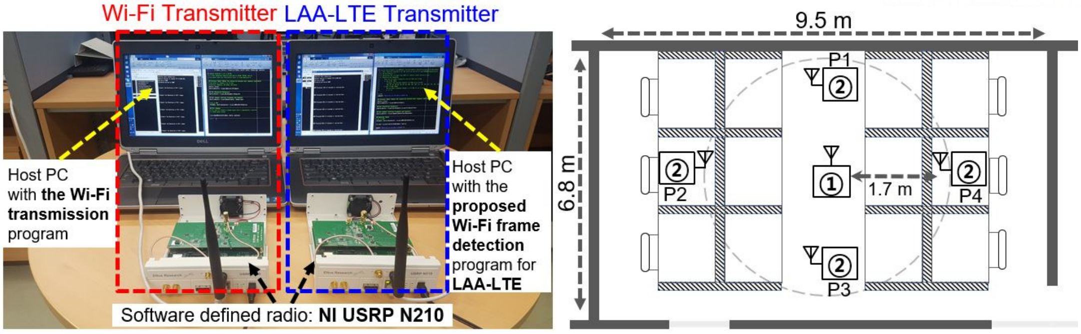

Wi-Fi Transmitter
LAA-LTE Transmitter
Host PC with the Wi-Fi transmission program
Host PC with the proposed Wi-Fi frame detection program for LAA-LTE
Software defined radio: NI USRP N210
9.5 m
6.8 m
P1
P2
P3
P4
①
②
1.7 m

Figure 21: (left) Implemented Wi-Fi and LAA transmitters based on USRP N210 and (right) experiments scenario, where ① is the Wi-Fi transmitter transmitting Wi-Fi signals and ② is the LAA transmitter or another Wi-Fi transmitter attempting to detect the Wi-Fi frame while they are performing CCA-based backoff.

Configuration of SCD Parameters: N_L, C_th, E_stop

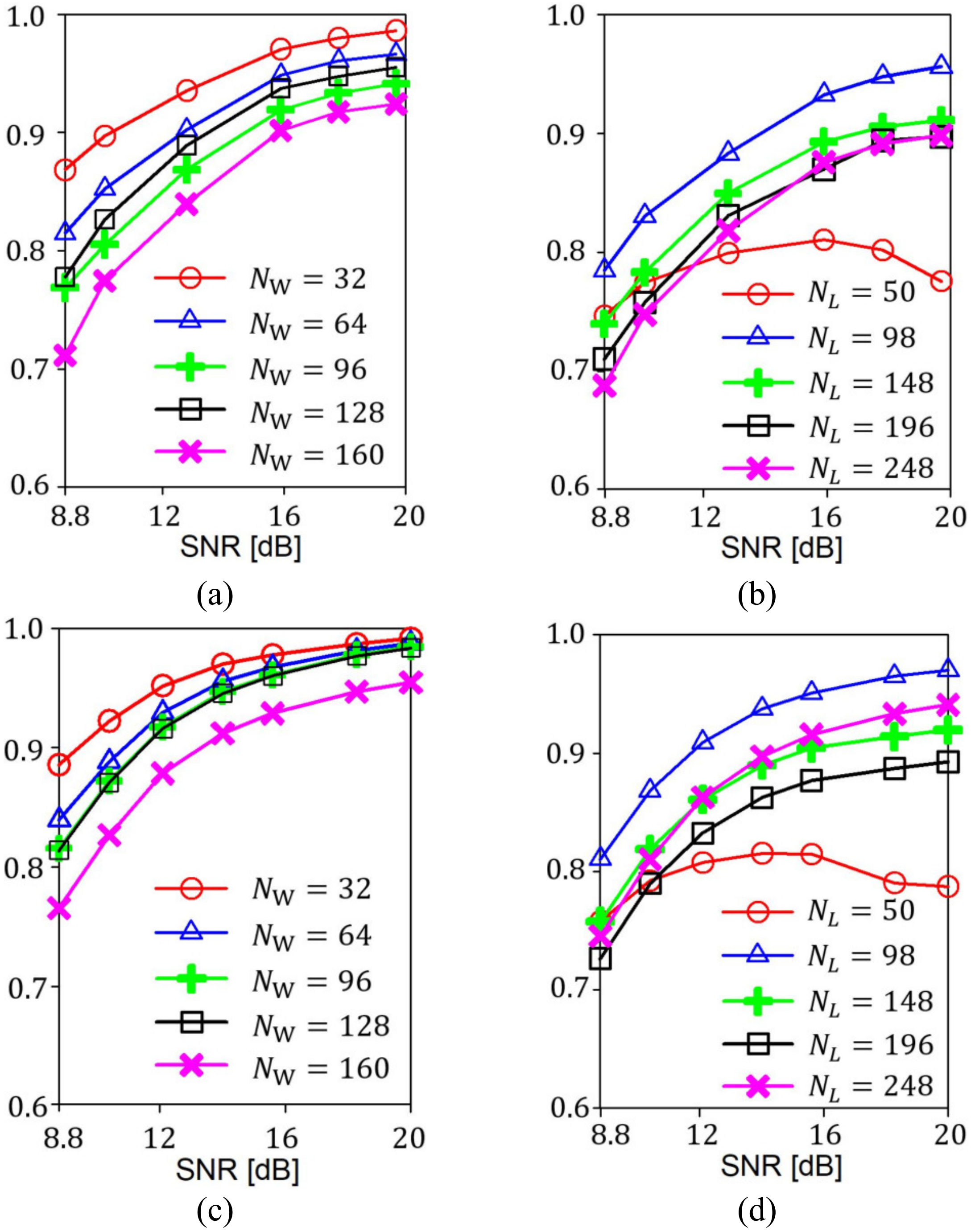
For successful adoption of SCD to LAA, proper configuration of SCD parameters is crucial. In what follows, we present how we performed a series of experiments in a real environment with an LAA transmitter implementing SCD, and suggest the best N_L, C_th, E_stop to use for the LAA transmitter to detect Wi-Fi frames accurately.

For our experiments, we utilized two USRP N210s, implementing a usual Wi-Fi transmitter and an LAA transmitter with the proposed Wi-Fi frame detection, as shown in Fig. 21. For comparison, another N210 was used to implement a Wi-Fi transmitter with conventional Wi-Fi frame detection and energy detection. Each USRP was equipped with an XCVR2450 daughterboard [75] and used the same 20 MHz channel in 5 GHz unlicensed bands. In addition, a USRP is connected to its host PC running MATLAB which sends a generated Wi-Fi preamble to the USRP and performs SCD using the received samples by the USRP [76].

Best N_L To determine the best N_L, we located two devices ① and ② as shown in Fig. 21, where ① is a Wi-Fi transmitter, and ② is *either* another Wi-Fi transmitter running conventional SCD *or* an LAA transmitter running its customized SCD. Then, we considered two scenarios regarding the channel between ① and ②: S_1 for a non-line-of-sight (NLOS) channel and S_2 for a line-of-sight (LOS) channel. That is, S_1 corresponds to the case when ② is placed at $P2$ or $P4$, whereas S_2 corresponds to the case when ② is placed at $P1$ or $P3$. The same scenario has been repeated around 500 times with varying SNR, where the minimum SNR was set to 8.8 dB which corresponds to the minimum required receiver sensitivity of -82 dBm in WLAN [73,74].[23]

Fig. 22 shows C_W and C_L measured under S_1 and S_2. In Figs. 22(a) and 22(c), Wi-Fi's C_W generally tends to enhance as N_W decreases, and thus $N_\mathrm{W} = 32$ achieves the best performance. LAA's C_L in Figs. 22(b) and 22(d), however, presents different behaviors. First, the correlation of

[23] Please refer to [74] for -82 dBm being mapped to 9 dB. Note that we chose 8.8 dB rather than 9 dB due to the difficulty in perfectly setting the desired SNR under channel fluctuation.



(a) (b) (c) (d)

Figure 22: SCD performance of Wi-Fi and LAA under S_1 (NLOS) and S_2 (LOS). (a) C_W in S_1. (b) C_L in S_1. (c) C_W in S_2. (d) C_L in S_2.

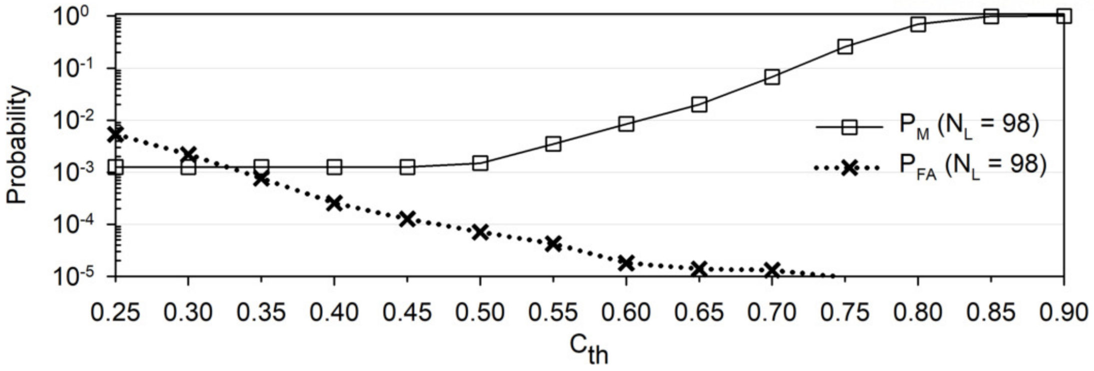


Figure 23: P_{M} and P_{FA} of the proposed SCD mechanism, with $N_{\mathrm{L}} = 98$ and various C_{th}

the sampled preamble is relatively low in the LAA case due to the sampling frequency mismatch. Unlike Wi-Fi, which shows the strongest correlation with $N_{\mathrm{W}} = 32$, the best N_{L} is 98 (not 50) for both S_1 and S_2, showing almost the same performance as Wi-Fi in the high SNR regime. It is seen from the figure that the LOS case, S_2, shows a higher correlation than the NLOS case thanks to better channel condition and less delay spread.

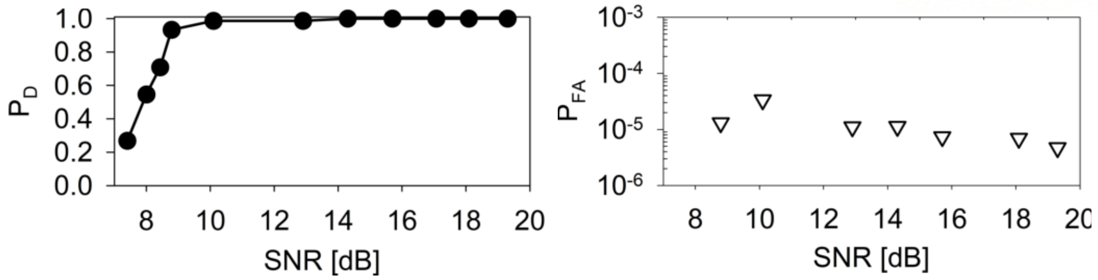
Considering the above observations, we suggest using $N_{\mathrm{L}} = 98$. Note that $N_{\mathrm{L}} = 98$ corresponds to $98/246 \times 8 = 3.18\ \mu s$, which is within the CCA delay requirement of 4 μs.

Best C_{th} Our next objective is to determine C_{th} via experiments that minimizes the false-alarm probability P_{FA}, while suppressing the miss-detection probability P_{M} less than 0.1 at the minimum required receiver sensitivity of −82dBm, as required by the Wi-Fi standard [73]. Here, P_{M} is measured as the ratio of the number of undetected preambles to the number of transmitted preambles, and P_{FA} is measured as the ratio of the number of falsely-detected preambles to the number of performed SCDs during when no preambles exist. Note that by considering −82 dBm as the ED threshold of LAA, instead of −72 dBm, our proposed scheme can better protect Wi-Fi transmissions.

A USRP-based experiment has been conducted under S_1 (i.e., NLOS) with the received SNR of 8.8 dB for 0.2 dB margin, corresponding to −82.2 dBm. Note that S_1 is chosen as it is a more challenging environment than S_2. In addition, our experimental setup is similar to the way Wi-Fi sets C_{th} [74]. Fig. 23 presents P_{M} and P_{FA} of the proposed method with $N_{\mathrm{L}} = 98$ and various C_{th}, which shows that $P_{\mathrm{M}} \leq 0.1$ is achieved for $C_{\mathrm{th}} \leq 0.7$, within which P_{FA} is minimized at $C_{\mathrm{th}} = 0.7$. Combined with the result in Section 3.3, the best SCD configuration becomes $N_{\mathrm{L}} = 98$ and $C_{\mathrm{th}} = 0.7$.

