Reinforcement Learning-based Methods for Wireless Access Optimization and Multi-Interface Connectivity

by

Dinh Thi Ha Ly

Dissertation

submitted to the Department of Informatics

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

The Graduate University for Advanced Studies, SOKENDAI
September 2022

Committee

Advisor	Dr. Megumi Kaneko Associate Professor of National Institute of Informatics/SOKENDAI, Japan
Subadvisor	Dr. Yusheng Ji Professor of National Institute of Informatics/SOKENDAI, Japan
Subadvisor	Dr. Kensuke Fukuda Associate Professor of National Institute of Informatics/SOKENDAI, Japan
Examiner	Dr. Shunji Abe Associate Professor of National Institute of Informatics/SOKENDAI, Japan
Examiner	Dr. Takashi Kurimoto Associate Professor of National Institute of Informatics/SOKENDAI, Japan
Examiner	Dr. Lila Boukhatem Associate Professor of University of Paris Saclay, France

Abstract

Recently, 5G wireless networks introduced three new use cases: Enhanced Mobile Broadband (eMBB) for high data rate transmissions, Ultra-Reliable Low Latency Communications (URLLC) enabling high reliability and low latency of connections, and Massive Machine Type Communications (mMTC) for supporting low data rate communications of a massive number of devices. Given the forecasted exponential growth of mobile data traffic and the unprecedented diversification of applications with the spread of the Internet of Things (IoT) applications, Beyond 5G (B5G) and 6G wireless networks will be facing more daunting challenges of Quality of Service (QoS) provision as compared to 5G. Towards this end, future networks are expected to leverage these promising directions: the joint exploitation of a wide range of spectrum from Sub-6GHz to mmWaves for multi-interface connectivity, AI-enabled network entities and energy efficient Deep Learning (DL).

In such a context, this thesis investigates the fundamental issues of wireless access design, namely user association and interference management at user side, and develops new radio resource allocation optimization methods at Access Point (AP) side to enhance global network performances while satisfying individual user QoS constraints. Unlike most of existing studies, we focus on the situation where both user devices and APs are equipped by multiple wireless interfaces, and by DL capabilities. For that, wireless access optimization methods to support multiple applications/interfaces simultaneously with heterogeneous types of QoS and levels, requested by each user device, are investigated. Namely, depending on the QoS requirements and the state of the dynamically varying wireless environment, each application should be served on the most suitable wireless interface at a given time, to offer the utmost user satisfaction to the maximum number of users over the whole network.

Namely, we first consider the problem of user-to-multiple APs association, where a user requesting several applications with different QoS constraints can be served by multiple APs simultaneously, in current Sub-6GHz wireless system. We propose two distributed user-to-multiple APs association methods that leverage Reinforcement Learning (RL), namely Q-Learning (QL) at each user device, enabling each user to optimize its own association decision while aiming at global network optimization. Then, to cope with large-scale networks, we extend this initial QL-based association method by making use of Deep RL (DRL) tools such as DQN and Double DQN (DDQN) based on Deep Neural Networks (DNN). Based on that, in the Sub-6GHz/mmWave integrated networks envisioned for B5G/6G, we handle the issue of joint user-to-multiple APs association and beamforming by proposing a scheme where DNN-enabled user devices optimize their requests (APs,interfaces), while APs perform a greedy-based beamforming to select their best sets of users. The goal is to maximize the system throughput while satisfying the users' QoS requirements and APs' load constraints.

Running such DL functionalities generally requires tremendous energy consumption, which may be prohibitive for battery-limited user devices. Indeed, a large amount of energy is consumed not only for DNN computations using massive data, but also to access, read and write data in the device memory. As Energy Efficiency (EE) will become one of the major Key Performance Indicators (KPI) in B5G/6G system, we also investigate the EE issue of DQN-based method at the user device. In particular, unlike existing works, we conduct a comprehensive analysis of the energy consumption for both computation and data access by DNN. Based on that, to obtain higher EE and better cope with dynamic environments, we enhance our proposed DQN-based user association and beamforming scheme by proposing an adaptive ε-greedy strategy which enables the user to explore whenever notable changes of its surrounding environment are detected. Moreover, to further improve the network performance, we design a beamforming method based on Branch-and-Bound algorithm at AP side, taking into account the features of mmWave bands. The trade-off between achievable network performances and energy costs at user side is then investigated.

Finally, since realizing extreme reliability is another of the major milestones paving the way towards B5G, we also consider the issue of reliability enhancement for mMTC use case under Sub6GHz/mmWave integrated systems. For that, we design a method based on the Risk-Averse Averaged Q-Learning (RAQL) framework, whereby each AP

avoids to transmit on interfaces with high risks of violating devices' Packet Loss Rate (PLR) targets, based on limited feedback from their associated devices.