Using $N_{\mathrm{L}} = 98$ and $C_{\mathrm{th}} = 0.7$, we performed another USRP-based experiment while varying the received SNR from 7.5 to 19.3 dB, which is intended to evaluate the efficacy of the proposed SCD configuration. Fig. 24 shows P_{D} and P_{FA}, where $P_{\mathrm{D}} = 1 - P_{\mathrm{M}}$.[24] As shown, P_{D} becomes larger than 0.9 for SNR greater than 9 dB, while P_{FA} is upper-bounded by 33×10^{-6} for all

[24] Please note that Fig. 24(b) omits P_{FA} at a few SNR values where the measured P_{FA} was zero.



(a) (b)

Figure 24: P_{D} and P_{FA} of the proposed SCD mechanism, with varying SNR. (a) P_{D}. (b) P_{FA}.

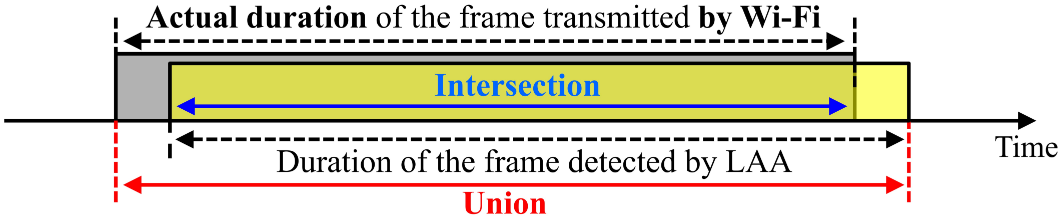

Actual duration of the frame transmitted by Wi-Fi
Intersection
Time
Duration of the frame detected by LAA
Union

Figure 25: Concept of IoU and IoA

SNR. Therefore, it is noticed that an LAA device can reliably detect Wi-Fi preambles by using the proposed scheme while much suppressing the false alarm.

Best E_{stop} In what follows, we discuss how to determine E_{stop} considering its impact on the accuracy of the duration detection. Here, we determine SNR$_{\mathrm{stop}}$ that corresponds to E_{stop}, through an experiment using a Wi-Fi transmitter and an LAA transmitter performing the proposed algorithm with $(N_{\mathrm{L}}, C_{\mathrm{th}}) = (98, 0.7)$. In addition, we varied the received SNR as $\{7.5, 8, 8.4, 8.8, 10, 13, 14, 15.7, 17, 19.3\}$, and the same test was repeated around 500 times for each SNR value.

We measured the performance of the proposed algorithm via the following two metrics:

- *Intersection over Union (IoU)*: IoU is obtained by dividing “Intersection” by “Union” in Fig. 25, which shows how accurately the algorithm can identify the start and the end of a frame. IoU equal to 1 implies the perfect capture of the frame in terms of timing and duration.

- *Intersection over Actual duration (IoA)*: IoA is obtained by dividing “Intersection” by “Actual duration" in Fig. 25, which indicates how well a Wi-Fi frame can be protected against LAA. IoA equal to 1 means that LAA will not interfere with Wi-Fi’s transmission (but does not mean that LAA perfectly recognizes timing and duration).

We conducted the experiment with varying SNR$_{\mathrm{stop}} \in \{7,\ 8,\ 8.8\}$ dB due to the following reason. For the received SNR larger than 9 dB (which corresponds to -82 dBm, the minimum required receiver sensitivity), we need to reliably identify a received signal. Therefore, SNR$_{\mathrm{stop}}$

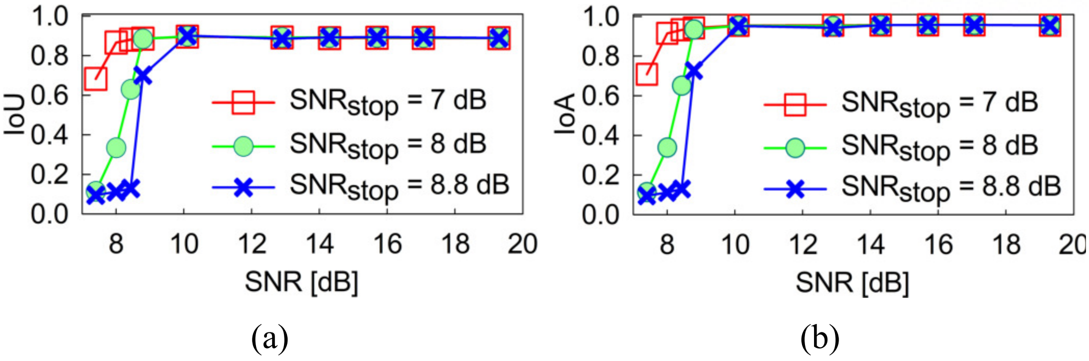


Figure 26: IoU, and IoA of frame detection scheme. (a) IoU. (b) IoA.

Table 5: Evaluation with the signals from vendor's Wi-Fi devices

Channel number	P_D (%)	P_{FA} (%)	IOU	IOA
36	93.8	4.05×10^{-6}	0.964	0.968
44	94.2	1.11×10^{-6}	0.980	0.982
153	98.7	0.03×10^{-6}	0.974	0.975

has to be set smaller than 9 dB, but as large as possible so that the end of a frame can be accurately recognized. In this regard, we have chosen the aforementioned values of SNR_{stop} as reference cases, among which the most desirable choice will be discussed. Note that, however, detailed adjustment of SNR_{stop} may require more fine grained candidate values below 9 dB.

Fig. 26 presents IoU and IoA with varying SNR. When $\text{SNR}_{\text{stop}} = 8.8$ dB, IoU and IoA at SNR = 9 dB become around 0.7, indicating that LAA neither accurately detects the Wi-Fi frame's duration nor fully protects the frame. When $\text{SNR}_{\text{stop}} = 7$ dB, although IoU and IoA at SNR = 9 dB become large enough (0.9 and 0.95 respectively), the two metrics stay at high values even for SNR smaller than 9 dB indicating that an LAA device would unnecessarily protect Wi-Fi frames weaker than −82 dBm. On the other hand, $\text{SNR}_{\text{stop}} = 8$ dB seems desirable in the sense that IoU and IoA are around 0.9 and 0.95 at SNR = 9 dB, and then sharply decrease for SNR smaller than 9 dB.

As a result, we suggest using $\text{SNR}_{\text{stop}} = 8$ dB since it can fully protect a Wi-Fi frame stronger than −82 dBm while tends to ignore weaker signals.

Real-world Performance of the Proposed Mechanism

To further show the practicality of our proposal, we performed an evaluation using the real captured signals from various Wi-Fi vendors existing on our campus. The evaluation configured two USRPs as an LAA transmitter and a Wi-Fi transmitter, each using SCD with Energy Tracking, where both of them detect Wi-Fi frames for 5 seconds simultaneously. The experiment was performed in WLAN channels 36, 44, and 153, in each of which the test was repeated 120 times with an inter-test pausing period of 60 seconds. The frames detected by the Wi-Fi transmitter were used as the ground truth for calculating the four performance metrics, P_D, P_{FA}, IoU, and IoA. Table 5 presents that all three channels achieve $P_D > 90$ %, negligible P_{FA}, IoU ≈ 1, and IoA ≈ 1, and thus confirms that our proposal performs really well in actual environments.

Discussion on Practical Issues

We have followed Wi-Fi's frame format in generating the Wi-Fi signal at the transmitter side, and varied the SNR of the received Wi-Fi signal at the receiver side. Then, we studied the detection performance via SCD based on each given SNR level. Since the aforementioned experimental setup and the SCD method do not depend on any chipset-specific features, we believe the derived SCD configuration is chipset-agnostic and thus generic. We admit, however, the suggested configuration relies on the indoor LOS/NLOS channel environment we have chosen, and thus it should not be applied to outdoor scenarios without further verification.

3.4 Spatial Reuse Enhancement by the Proposal

In this section, we evaluate how much the proposed scheme helps LAA achieve better performance and what impact it has on coexisting Wi-Fi devices. Assuming an LAA network coexists with a Wi-Fi network, we implemented our proposed mechanism by modifying the NS-3 LAA and Wi-Fi simulators [77] as follows.

- If a Wi-Fi preamble is detected (by the proposed SCD-based scheme), the ED threshold of -82 dBm is applied. In other words, a Wi-Fi signal is protected by LAA devices in the same way it is by other Wi-Fi devices. In this way, our scheme provides stronger protection to Wi-Fi compared to the standard LAA with the constant ED threshold of -72 dBm.
- If an LAA transmitter detects an LAA signal, the ED threshold of -52 dBm is applied. By doing so, LAA transmitters can access the channel more aggressively than the standard LAA during when Wi-Fi devices are less active.

In addition, we implement the proposed schemes such that an LAA device detects a Wi-Fi preamble (during CCA) with the probability presented in Table 5.

Based on 3GPP TR 36.889 [78], which defines methodologies for LAA coexistence evaluations, we configure the simulation environment as follows.

- **Coexistence Scenario**: We consider a multi-BS(Base Station) LAA network managed by Operator L, co-existing with a multi-AP(Access Point) Wi-Fi network managed by Operator W. We assume both networks use channel 36 with the center frequency of 5180 MHz and the bandwidth of 20 MHz. In addition, we assume both networks use the transmission power of 23 dBm.

- **Channel Model**: We consider the Indoor Hotspot (InH) model in [79] which defines the pathloss and shadow fading parameters for both line-of-sight (LOS) and nonline-of-sight (NLOS) channels as follows. When a receiver is d meters away from a transmitter, the pathloss is determined as

$$PL \text{ (dB)} = \begin{cases} 16.9 \log_{10}(d) + 32.8 + 20 \log_{10}(f_c), & \text{for LOS,} \\ 43.3 \log_{10}(d) + 11.5 + 20 \log_{10}(f_c), & \text{for NLOS,} \end{cases}$$

  where f_c is the center frequency in GHz, and the applicability ranges are $3 < d < 100$ for LOS and $10 < d < 150$ for NLOS. The shadow fading is given by the log-normal distribution with zero mean and the standard deviation of 3 dB for LOS and 4 dB for NLOS. In addition, the channel between a given transmitter-receiver pair is determined as LOS with probability P_{LOS} such as

$$P_{LOS} = \begin{cases} 1, & \text{when } d \leq 18, \\ \exp\left(-(d-18)/27\right), & \text{when } 18 < d < 37, \\ 0.5, & \text{when } d \geq 37, \end{cases}$$

  and the channel is determined as NLOS with probability $(1 - P_{LOS})$.

- **Network Topology**: We assume each of the two networks L and W consists of 19 hexagonal cells as shown in Fig. 27. Operator L deploys *either* an LAA network with the legacy CCA (henceforth referred to as 'Standard LAA') *or* an LAA network with the proposed schemes (henceforth referred to as 'Proposed LAA'), and operator W deploys a Wi-Fi network (henceforth referred to as 'Wi-Fi'). The distance between two adjacent cells of the same network is 40 meters, at which their inter-cell received power lies between -72 dBm and -52 dBm in case of an NLOS channel, thus enabling concurrent channel access by adjacent LAA BSs. In addition, the two operator networks are apart by 3 meters along the x-axis, i.e., one network is a shifted version of another.[25] We deploy 190 User Equipments (UEs) for network L and 190 Wi-Fi stations (STAs) for network W, which are uniformly distributed within the whole coverage area of each network so that 10 user devices are associated with each cell on average.

- **Traffic Model**: Based on 3GPP TR 36.814 [79],[26] the traffic model we consider follows the Poisson distribution where 0.5 MB files arrive at each cell with an arrival rate $\lambda \in \{0.25, 0.5, 0.75, 1.0, 1.25, 1.5\}$. The first four arrival rates are obtained from [79], and the last two (i.e., $\lambda = 1.25, 1.5$) are added in our simulation to test heavier traffic environments. Note that we consider downlink traffic only in this evaluation.[27] In the sequel, we denote

[25] Note that 3GPP TR 36.889 [78] also suggests a simulation scenario that deploys two coexisting networks with the same topology but shifted to each other.

[26] Note that [78] includes [79].

[27] Consideration to uplink/downlink combined traffic is our future work. Our model, however, provides good intuition since downlink traffic would become 8 times larger than uplink traffic by 2020 [80].

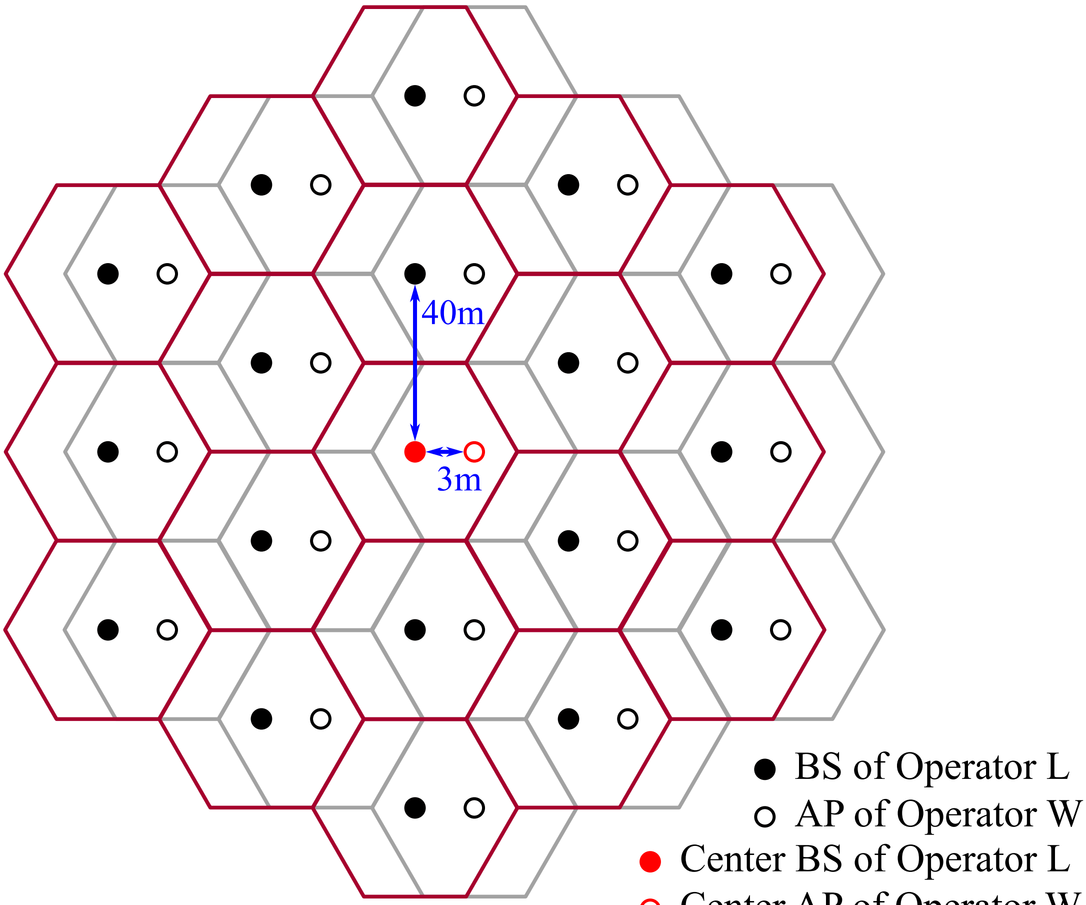


Figure 27: Hexagonal deployment of the two coexisting networks

λ of operators L and W by λ_L and λ_W, respectively.

Under the aforementioned simulation environment, we generate 20 random realizations of UE/STA deployment and run both networks with their traffic generated according to the simulation model. Then, we evaluate the performance of Operator L to show how much Proposed LAA improves the performance compared to Standard LAA, and the performance of Operator W to show how Wi-Fi is affected by the coexisting (Standard or Proposed) LAA network. Among the 19 cells of each operator, we focused on the performance of the two center cells (i.e., the center BS and the center AP) because they experience most unbiased interferences from surrounding cells, i.e., interferences come from virtually every direction.

The considered performance metrics are throughput and delay, measured per packet basis as described in [81]. Specifically, throughput is calculated by averaging per-packet throughput, where per-packet throughput is the amount of correctly received bits (which is either the packet size or zero) divided by the time consumed for sending a given packet. Next, delay is calculated by averaging per-packet delay, where per-packet delay includes channel access delay and transmission delay (including retransmissions).

Performance Improvement of LAA

In this section, we compare the performance of Proposed LAA with Standard LAA in terms of throughput and delay. Note that in Figs. 28 and 29, the results of Standard LAA and Proposed LAA are denoted by Standard and Proposed, respectively. Please remind that Operator L deploys only standard LAA schemes in 'Standard LAA', and only proposed LAA schemes in 'Proposed LAA'.

Fig. 28 shows the throughput performance with varying λ_L and λ_W. As shown, both throughputs are inversely proportional to λ_L and λ_W, and thus the maximum throughput is achieved when $\lambda_L = \lambda_W = 0.25$ (76.92 Mbps for Standard, 83.00 Mbps for Proposed) and the minimum throughput is achieved when $\lambda_L = \lambda_W = 1.5$ (9.38 Mbps for Standard, 10.74 Mbps for Proposed). Table 6 shows that the throughput improvement of Proposed LAA against Standard LAA for varying (λ_L, λ_W), which is calculated as

$$\text{Throughput improvement} = \frac{T_{L,P} - T_{L,S}}{T_{L,S}} \times 100, \tag{67}$$

where $T_{L,S}$ is the throughput of Standard LAA, and $T_{L,P}$ is the throughput of Proposed LAA. When $\lambda_W \leq 0.75$, Proposed LAA always achieves higher throughput than Standard LAA regardless of λ_L, where the gain tends to increase with λ_L (with very few exceptions). When $\lambda_L \geq 1.25$, Proposed LAA always achieves better throughput than Standard LAA regardless of λ_W. In general, performance improvement tends to grow with increasing λ_L and decreasing λ_W, achieving as much as 23.68%. Moreover, Proposed LAA improves the LAA throughput by 5.89% on average, and by 9.29% on average if we only consider the blue entries (which are the cases with improvement).

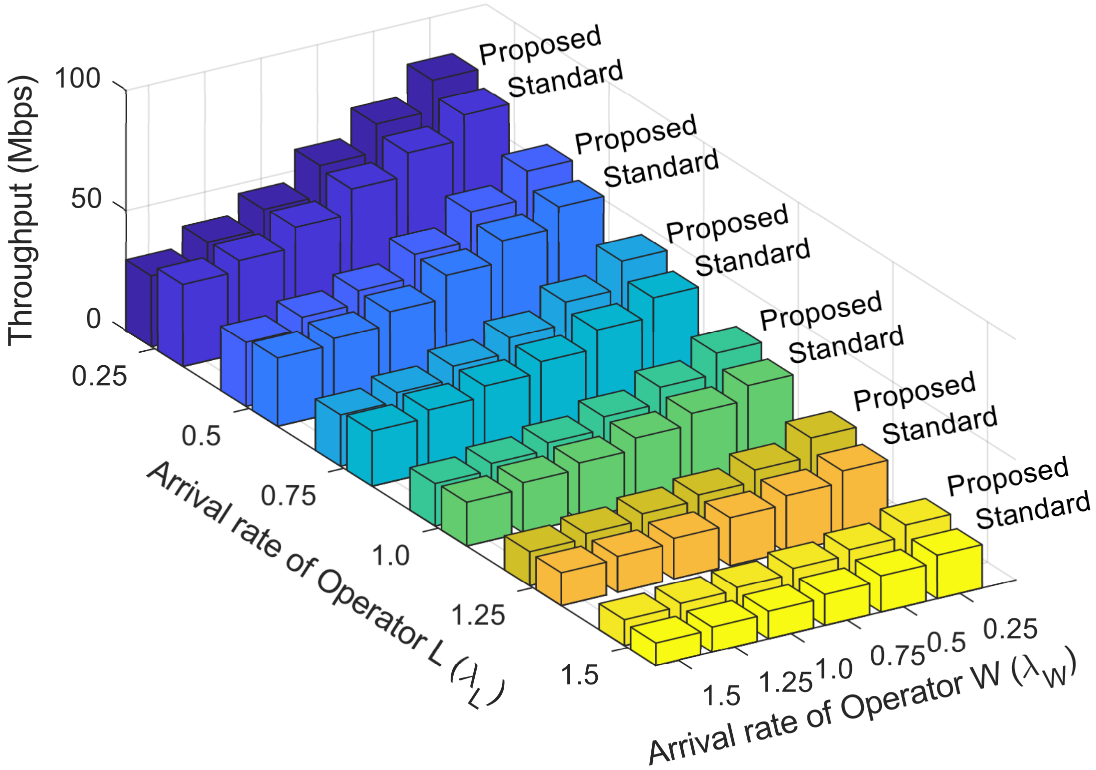


Figure 28: Throughput of the center LAA BS

Table 6: LAA throughput improvement by Proposed LAA (in %)

		Operator W (λ_W)					
		0.25	**0.5**	**0.75**	**1.0**	**1.25**	**1.5**
Operator L (λ_L)	**0.25**	7.92	5.59	2.60	-1.62	-2.00	-13.72
	0.5	10.31	6.95	0.88	1.42	1.59	-6.13
	0.75	13.72	7.83	4.97	-0.09	-4.00	-6.78
	1.0	13.51	7.77	1.90	0.41	-1.47	-2.90
	1.25	19.15	10.18	6.01	11.37	8.44	3.45
	1.5	23.68	17.50	19.89	14.16	15.16	14.47

Note: Blue means improvement. Higher value the better.

Table 7: LAA delay improvement by Proposed LAA (in %)

		Operator W (λ_W)					
		0.25	**0.5**	**0.75**	**1.0**	**1.25**	**1.5**
Operator L (λ_L)	**0.25**	10.44	2.23	-1.28	-7.84	-7.27	-17.75
	0.5	11.74	6.48	-0.47	-0.74	-2.25	-7.28
	0.75	10.48	8.98	4.57	0.62	-0.76	-3.84
	1.0	12.00	7.80	3.20	0.29	-1.78	-1.81
	1.25	10.80	11.79	6.46	12.20	13.20	9.22
	1.5	16.35	27.10	27.11	42.70	35.44	14.73

Note: Blue means improvement. Higher value the better.

Fig. 29 shows the delay of Proposed LAA and Standard LAA with varying λ_L and λ_W, and Table 7 presents how much Proposed LAA reduces delay against Standard LAA which is calculated as

$$\text{Delay improvement} = \frac{D_{L,S} - D_{L,P}}{D_{L,S}} \times 100, \tag{68}$$

where $D_{L,S}$ is the delay of Standard LAA, and $D_{L,P}$ is the delay of Proposed LAA. Based on Fig. 29, the delay sharply increases as λ_L grows for both Proposed LAA and Standard LAA. Proposed LAA, however, achieves up to 42.70% smaller delay as presented in Table 7. The delay performance of Proposed LAA is improved by 7.02% on average, and for the blue entries by 12.75% on average.

Discussion: If we can selectively apply the proposed mechanism depending on (λ_L, λ_W), we can utilize the mechanism only when it's beneficial. Based on Tables 6 and 7, 23 out of 36 cases (almost 2/3 of the total cases) improve both throughput and delay against Standard LAA, which are roughly corresponding to the cases when $\lambda_L \geq \lambda_W$. In addition, the performance improvement tends to get larger with larger λ_L and smaller λ_W. This confirms that, when LAA has more active traffic than Wi-Fi and as the ratio of traffic heaviness of LAA to Wi-Fi gets larger, the proposed mechanism becomes more advantageous in improving the spatial efficiency of an LAA network. Moreover, even though selective use between Proposed LAA and Standard LAA is not applicable, Proposed LAA achieve better performance in throughput and delay, on average.