We assess our proposed methods through numerical evaluations over various network settings. These results show that the proposed approach enables all users to associate to multiple APs/interfaces distributively and efficiently, while satisfying their heterogeneous QoS requirements and enhancing the long-term global sum-rate. Moreover, the proposed algorithms are also shown to outperform benchmark methods, both in terms of global sum-rate and application outage probabilities. In particular, the proposed methods enable to cope with dynamic environments and to strike a balanced trade-off between network sum-rate, QoS satisfaction of diverse applications, as well as user energy consumption. In case of reliability enhancement, the proposed RAQL-based method can significantly improve network performance by increasing the global successful packet delivery rate while reducing individual PLRs as compared to baseline algorithms.

Acknowledgements

This has been a long journey full of new things, which would not have been possible without various supports and guidance from a lot of people.

First and foremost, words cannot express my sincere gratitude to my supervisor, Professor Megumi Kaneko, for her valuable advice and especially, her patience. It is truly a blessing for me to have this chance of working with her and receiving her guides. During five past years, I have studied many precious things from her, not only in the academic field, but also in the working manner, which are completely useful to my work in the future. In deed, I would also like to say a sincere apology to her for being not able to fulfill all her requirements and sometimes making her disappointed. I will try harder to not let her expectations down.

I would also like to thank my committee members, Professor Yusheng Ji, Professor Kensuke Fukuda, Professor Shunji Abe, Professor Takashi Kurimoto in National Institute of Informatics (NII) and Professor Lila Boukhatem at Paris-Saclay University, for their valuable feedback and comments, which helps me improve my thesis a lot.

I would also like to deeply thank to Professor Huynh Thi Thanh Binh at Hanoi University of Science and Technology (HUST) who guided me very first steps on the research path. Thanks to her, I got a chance of visiting and applying for a PhD course at NII/SOKENDAI where I have been pursing my doctoral degree. Her encouragements have helped me stay firmly on the path I chose.

I would also like to greatly thank to Dr. Nguyen Phi Le who not only supported me, but also motivated me a lot to complete my PhD course. She is really a great example to follow, which brought me through all my tough times on this journey.

Many thanks send to all my Vietnamese friends at NII, especially Ha Dao, An Le, Vuong Hong, Dr. Nguyen Hong Huy, Dr. Truong Thao Nguyen and his family, Ms.

Quynh Anh and Min-kun, who have been always with me to share with me all things in study and life. This is my first time to be far from my family for such a long time and distance. It was really lucky to have them here, making me feel like a family and relieving my homesickness.

I would also like to acknowledge all NII and SOKENDAI's staffs for all their supports to handle all administrative procedures and to give us a good research environment.

I owe a sweet thank to my mother, Hoang Thi Thanh Lam, and my older sister, Dinh Thi Ha My, who have always been a solid mental support, helping me to never give up and to overcome any challenges during my life.

Last but not least, a special thank to Tran Bao Loc, my boyfriend and my soul mate also, who has always been there patiently with his endless love for such a long time.

List of Publications

The content of this thesis is mostly based on following papers and patent applications that were published or submitted during my PhD study.

Journal papers

1. Thi Ha Ly Dinh, Megumi Kaneko, Kaito Fujii, "Device Selection and Beamforming Optimization in Large-Scale mmWave IoT Networks", *accepted in IEEE Internet of Things Journal, July 2022*,

2. Thi Ha Ly Dinh, Megumi Kaneko, Keisuke Wakao, Kenichi Kawamura, Takatsune Moriyama, Hirantha Abeysekera, Yasushi Takatori, "Distributed User-to-Multiple Access Points Association through Deep Learning for Beyond 5G", *Computer Networks*, vol. 197, no. 108258, 12 pages, 2021.

Conference papers

1. Thi Ha Ly Dinh, Megumi Kaneko, Kenichi Kawamura,Takatsune Moriyama, Yasushi Takatori, "Improving Reliability by Risk-Averse Reinforcement Learning over Sub6GHz/mmWave Integrated Networks", in *IEEE International Conference on Communications (IEEE ICC)*, 6 pages, May 2022 (Top conference),

2. Thi Ha Ly Dinh, Megumi Kaneko, Keisuke Wakao, Kenichi Kawamura, Takatsune Moriyama, Yasushi Takatori, "Towards an Energy-Efficient DQN-based User Association in Sub6GHz/mmWave Integrated Networks", in *the 17th International*

Conference on Mobility, Sensing and Networking (IEEE MSN), 7 pages, December 2021 (Invited paper),

3. Thi Ha Ly Dinh, Megumi Kaneko, Keisuke Wakao, Kenichi Kawamura, Takatsune Moriyama, Hirantha Abeysekera and Yasushi Takatori, "Deep Reinforcement Learning-based User Association in Sub6GHz/mmWave Integrated Networks", in *IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 7 pages, January 2021,

4. Thi Ha Ly Dinh, Megumi Kaneko, Keisuke Wakao, Hirantha Abeysekera and Yasushi Takatori, "Reinforcement Learning-aided Distributed User-to-Access Points Association in Interfering Networks", in *IEEE Global Communications Conference (GLOBECOM)*, 6 pages, December 2019 (Top conference).