Performance Improvement of Wi-Fi

In this section, we consider the performance of Wi-Fi in terms of throughput and delay, and *compare* the performance of Wi-Fi coexisting with Proposed LAA *with* the performance of Wi-Fi

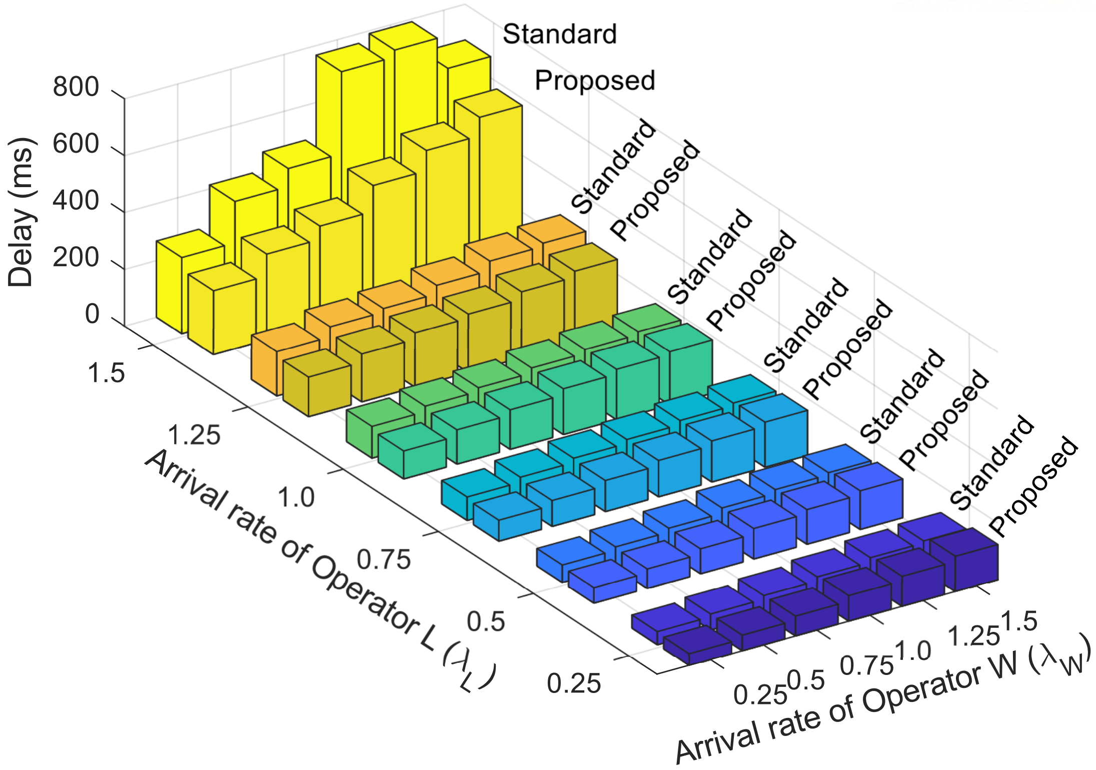


Figure 29: Delay of the center LAA BS

coexisting with Standard LAA. By such comparison, we would like to show that Proposed LAA not only improves the LAA performance but also is more beneficial to Wi-Fi. In Figs. 30 and 31, we denote the case of Wi-Fi coexisting with Standard LAA by 'Standard', and the case of Wi-Fi coexisting with Proposed LAA by 'Proposed'.

Fig. 30 shows the throughput of Wi-Fi with varying λ_L and λ_W. Similar to Fig. 28, the Wi-Fi throughput is inversely proportional to λ_L and λ_W. Table 8 presents how much the Wi-Fi throughput is improved when it is coexisting with Proposed LAA, against when it is coexisting with Standard LAA, which is calculated as

$$\text{Throughput improvement} = \frac{T_{W,P} - T_{W,S}}{T_{W,S}} \times 100, \tag{69}$$

where $T_{W,S}$ is the throughput of Wi-Fi coexisting with Standard LAA, and $T_{W,P}$ is the throughput of Wi-Fi coexisting with Proposed LAA. For $\lambda_L \leq 0.75$, it is always more beneficial for Wi-Fi to coexist with Proposed LAA, regardless of λ_W. In addition, for $\lambda_W = 0.25, 0.5$, the Wi-Fi throughput gets improved by Proposed LAA regardless of λ_L except one case. Overall, the Wi-Fi throughput is improved by as much as 22.87%, by 4.14% on average, and by 8.35% on average for the blue entries.

Fig. 31 shows the delay of Wi-Fi with varying λ_L and λ_W, and Table 9 presents how much Proposed LAA reduces the delay of coexisting Wi-Fi against Wi-Fi coexisting with Standard LAA, which is calculated as

$$\text{Delay improvement} = \frac{D_{W,S} - D_{W,P}}{D_{W,S}} \times 100 \tag{70}$$

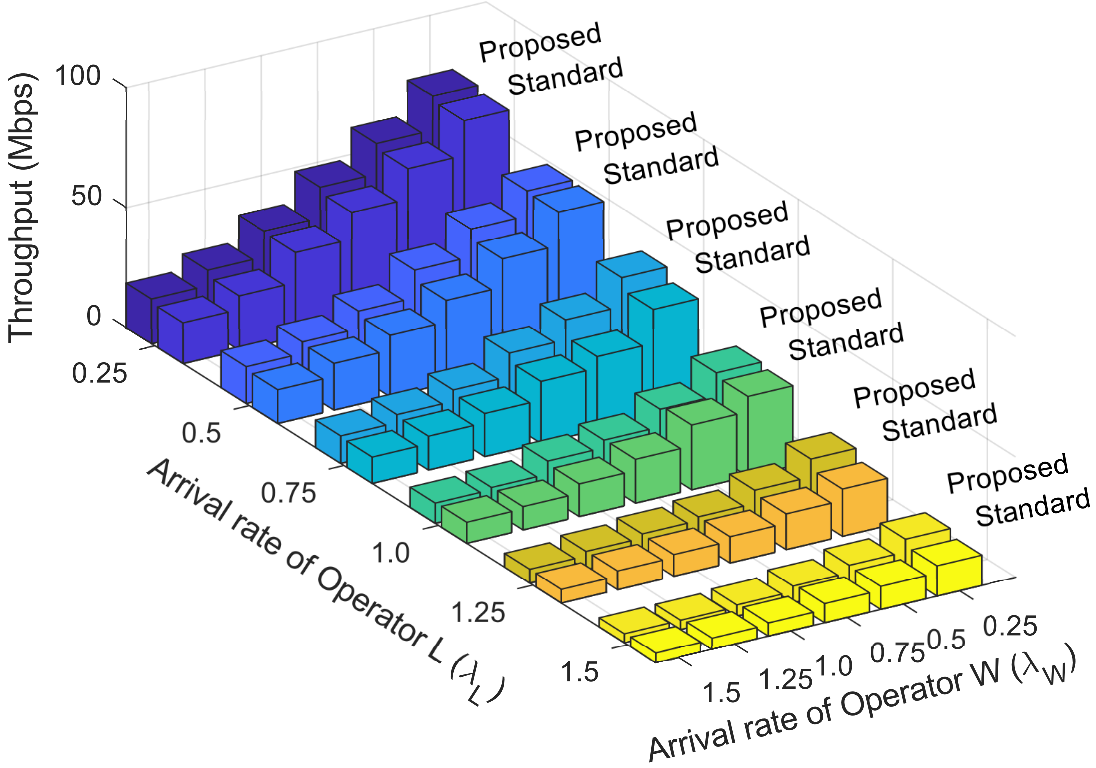


Figure 30: Throughput of the center Wi-Fi AP

Table 8: Wi-Fi throughput improvement by Proposed LAA (in %)

		Operator W (λ_W)					
		0.25	**0.5**	**0.75**	**1.0**	**1.25**	**1.5**
Operator L (λ_L)	**0.25**	2.68	3.71	4.73	1.53	10.73	9.52
	0.5	0.97	8.60	12.63	6.92	3.52	9.19
	0.75	11.17	22.87	13.69	8.32	4.79	4.58
	1.0	4.64	-6.50	-1.89	8.68	-3.45	-3.83
	1.25	21.12	10.32	-0.17	3.60	-1.89	1.30
	1.5	22.19	5.17	-12.60	-13.98	-17.87	-5.78

Note: Blue means improvement. Higher value the better.

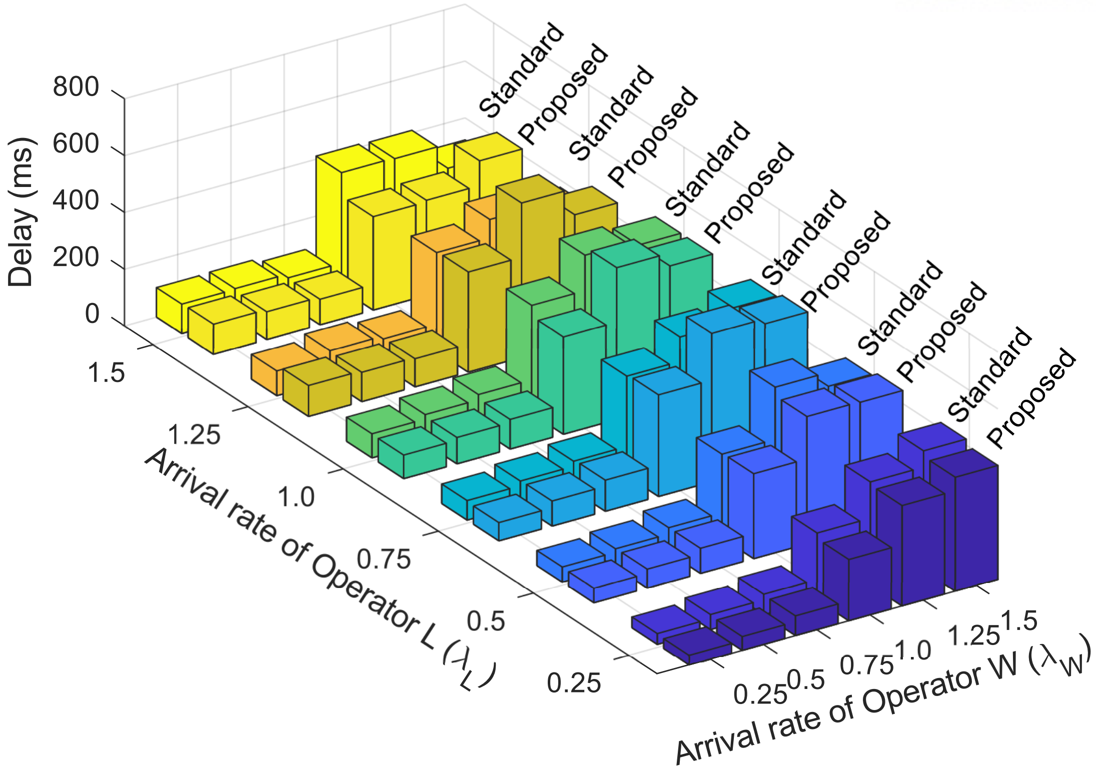


Figure 31: Delay of the center Wi-Fi AP

where $D_{W,S}$ is the delay of Wi-Fi coexisting with Standard LAA, and $D_{W,P}$ is the delay of Wi-Fi coexisting with Proposed LAA. As shown in Fig. 31, the delay sharply increases at $\lambda_W = 1.0$ and beyond. Specifically, the average values of $D_{W,S}$ and $D_{W,P}$ are 84.84 and 83.77 ms when $\lambda_W < 1.0$, but 397.36 and 408.13 ms when $\lambda_W \geq 1.0$; that is, the delay $D_{W,S}$ and $D_{W,P}$ at $\lambda_W \geq 1.0$ are 4.68 and 4.87 times larger than the ones at $\lambda_W < 1.0$. In the meantime, Table 9 shows that the delay improvement by Proposed LAA is -0.61% on average, i.e., choosing between Standard LAA and Proposed LAA has a minimal impact on the average-sense Wi-Fi delay. Combined together, the results suggest that Proposed LAA does not affect the Wi-Fi delay much, while the arrival rate of Wi-Fi has more significant influence.

Discussion: The results on Wi-Fi's throughput and delay revealed that Proposed LAA can help Wi-Fi enhance its throughput by 8.35% (if Proposed LAA is adopted selectively, as discussed earlier in Section 3.4), while maintaining similar delay performance. Based on Tables 8 and 9, Proposed LAA can improve throughput in 26 out of 36 cases (more than 2/3 of the total cases) and both throughput and delay in 13 out of 36 cases, compared to Standard LAA. Combining all the results in Tables 6, 7, 8, and 9, 14 out of 36 cases can improve LAA's throughput and delay and Wi-Fi's throughput, while achieving similar performance in Wi-Fi's delay. In summary, we can conclude that Proposed LAA can bring enough benefits for both LAA and Wi-Fi if it is adopted either exclusively or together with Standard LAA.

Table 9: WiFi delay improvement by Proposed LAA (in %)

		Operator W (λ_W)					
		0.25	**0.5**	**0.75**	**1.0**	**1.25**	**1.5**
Operator L (λ_L)	**0.25**	9.12	8.10	4.09	8.65	2.80	4.72
	0.5	4.33	-1.83	-0.54	-1.42	6.35	-3.85
	0.75	6.59	-3.05	-6.18	-2.00	-19.60	-3.37
	1.0	2.28	7.52	12.20	10.11	-5.70	0.10
	1.25	-24.57	3.19	-3.13	-1.32	-31.67	-20.37
	1.5	-1.86	8.11	2.92	19.95	18.02	-30.70

Note: Blue means improvement. Higher value the better.

3.5 Conclusion and Future Work

We proposed SCD-based Wi-Fi preamble detection for LAA, along with Wi-Fi frame duration detection and dynamic ED threshold selection. The proposed methods were evaluated via MATLAB simulations and USRP-based experiments, from which we conclude that our proposal allows LAA to reliably recognize the existence and the timing of Wi-Fi frames by only utilizing LAA's own time domain samples. Moreover, via extensive NS-3 based simulations, we have further shown that the proposed LAA mechanism not only enhances the spatial efficiency of LAA, but also protects coexisting Wi-Fi better than the legacy LAA.

In the future, we would like to apply the developed methods to more diverse usage scenarios such as indoor/outdoor combined environments, uplink/downlink mixed traffic, etc.

IV AI-powered Unlicensed Bands Communication for Enhanced Coexistence Performance

4.1 Introduction

LTE Licensed Assisted Access, also known as LTE-LAA, is first introduced in 3GPP Release 13 [82] to make LTE more versatile in utilizing the spectrum. LTE is originally developed for usage in the licensed spectrum, and thus does not consider the coexistence with other wireless technologies. Therefore, LTE-LAA has defined additional mechanisms to operate in the unlicensed spectrum, such as listen-before-talk and the exponential backoff mechanism. Moreover, to peacefully coexist with other technologies like Wi-Fi, 3GPP TR 36.889 [83] specifies the following guideline: LTE-LAA should not have an influence on the services of coexisting Wi-Fi networks more than an additional Wi-Fi network would have, in terms of throughput and latency.

Such a guideline, however, is not always achievable because the influence varies with many factors including the location of devices, building structures, etc. As a result, LTE-LAA specifies its standard in a conservative way, e.g., [84] revealed that the performance of Wi-Fi is even improved when coexisting with LTE-LAA rather than another Wi-Fi. In other words, the standard LTE-LAA has room to improve its performance by slightly deteriorating the Wi-Fi's performance while still limiting its impact on the coexisting Wi-Fi network as initially intended.

We believe machine learning techniques can help to solve the aforementioned problem, e.g., reinforcement learning with the adversarial concept. The adversarial approach with a discriminator network has been widely used since the Generative Adversarial Network (GAN) introduced the adversarial concept in the machine learning domain [85,86]. A key strength of the discriminator neural network is learning the distribution of the input data by approximating a distribution function via a neural network.

Motivated by such a strength, there exist several pieces of research combining reinforcement learning with a discriminator network, such as using a discriminator output as a reward by comparing the distribution of genuine data with the resulting data of reinforcement learning [87, 88]. More specifically, [87] proposes an adversarially trained agent that takes actions of an image editor to generate images based on input images. In addition, [88] presents a model-free imitation learning algorithm to train an agent that mimics export demonstrations under a given situation.

In this chapter, we propose the Reinforcement Learning Algorithm for Coexistence in Unlicensed spectrum bands (RLA-CU). Specifically, RLA-CU improves the performance of coexisting networks via reinforcement learning while satisfying the guidelines of 3GPP through reinforcement learning with a discriminator network. We evaluate the performance of RLA-CU and compare it with the standard LAA.

This chapter is organized as follows. Section 4.2 explains some prerequisite knowledge including LTE-LAA and IEEE 802.11. Then, Section 4.3 introduces the system model and considered scenarios, and Section 4.4 proposes the proposed RLA-CU. Section 4.5 evaluates the proposed algorithm, and finally the chapter concludes with Section 4.6.

4.2 Preliminaries

In this section, we briefly explain the difference between IEEE 802.11 and LTE-LAA in terms of their coexistence mechanisms in unlicensed spectrum bands.

IEEE 802.11

One of the key features of IEEE 802.11 [89] is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to coexist not only with other IEEE 802.11 devices but also with other incompatible technologies in the unlicensed spectrum. Before initiating transmission, an IEEE 802.11 device determines the wireless channel state between Busy or Idle, where the former implies another device is transmitting in the same channel with non-trivial received power. In addition, a threshold for the channel state is varying according to whether an IEEE 802.11 preamble is detected or not from the received signal. More specifically, when the bandwidth of 20 MHz is being utilized, the threshold of -82 dBm is applied when a preamble is detected and -62 dBm is used otherwise. Please note that in this chapter we do not modify any specifications of the IEEE 802.11 standard.

LTE Licensed-Assisted Access (LAA)

LTE-LAA [90] adopts the listen-before-talk mechanism with energy detection to determine the channel state before initiating transmission. Contrary to IEEE 802.11, LTE-LAA only considers the strength of the received power to decide the channel state, where the detection threshold is varying according to whether LTE-LAA coexists with other technologies or not. To be more specific, when LTE-LAA coexists with WLAN, the threshold of -72 dBm is used when it utilizes the bandwidth of 20 MHz and has the transmission power of 23 dBm; otherwise, -52 dBm is applied. The detailed information about the relationship between the energy detection threshold and the transmission power will be further described in Section 4.3.

4.3 System Model

In this section, we define our considered coexistence scenarios and describe the method for exchanging data, which will be used for the proposed algorithm in Section 4.4.

Coexistence Scenarios

In this chapter, we consider a network operator's perspectives, especially for the downlink traffic. Let an *LAA operator* refer to a network operator that deploys LTE-LAA eNBs, and let a *Wi-Fi* *operator* refer to a network operator that deploys 802.11n APs.[28] Also, we assume that an LAA operator manages n_l eNBs for n_{ue} UEs, while a Wi-Fi operator manages n_w APs for n_{sta} stations (STAs). Note that we focus on the coexistence between LTE-LAA and IEEE 802.11 which is the most significant coexistence issue in the unlicensed spectrum.

We define three coexistence scenarios as follows.

- **LAA–WiFi**: legacy LTE-LAA devices (that follow the current LTE-LAA standard) coexist with IEEE 802.11 devices.
- **LAA$^+$–WiFi**: LTE-LAA devices controlled by RLA-CU coexist with IEEE 802.11 devices.
- **WiFi–WiFi**: Only IEEE 802.11 devices exist in the network.

In the above, the LAA-WiFi scenario serves as a baseline to present how much the proposed RLA-CU in the LAA$^+$–WiFi scenario can enhance the performance of the LAA network against the legacy LAA. In addition, comparison between LAA-WiFi and WiFi-WiFi can measure how much the Wi-Fi operator is affected by the coexisting legacy LAA network.

Data Exchange Method

We assume that each of LAA and Wi-Fi operators monitors and controls its own devices, as described in detail below.

Network Monitoring A Wi-Fi operator periodically monitors the achieved throughput and the latency of each Wi-Fi STA associated with the operator's APs. On the other hand, an LAA operator collects not only the achieved throughput and the latency, but also average Received Signal Strength Indicator (RSSI) and the channel occupancy values which are newly presented in 3GPP Release 13 to support communications in the unlicensed spectrum band [91]. Average RSSI is used to estimate the overall channel load condition such as interference, and channel occupancy is defined as the ratio of the channel's busy duration to the measurement duration, based on the measured RSSI. The measurement duration and the measurement period are configured through RSSI measurement timing configuration (RMTC) [92]. For example, the measurement duration can be set in the range of 1 to 5 milliseconds, while the measurement period can be one of 40, 80, 160, 320 milliseconds.

Controlling an LTE-LAA Network Based on the collected information, an LAA operator changes LTE-LAA configurations, such as the energy detection threshold and the transmission power.According to [90], the maximum energy detection threshold E_{max} (in dBm) varies

[28] IEEE 802.11n is used for evaluating the coexistence performance of LTE-LAA [83], while IEEE 802.11ac is adopted for evaluating the coexistence performance of NR-U [5].

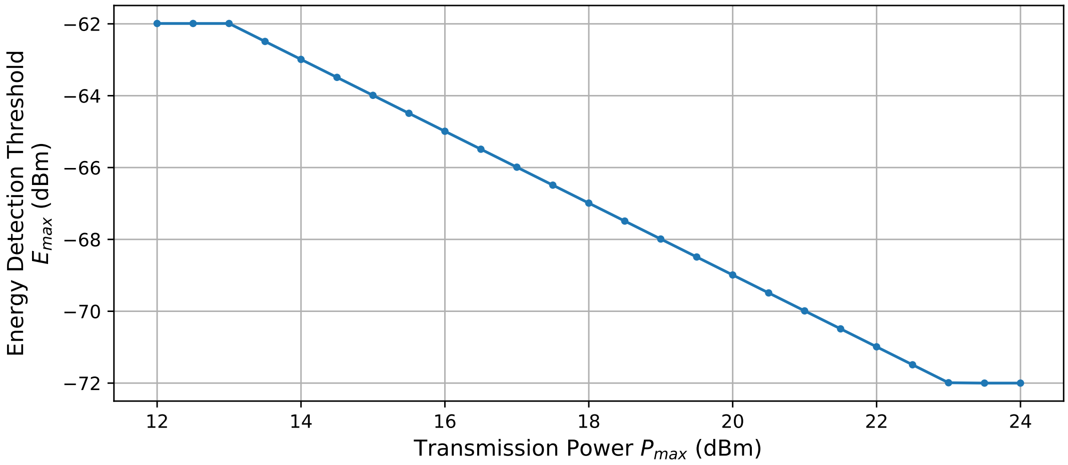


Figure 32: Variation of E_{max} with varying P_{max}

with the maximum transmission power P_{max} (in dBm) as follows:

$$E_{max} = \max \left\{ \begin{array}{l} -72 + 10 \cdot \log_{10}(B/20) \\ \min \left\{ \begin{array}{l} T \\ T + 13 + 10 \cdot \log_{10}(B/20) - P_{max} \end{array} \right\} \end{array} \right\} \tag{71}$$

where $T = 10 \cdot \log_{10}\left(3.16228 \cdot 10^{-8} \cdot B\right)$ and B is the channel bandwidth in MHz, when the transmission includes PDSCH. Fig. 32 shows the variation of E_{max} according to P_{max}. Finally, we assume that the transmission power P_{tx} is in the range of $[P_{min}, P_{max}]$, and the energy detection threshold E_{th} is in the range of $[E_{min}, E_{max}]$.

4.4 Proposed Algorithm

In this section, we propose the Reinforcement Learning Algorithm for Coexistence in Unlicensed spectrum bands (RLA-CU) that enhances the coexistence between LTE-LAA and Wi-Fi by adjusting the control parameters of LTE-LAA through machine learning techniques. More specifically, the proposed algorithm trains an agent that administers all the eNBs of the LAA operator to improve the level of coexistence with a Wi-Fi operator, while maintaining the performance of the LAA operator. To achieve this goal, we install an agent on the LAA operator's server and a discriminator network on the Wi-Fi operator's server. An overview of the proposed model is presented in Fig. 33.