Technical reports

1. Thi Ha Ly Dinh, Megumi Kaneko, "Optimized Multi-Connectivity and Resource Utilization for High Reliability Wireless Communications (FY2021)", *NTT Technical Report*, pp. 1-66, March 2022,

2. Thi Ha Ly Dinh, Megumi Kaneko, "Machine Learning-based Radio Environment Prediction Methods for Enabling Distributed Wireless Systems", *NTT Technical Report*, pp. 1-60, March 2021,

3. Thi Ha Ly Dinh, Megumi Kaneko, "Machine Learning-based Radio Environment Prediction Methods for Enabling Distributed Wireless Systems", *NTT Technical Report*, pp. 1-50, March 2020.

Patent Applications

1. 5211069JP1: "Communication Apparatus, Communication System and Communication Method", K. Kawamura, T. Nakahira, D. Murayama, T. Moriyama, M. Kaneko, T.H.L. Dinh, December 2021,

2-5. 5201447JP1, 5201446JP1, 5201445JP1, 5201444JP1: “Wireless Communication Method, Wireless Terminal and Program for Wireless Terminal”, K. Wakao, K. Kawamura, T. Moriyama, M. Kaneko, T.H.L. Dinh, Jun. 2021,

6. 2020-034683: “Method for Wireless Communication Control, Wireless Communication System, Wireless Terminal, and Wireless Communication Program”, K. Wakao, K. Kawamura, T. Moriyama, H. Abeysekera, Y. Takatori, M. Kaneko, T.H.L. Dinh, March 2020,

7. 2020-034682: “Wireless Communication Method, Wireless Communication System, Base Station Architecture, Wireless Terminal, and Wireless Communication Program”, K. Wakao, K. Kawamura, T. Moriyama, H. Abeysekera, Y. Takatori, M. Kaneko, T.H.L. Dinh, March 2020,

8. 2019-033386: “Apparatus and Methods for Wireless Communications”, K. Wakao, H. Abeysekera, Y. Takatori, M. Kaneko, T.H.L. Dinh, March 2019,

9. 2019-033385: “Wireless Communication System, Wireless Terminal, Centralized Controller and Wireless Communication Methods”, K. Wakao, H. Abeysekera, Y. Takatori, M. Kaneko, T.H.L. Dinh, March 2019.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background and Research Motivation

The fifth generation of wireless network, 5G, enabled to provide higher data rate, lower latency and higher capacity than its preceding 4G Long-Term Evolution (LTE) network by introducing three use cases with different characteristics: Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC) and Massive Machine Type Communications (mMTC) [3]. Namely, eMBB enables stable connections with very high data rates as required by Virtual Reality (VR), while URLLC provides low latency and high reliability communications, but at a low data rate transmission. Requiring typically low data rates and low reliability level of Packet Error Rate (PER) (10^{-1}), mMTC supports a massive number of Internet of Things (IoT) devices.

While these initial 5G networks and services were launched in 2020 in several countries, academia and industries worldwide are currently advancing research on future communication systems, namely Beyond 5G (B5G) and 6G. Compared