Discriminator on the Wi-Fi Operator

The goal of the discriminator on the Wi-Fi operator is to measure the coexistence level based on the performance of the Wi-Fi operator. The coexistence level is measured in the following manner. First, all stations connected to the Wi-Fi operator report the achieved throughput and latency to the operator's server every d_w seconds. The gathered information is defined as a

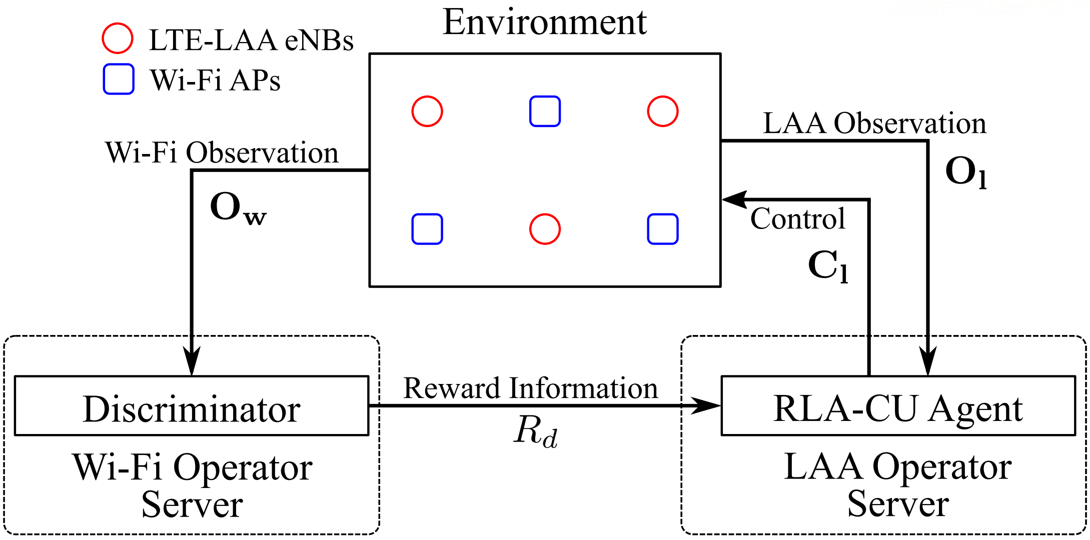


Figure 33: Overview of RLA-CU

vector $\mathbf{O_w} = [o_1^{w;t}, \dots, o_{n_{sta}}^{w;t}, o_1^{w;l}, \dots, o_{n_{sta}}^{w;l}]$ where $o_n^{w;t}$ and $o_n^{w;l}$ are the throughput and latency of station n normalized by m_t and m_l, respectively, and m_t and m_l are normalization values of throughput and latency. The discriminator analyzes $\mathbf{O_w}$ to determine the level of coexistence R_d. For example, the coexistence level is high when the distribution of $\mathbf{O_w}$ is similar to the distribution measured in the WiFi-WiFi scenario. Otherwise, the level is judged as low.

Reinforcement Learning Agent on the LAA Operator

The goal of the agent on an LAA operator is to control eNBs according to the operator's policy. The agent works in the order of observation and execution. First, similar to a Wi-Fi operator, all UEs connected to the LAA operator notify the measurements such as average RSSI, channel occupancy, throughput and latency to the operator's server every d_l seconds. The agent uses the information vector as a input to the reinforcement learning algorithm, where the information vector is defined as $\mathbf{O_l} = [o_1^{l;r}, \dots, o_{n_{ue}}^{l;r}, o_1^{l;c}, \dots, o_{n_{ue}}^{l;c}, o_1^{l;t}, \dots, o_{n_{ue}}^{l;t}, o_1^{l;l}, \dots, o_{n_{ue}}^{l;l}]$ where $o_n^{l;r}$ is an average RSSI normalized by m_r, $o_b^{l;c}$ is a channel occupancy ratio, and $o_n^{l;t}$ and $o_n^{l;l}$ are the throughput and latency of UE n normalized by m_t and m_l, respectively. This information is used to understand the wireless channel environment such as estimating the performance of a coexisting Wi-Fi operator.

Based on this observation, the agent tasks an action $\mathbf{C_l} = [P_{tx}, E_{th}]$ that should satisfy Eq. (71). In addition, an action is applied to all eNBs in an LAA operator. Finally, we define the reward function as

$$R = R_d \times \left(\frac{1}{n_{ue}} \sum_{n=1}^{n_{ue}} o_n^{l;t} - \frac{1}{n_{ue}} \sum_{n=1}^{n_{ue}} o_n^{l;l} \right), \tag{72}$$

which means that the reward is larger when the level of coexistence is high and an LAA operator achieves high average throughput with low average latency.

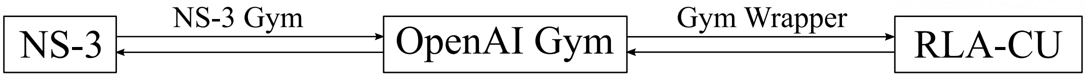


Figure 34: Simulator System for RLA-CU

4.5 Experiment

In this section, we first introduce the simulation system for evaluating the coexistence scenarios. After that, we present the design of RLA-CU with evaluation results.

Simulation System Architecture

The simulation system consists of three layers, RLA-CU, OpenAI Gym Interfaces (NS-3 Gym and Wrapper for RLA-CU) and NS-3 Simulation Environment as presented in Fig. 34. Each layer is briefly described as follows.

NS-3 for LAA-WiFi Coexistence Simulation NS-3 [77] is a network simulator based on discrete-time events. In this chapter, we adopt a modified version of NS-3 to simulate a situation where LTE-LAA and Wi-Fi networks coexist.

OpenAI Gym OpenAI Gym [93] is the popular framework for reinforcement learning providing an interface between a reinforcement learning algorithm and a simulation environment. For example, when a simulator generates an arbitrary environment, OpenAI Gym sends an action to an agent in the environment and receives an information as a consequence of the action such as a variation in the environment (called ‘observation’), an assessment of the variation (called ‘reward’), information on whether the simulator is terminated or not due to the action (called ‘done’), and extra information (called ‘info’). An example code is as follows,

```
Observation, Reward, Done, Info = env.step(Action)
```

where *env.step()* is the function with which the environment can take an action.

OpenAI Gym also provides a feature called Wrapper, which transforms an environment in a modular way. For example, the Observation Wrapper reconstructs the structure of observation and sends it to a reinforcement learning algorithm instead of keeping the original form received from an environment. In our experiment, we use the Observation Wrapper as a buffer to obtain the statistical information.

NS-3 Gym NS-3 Gym [94] helps us use the NS-3 simulator as an environment for OpenAI Gym through the socket communication between NS-3 and OpenAI Gym. To use NS-3 Gym in our experiment, we need to configure NS-3 regarding what kind of observation is transferred and how to react when receiving an action.

Table 10: Simulation Parameter

Simulation Parameter	**Value**
n_l, n_w	3
n_{ue}, n_{sta}	15
P_{min}	13 dBm
P_{max}	23 dBm
E_{min}	-82 dBm
d_w, d_l	10 seconds
m_t	150 Mbps
m_l	300 ms
m_r	-101 dBm

Simulation Environment

We consider an indoor simulation environment based on [5, 83]. Although [83] describes the indoor environment to evaluate the performance of LTE-LAA, it has some drawbacks such as the distribution of links is not properly considered [5]. Therefore, we compose the simulation environment based on [83] except for the layout of the indoor scenario and the path loss model, which are adopted from [5]. Simulation parameters are summarized in Table 10.

In the following sections, we illustrate the discriminator network and the reinforcement algorithm, along with the performance evaluation results.

Discriminator Network

For the discriminator network, we build a multilayer perceptron (MLP), where the input layer takes 30 values and the output layer presents the classification result though the Sigmoid activation function. There also exist four hidden layers, each of which has 128 nodes with the ReLU activation function. Finally, we apply a dropout rate of 0.3 and adopt the binary cross-entropy loss function with Nadam optimizer.

To train the discriminator network, we build a dataset including the throughput and latency data of the Wi-Fi operator for LAA-WiFi and WiFi-WiFi scenarios. For each scenario, we consider 482 random topologies, for each of which we measure $\mathbf{O_w}$ for 31 times. As a result, we collect 29,884 measurement data (14,942 for each case), and split them into two groups, one group with 20,918 data for training and another group with 8,966 data for testing (at a ratio of 7:3).

When we train the network with the batch size of 62 and 10 epochs, the proposed discriminator achieves 96.152% accuracy for the training dataset, and 95.126% accuracy for the test dataset, verifying that using throughput and latency the Wi-Fi operator can distinguish whether the Wi-Fi operator's service is affected similarly to coexisting with another Wi-Fi operator or not.

Reinforcement Learning Agent

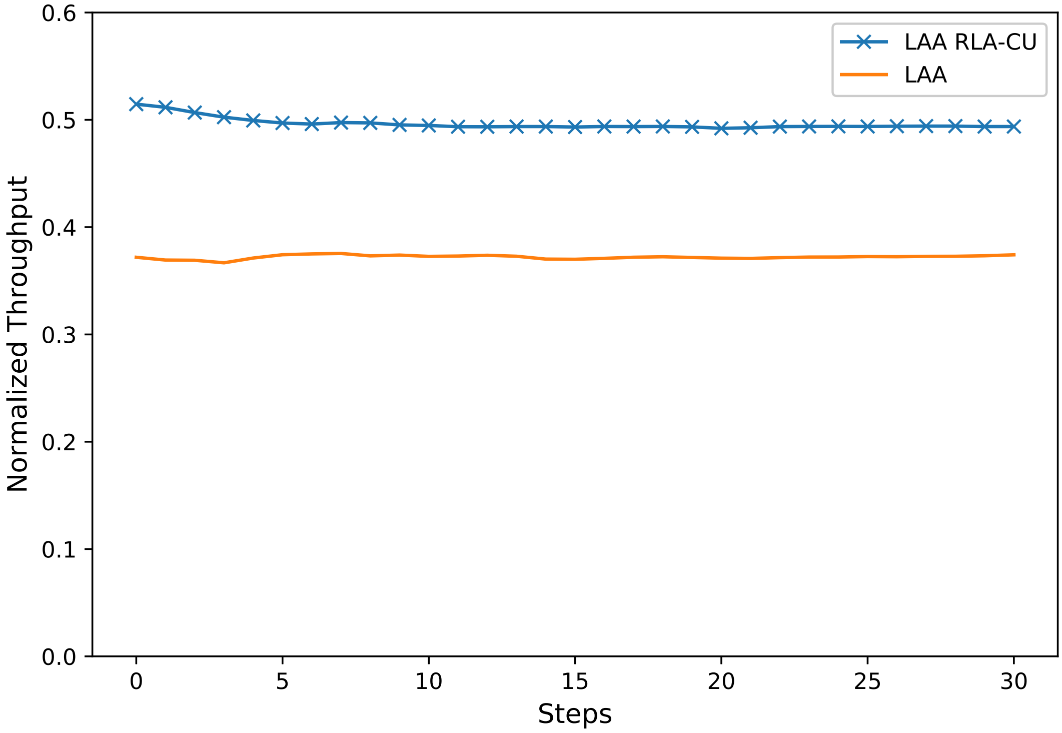
We have used the twin Delayed DDPG (TD3) [95] method to train an agent of an LAA operator. Since TD3 can be used for the continuous action space, the agent can fine-tune the values of P_{tx} and E_{th}. Note that we followed the architecture of [95].

We evaluate the RLA-CU on the seven random topologies and present the average results as follows. Fig. 35 shows that LAA with RLA-CU achieves better performance in terms of throughput and latency than the conventional LAA. Fig. 36 presents the performance variation of a Wi-Fi operator under three scenarios. In the LAA$^+$-WiFi scenario, the coexisting Wi-Fi network achieves a similar performance of throughput and latency compared to the Wi-Fi network of the LAA-WiFi scenario.

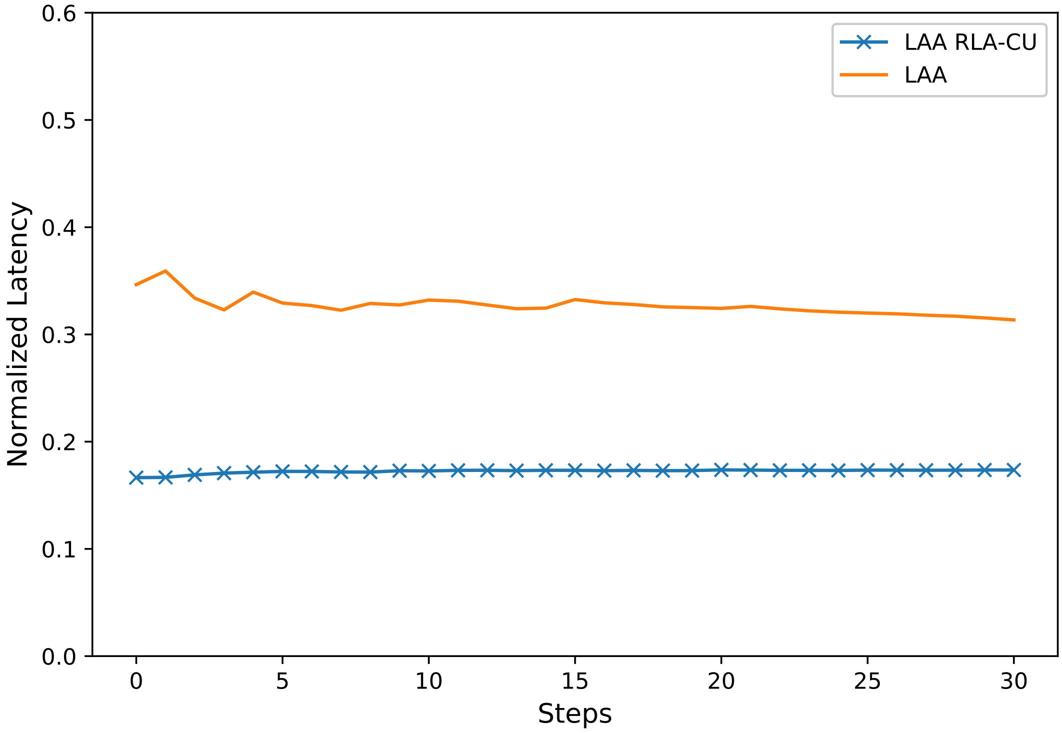
4.6 Conclusion

In this chapter, we proposed an algorithm that enhances the coexistence performance of LTE-LAA and Wi-Fi networks using reinforcement learning with a discriminator neural network. Through extensive simulations, we showed that the proposed discriminator network can distinguish the two coexistence scenarios, i.e., Wi-Fi with Wi-Fi vs. Wi-Fi with LAA, with high probability. Based on the discriminator, the proposed algorithm enhances not only the performance of an LAA network but also the performance of a coexisting Wi-Fi network.

In the future, we would like to extend our model to address more diverse situations, such as varying the position of eNBs and APs with mobile UEs and Wi-Fi stations. In addition, we would like to propose the coexistence mechanism for 5G NR operating in the unlicensed spectrum.

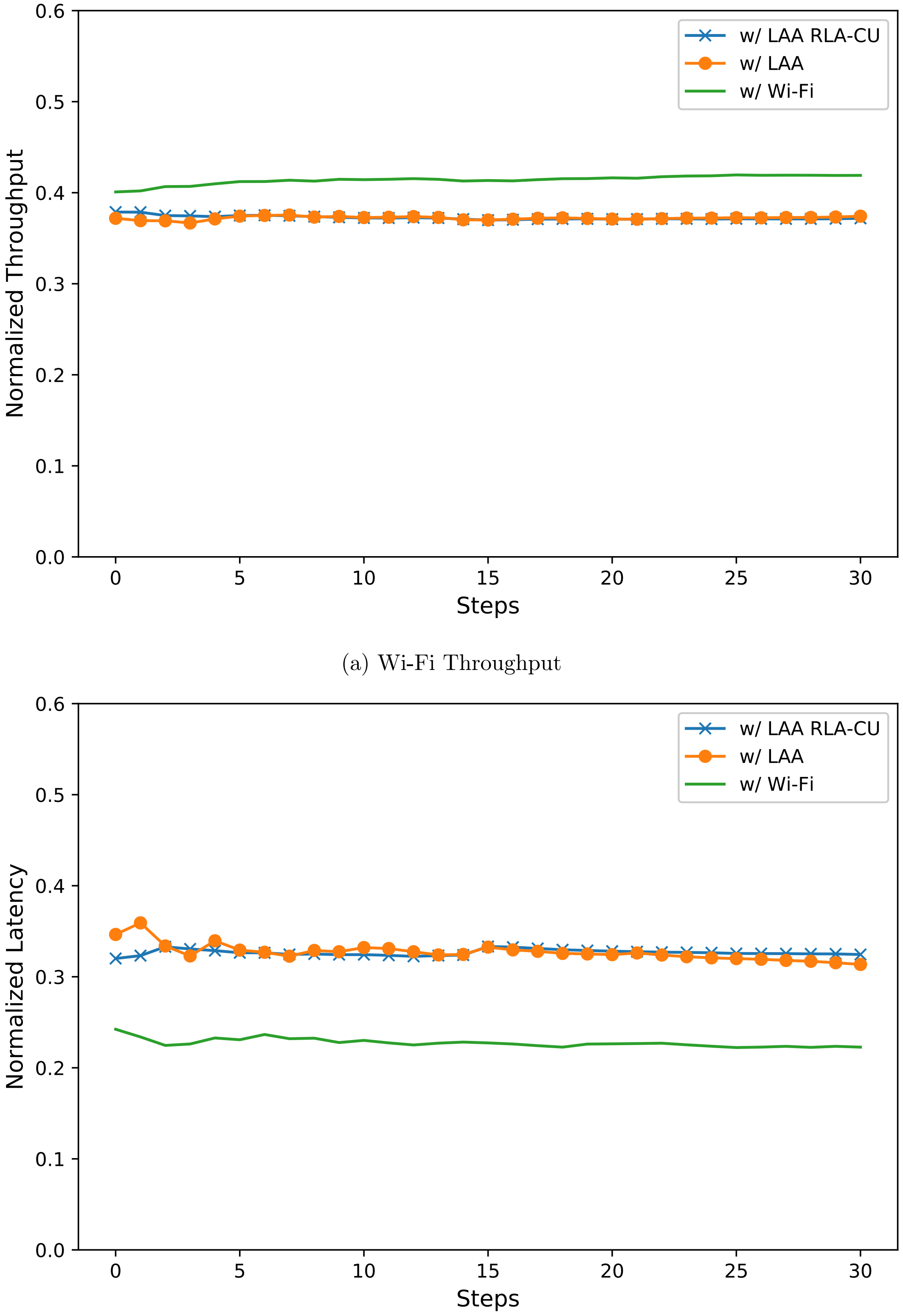


(a) LAA Throughput



(b) LAA Latency

Figure 35: Performance variation of an LAA operator



(a) Wi-Fi Throughput

(b) Wi-Fi Latency

Figure 36: Performance variation of a Wi-Fi operator

V Conclusion

The 5G achieves progress compared to its predecessor in terms of massive Machine Type Communications (mMTC), enhanced Mobile Broadband (eMBB), and Ultra Reliable Low Latency Communications (URLLC) by applying mobile edge cloud and unlicensed spectrum band operations, but also it raises problems not considered before. This dissertation has addressed algorithms to realize more efficient 5G networks in terms of eMBB and URLLC as follows.

Quality-of-Experience based computation offloading framework is proposed to make a strategy to maximize user satisfaction. The proposed framework considers energy consumption, latency, and monetary cost to measure user satisfaction. And then, based on the measured user satisfaction, the optimal strategy is determined in two steps; selecting the computation resource of the mobile edge cloud and scheduling for data transmission. Through extensive simulation, it is proved that the proposed framework enhances user satisfaction under various environments.

The dynamic CCA threshold algorithm with the lightweight Wi-Fi frame detection is introduced for enhancing the spatial efficiency of an LTE-LAA network. The proposed algorithm first detects a Wi-Fi preamble without extra hardware, and then the different CCA threshold is applied according to the detection result. When the proposed algorithm is implemented on a USRP device, it successfully detects a Wi-Fi preamble with high probability. In addition, through the NS-3 simulation, the proposed dynamic CCA threshold algorithm enhances the performance of LTE-LAA and Wi-Fi devices in terms of throughput and latency.

The adaptive control algorithm for an LTE-LAA network is presented for adjusting the network configuration based on the situation assessment. A discriminator neural network estimates a coexistence situation between LTE-LAA and Wi-Fi networks. With this assessment, the reinforcement learning algorithm tries to improve the coexistence between LTE-LAA and Wi-Fi networks. Through the NS-3 simulation, the proposed AI-based algorithm improves LTE-LAA and Wi-Fi networks by continuously adjusting the configuration of the LTE-LAA network.

References

[1] “New GSMA Study: 5G to Account for 155G Network Launches Accelerate,” Feb 2019. [Online]. Available: https://www.gsma.com/newsroom/press-release/new-gsma-study-5g-to-account-for-15-of-global-mobile-industry-by-2025/

[2] Radiocommunication Sector of ITU (ITU-R), “IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond,” International Telecommunication Union (ITU), Technical Report (TR) M.2083-0, 09 2015.

[3] “Multi-access Edge Computing (MEC),” Nov 2019. [Online]. Available: https://www.etsi.org/technologies/multi-access-edge-computing

[4] 3GPP, “Physical layer procedures,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.213, 12 2015, version 13.0.0.

[5] ——, “Study on NR-based access to unlicensed spectrum,” 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.889, 12 2018, version 16.0.0.

[6] S. Hong and H. Kim, “Qoe-aware computation offloading scheduling to capture energy-latency tradeoff in mobile clouds,” in *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2016, pp. 1–9.

[7] ——, “Qoe-aware computation offloading to capture energy-latency-pricing tradeoff in mobile clouds,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2174–2189, Sep. 2019.

[8] S. Hong, H. Lee, H. Kim, and H. J. Yang, “Lightweight wi-fi frame detection for licensed assisted access lte,” *IEEE Access*, vol. 7, pp. 77 618–77 628, June 2019.

[9] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021,” White Paper, February 2017.

[10] G. Crabtree, E. Kocs, and L. Trahey, “The energy-storage frontier: Lithium-ion batteries and beyond,” *MRS Bulletin*, vol. 40, no. 12, pp. 1067–1078, December 2015.

[11] M. Halpern, Y. Zhu, and V. J. Reddi, “Mobile CPU’s rise to power: Quantifying the impact of generational mobile CPU design trends on performance, energy, and user satisfaction,”

in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, March 2016, pp. 64–76.

[12] J. Power and Associates, "Battery Life: Is That All There Is?" 2012. [Online]. Available: http://www.jdpower.com/resource/jd-power-insights-i-battery-life-all-there

[13] K. Kumar and Y. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" *IEEE Computer*, vol. 43, no. 4, pp. 51–56, April 2010.

[14] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making Smartphones Last Longer with Code Offload," in *Proc. ACM MobiSys*, 2010.

[15] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic Execution between Mobile Device and Cloud," in *Proc. ACM EuroSys*, 2011.

[16] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, October-December 2009.

[17] Z. Guan and T. Melodia, "Cloud-assisted smart camera networks for energy-efficient 3d video streaming," *IEEE Computer*, vol. 47, no. 5, pp. 60–66, May 2014.

[18] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. Wu, "Energy-Optimal Mobile Cloud Computing under Stochastic Wireless Channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, September 2013.

[19] E. Uysal-Biyikoglu, B. Prabhakar, and A. E. Gamal, "Energy-Efficient Packet Transmission Over a Wireless Link," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 487–499, August 2002.

[20] The Nielson Company, "The Female/Male Digital Divide," 2014. [Online]. Available: http://www.nielsen.com/us/en/insights/news/2014/the-female-male-digital-divide.html

[21] L. Xiang, S. Ye, Y. Feng, B. Li, and B. Li, "Ready, Set, Go: Coalesced offloading from mobile devices to the cloud," in *Proc. INFOCOM*, April 2014, pp. 2373–2381.

[22] W. Zhang, R. Fan, Y. Wen, and F. Liu, "Energy-efficient mobile video streaming: A location-aware approach," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 1, pp. 6:1–6:16, Aug. 2017.

[23] F. Liu, P. Shu, and J. C. Lui, "AppATP: An Energy Conserving Adaptive Mobile-Cloud Transmission Protocol," *IEEE Trans. Comput.*, vol. 64, no. 11, pp. 3051–3063, November 2015.

[24] L. Ruan, Z. Liu, X. Qiu, Z. Wang, S. Guo, and F. Qi, "Resource allocation and distributed uplink offloading mechanism in fog environment," *Journal of Communications and Networks*, vol. 20, no. 3, pp. 247–256, June 2018.

[25] H. Shah-Mansouri, V. W. S. Wong, and R. Schober, "Joint optimal pricing and task scheduling in mobile cloud computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5218–5232, Aug 2017.

[26] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 81–93, Jan 2015.

[27] W. Zhang and Y. Wen, "Energy-efficient task execution for application as a general topology in mobile cloud computing," *IEEE Trans. Cloud Comput.*, pp. 1–1, December 2015.

[28] D. Yao, C. Yu, L. Yang, and H. Jin, "Using crowdsourcing to provide qos for mobile cloud computing," *IEEE Trans. Cloud Comput.*, pp. 1–1, Dec 2015.

[29] Z. Guan and T. Melodia, "The value of cooperation: Minimizing user costs in multi-broker mobile cloud computing networks," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 780–791, Oct 2017.

[30] K. Laghari, N. Crespi, and K. Connelly, "Toward Total Quality of Experience: A QoE Model in a Communication Ecosystem," *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 58–65, April 2012.

[31] D. Bertsekas, *Dynamic Programming and Optimal Control.* Belmont, MA: Athena Scientific, 2005.

[32] ETSI GS MEC 001, *Mobile Edge Computing (MEC) Terminology*, ETSI Std., March 2016.

[33] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile Edge Computing - A key technology towards 5G," ETSI, Tech. Rep., September 2015.

[34] ETSI GS MEC 003, *Mobile Edge Computing (MEC); Framework and Reference Architecture*, ETSI Std., March 2016.

[35] ETSI GS MEC 002, *Mobile Edge Computing (MEC); Technical Requirements*, ETSI Std., March 2016.

[36] S. Zhuravlev, J. Saez, S. Blagodurov, A. Fedorova, and M. Prieto, "Survey of Energy-Cognizant Scheduling Techniques," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1447–1464, July 2013.

[37] A. K. Singh, P. Dziurzanski, H. R. Mendis, and L. S. Indrusiak, “A Survey and Comparative Study of Hard and Soft Real-Time Dynamic Resource Allocation Strategies for Multi-/Many-Core Systems,” *ACM Comput. Surv.*, vol. 50, no. 2, pp. 24:1–24:40, Apr. 2017.

[38] H. Wang and N. Moayeri, “Finite-State Markov Channel—A Useful Model for Radio Communication Channels,” *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163–171, February 1995.

[39] Q. Zhang and S. Kassam, “Finite-state markov model for rayleigh fading channels,” *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, November 1999.

[40] N. Abu-Ali, A. E. M. Taha, M. Salah, and H. Hassanein, “Uplink scheduling in lte and lte-advanced: Tutorial, survey and evaluation framework,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1239–1265, Third 2014.

[41] 3GPP TS 36.104, *Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception*, 3GPP Std., March 2018.

[42] T. S. Rappaport, *Wireless communications: Principles and practice*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[43] J. Kwak, Y. Kim, J. Lee, and S. Chong, “Dream: Dynamic resource and task allocation for energy minimization in mobile cloud systems,” *IEEE J. Sel. Areas Commun.*, pp. 2510–2523, December 2015.

[44] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in *Proc. 2nd USENIX Conf. Hot topics in cloud computing*, 2010, pp. 4–4.

[45] E. Blem, J. Menon, T. Vijayaraghavan, and K. Sankaralingam, “ISA Wars: Understanding the Relevance of ISA Being RISC or CISC to Performance, Power, and Energy on Modern Architectures,” *ACM Trans. Comput. Syst.*, vol. 33, no. 1, pp. 3:1–3:34, Mar. 2015.

[46] Microsoft Azure, “Azure pricing,” 2017. [Online]. Available: https://azure.microsoft.com/en-us/pricing

[47] Amazon Web Services, “Per-Second Billing for EC2 Instances and EBS Volumes,” 2017. [Online]. Available: https://aws.amazon.com/blogs/aws/new-per-second-billing-for-ec2-instances-and-ebs-volumes

[48] M. Zafer and E. Modiano, “Delay-Constrained Energy Efficient Data Transmission over a Wireless Fading Channel,” in *ITA Workshop*, 2007.

[49] GSA, “Mobile Fast Facts,” February 2017. [Online]. Available: https://gsacom.com/paper/mobile-fast-facts

[50] B. Jeff, "big.LITTLE Technology Moves Towards Fully Heterogeneous Global Task Scheduling," ARM, Tech. Rep., November 2013.

[51] AnandTech, "Improving The Exynos 9810 Galaxy S9: Part 2 – Catching Up With The Snapdragon," 2018. [Online]. Available: https://www.anandtech.com/show/12620/improving-the-exynos-9810-galaxy-s9-part-2/2

[52] 3GPP TS 36.942, *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios*, 3GPP Std., March 2017.

[53] Samsung Electronics, "Samsung Galaxy S9," 2018. [Online]. Available: http://www.samsung.com/global/galaxy/galaxy-s9

[54] AnandTech, "Galaxy S9 Exynos 9810 Hands-on - Awkward First Results," 2018. [Online]. Available: https://www.anandtech.com/show/12478/exynos-9810-handson-awkward-first-results

[55] ——, "Exploring DynamIQ and ARM's New CPUs: Cortex-A75, Cirtex-A55," 2017. [Online]. Available: https://www.anandtech.com/show/11441/dynamiq-and-arms-new-cpus-cortex-a75-a55

[56] Amazon Web Services, "Amazon EC2 Instance Types," 2018. [Online]. Available: https://aws.amazon.com/ec2/instance-types

[57] J. Huang, M. Leng, and M. Parlar, "Demand Functions in Decision Modeling: A Comprehensive Survey and Research Directions," *A Journal of the Decision Sciences Institue*, vol. 44, no. 3, pp. 557–609, June 2013.

[58] K. Liu, J. Peng, H. Li, X. Zhang, and W. Liu, "Multi-device task offloading with time-constraints for energy efficiency in mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 64, no. C, pp. 1–14, Nov. 2016.

[59] 802.11 Working Group, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Computer Society Std., December 2016.

[60] H. Ye, W.-J. Choi, N. Zhang, and J. M. Gilbert, "Selecting MCS in a MIMO system," U.S. Patent US7 764 727B2, Jul. 27, 2010.

[61] Project Fi, "Phone plans and prices," 2018. [Online]. Available: https://fi.google.com/about/plan

[62] 3GPP TS 37.213, "Technical Specification Group Radio Access Network; Physical layer procedures for shared spectrum channel access (Release 15)," 3GPP, Tech. Rep., Mar. 2019.

[63] Telesystem Innovations Inc., "LTE in a Nutshell: The Physical Layer," 2010.

[64] S. Yun and L. Qiu, "Supporting WiFi and LTE Co-existence," in *Proc. IEEE INFOCOM*, 2015.

[65] K. Yoon et al., "COTA: Channel occupancy time adaptation for LTE in unlicensed spectrum," in *Proc. IEEE DySPAN*, 2017.

[66] A. Garcia-Saavedra, P. Patras, V. Valls, X. Costa-Perez, and D. J. Leith, "ORLA/OLAA: Orthogonal Coexistence of LAA and WiFi in Unlicensed Spectrum," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2665–2678, Dec 2018.

[67] Z. Guan and T. Melodia, "CU-LTE: Spectrally-efficient and Fair Coexistence between LTE and Wi-Fi in Unlicensed Bands," in *Proc. IEEE INFOCOM*, April 2016, pp. 1–9.

[68] L. Li, A. H. Jafari, X. Chu, and J. Zhang, "Simultaneous Transmission Opportunities for LTE-LAA Smallcells Coexisting with WiFi in Unlicensed Spectrum," in *Proc. IEEE ICC*, May 2016, pp. 1–7.

[69] D. Croce, D. Garlisi, F. Giuliano, N. Inzerillo, and I. Tinnirello, "Learning From Errors: Detecting Cross-Technology Interference in WiFi Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 347–356, June 2018.

[70] V. Sathya, M. Merhnoush, M. Ghosh, and S. Roy, "Energy Detection Based Sensing of Multiple Wi-Fi BSSs for LTE-U CSAT," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2018, pp. 1–7.

[71] M. Mehrnoush, V. Sathya, S. Roy, and M. Ghosh, "Analytical Modeling of Wi-Fi and LTE-LAA Coexistence: Throughput and Impact of Energy Detection Threshold," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1990–2003, Aug. 2018.

[72] J. Li, T. Y. Cheng, X. Jia, and L. M. Ni, "Throughput Optimization in WLAN/Cellular Integrated Network Using Partially Overlapped Channels," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 157–169, Jan 2018.

[73] IEEE 802.11 Working Group, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," IEEE, Tech. Rep., March 2012.

[74] R. Bhardwaj, K. Chintalapudi, and R. Ramjee, "Skip-Correlation for Multi-Power Wireless Carrier Sensing," in *Proc. USENIX NSDI*, 2017.

[75] EttusResearch, "Knowledge Base," https://kb.ettus.com/Knowledge_Base.

[76] MATLAB, "The USRP Support Package from Communications System Toolbox," https://www.mathworks.com/hardware-support/usrp.html.

[77] L. Giupponi, T. Henderson, B. Bojovic, and M. Miozzo, "Simulating LTE and Wi-Fi Coexistence in Unlicensed Spectrum with ns-3," *arXiv e-prints*, p. arXiv:1604.06826, Apr 2016.

[78] 3GPP TR 36.889, "Study on Licensed-Assisted Access to Unlicensed Spectrum (Release 13)," 3GPP, Tech. Rep., Jun. 2015.

[79] 3GPP TR 36.814, "Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects (Release 9)," 3GPP, Tech. Rep., Mar. 2017.

[80] ITU, "IMT Traffic Estimates for the Years 2020 to 2030," July 2015.

[81] G. Carneiro, P. Fortuna, and M. Ricardo, "FlowMonitor - A Network Monitoring Framework for the Network Simulator 3 (NS-3)," in *Proc. VALUETOOLS*, 2009.

[82] "3GPP Release 13." [Online]. Available: https://www.3gpp.org/release-13

[83] 3GPP, "Study on Licensed-Assisted Access to Unlicensed Spectrum," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 36.889, 06 2015, version 13.0.0.

[84] Broadcom Corporation, "Coexistence Evaluation Results Using LBT Category 4 for Wi-Fi DL and LAA DL only Scenario," Technical Report R1-152936, 05 2015.

[85] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 09 2017.

[86] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[87] Y. Ganin, T. Kulkarni, I. Babuschkin, S. M. A. Eslami, and O. Vinyals, "Synthesizing Programs for Images using Reinforced Adversarial Learning," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, Jul 2018, pp. 1666–1675.

[88] J. Ho and S. Ermon, "Generative Adversarial Imitation Learning," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4565–4573.

[89] "IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2016*, pp. 1–3534, Dec 2016.

[90] 3GPP, "Physical layer procedures for shared spectrum channel access," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 37.213, 03 2019, version 15.2.0.

[91] H. Kwon, J. Jeon, A. Bhorkar, Q. Ye, H. Harada, Y. Jiang, L. Liu, S. Nagata, B. L. Ng, T. Novlan, J. Oh, and W. Yi, "Licensed-Assisted Access to Unlicensed Spectrum in LTE Release 13," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 201–207, February 2017.

[92] B. L. Ng, H. Si, A. Papasakellariou, and J. C. Zhang, "Unified access in licensed and unlicensed bands in LTE-A Pro and 5G," *APSIPA Transactions on Signal and Information Processing*, vol. 6, p. e6, 2017.

[93] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[94] P. Gawłowicz and A. Zubow, "ns-3 meets OpenAI Gym: The Playground for Machine Learning in Networking Research," in *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, November 2019.

[95] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 1587–1596.

Acknowledgements

I want to convey my profound gratitude to the people who heartily supported me during my doctoral studies. As my advisor, Professor Hyoil Kim is definitely my biggest supporter. He has advised me to grow into a good researcher. Also, thanks to his passionate supervision, I can engage in research enthusiastically despite obstacles and produce good research results. Again, I thank Professor Hyoil Kim for his dedicated guidance. In addition, I would like to thank the dissertation committee–Professor Kyunghan Lee, Professor Jin Ho Chung, Professor Hyun Jong Yang, and Professor Youngbin Im. I am also appreciative of the WMNL members, Kyubo Shin, Jeong Tak Kim, Seokwoo Choi, and Suhwan Jung. Working together, we helped each other a lot to attain the best research results. I especially want to say thank you to Kyubo Shin, who has been with me from beginning to end.

Lastly, I would like to say thank you to my parents for helping me study for such a long time. Although it wasn't a short time, they helped me to concentrate on my studies. I wouldn't have finished my degree without my parents' sacrifice.