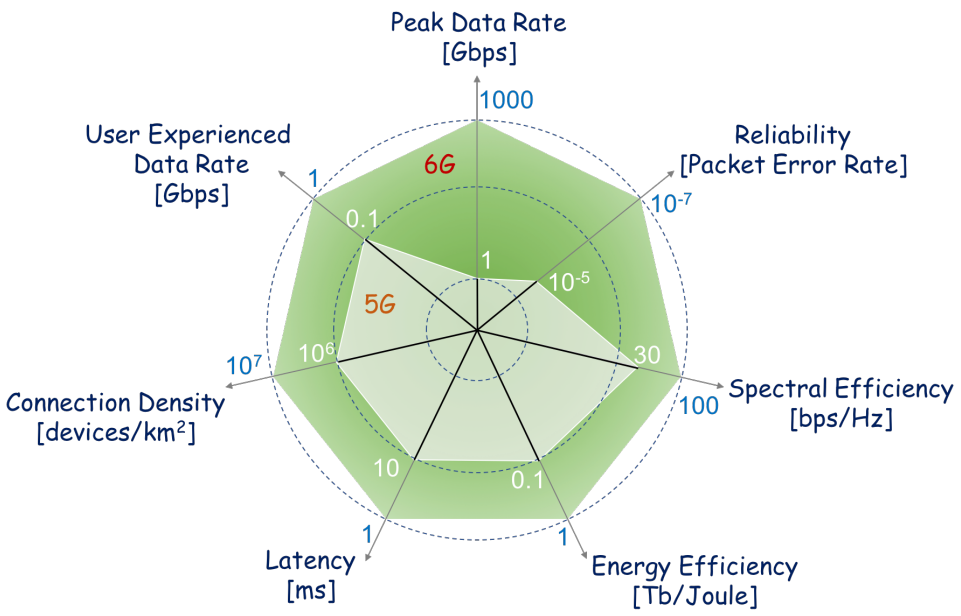


Figure 1.1: Multiple requirements for Beyond 5G/6G networks described in [1]

to 5G, B5G networks are expected to support more extreme amounts of mobile data traffic, owing to the ever increasing number of mobile subscribers across the world. This tendency is accelerated by the spread of IoT and Internet of Everything (IoE) services [1, 4], e.g., smart cities, factory automation, connected vehicles, digital health, etc.. To this end, such diversification of IoT/IoE applications will also require various Quality of Service (QoS) constraints as defined in 5G, but with more stringent levels encompassing ultra-high data rate, ultra-low latency, extreme reliability and ultra-massive connectivity. In particular, these diverse and stringent QoS requirements should not only scale up each of the 5G performance objectives (e.g., thousand-fold increase of data rates, ten-fold increase of connected devices per km^2, ten-fold decrease of latency) but should be also achieved simultaneously [1] as shown in Fig. 1.1. As a result, the three aforementioned 5G use cases would give rise to new use cases, for instance, eMBB high data rate services along with very low latency, or massive connectivity with extreme reliability levels that could go down to 10^{-9}.

In addition, energy efficiency is considered as one of the new major Key Performance Indicators (KPI) for B5G and 6G. As indicated in [5], the energy consumption of information and communication technology (ICT) can exceed 50% of the global

electricity, contributing to 23% of the global greenhouse gas emissions by 2030, most of which is accounted for wireless networks [6, 7]. Namely, it is estimated that 1 GWh of energy is consumed for 1 Petabyte of mobile data communication, which will entail an extremely high power consumption, given the 2 Zettabytes of expected data annually generated by the massive amounts of future IoT devices. Along with the growing awareness of the global warming situation, energy efficiency should be become the prime goal wireless system optimization, while enhancing the other network performances.

To meet such technical challenges, future wireless networks are envisioned to build upon the following major axes: 1) the joint exploitation of a wide range of frequencies from Sub-6GHz to millimeter waves (mmWave) and Terahertz towards 6G, 2) Artificial Intelligence (AI) and Machine Learning (ML)-based technologies at all network levels ranging from core cloud to edge devices, and 3) energy-saving techniques for highly energy-efficient next-generation systems as provisioned by B5G/6G networks. Such prospects come from the following reasons.

Firstly, the conventional frequency bands for mobile communication systems, located in the Sub-6GHz region, will soon be unable to cope with the aforementioned stringent requirements of B5G and 6G applications, due to severe spectrum scarcity. High-frequency mmWave utilization is hence essential due to their high capacity and massive availability. However, these high-frequency bands suffer from high path loss and signal blockage sensitivity [8]. Thus, the joint exploitation of Sub-6GHz and mmWave bands is considered as a potential solution by jointly achieving high system capacity (of mmWave) and robustness (of Sub-6GHz) to wireless channel impairments [9]. Moreover, such a wide range of spectrum allows each wireless entity to be equipped with multiple wireless interfaces for the joint usage of B5G, Wireless Local Area Network (WLAN), Wireless Body Area Network (WBAN), IoT protocols, etc., thereby providing multi-interface connectivity [10].

Secondly, due to the uncertain and highly dynamic and interfering wireless environments of the dense and large-scale future networks, the integration of AI and ML-based techniques at all network levels becomes inevitable. So far, many research works have developed AI-based technologies, in particular, Deep Learning (DL) methods to improve the performance of more and more complex networks. However, AI capabilities have been mostly assumed at the cloud core or edge cloud

as in Mobile Edge Computing (MEC). As we move towards 6G, user devices should be equipped with such AI capabilities, which is necessary to improve the network performance comprehensively for fulfilling the stringent requirements of B5G/6G networks. Nevertheless, performing these AI functionalities comes at the cost of tremendous energy requirements, as current DL techniques based on Deep Neural Networks (DNN) consume a large amount of energy, not only for computing a huge amount of data, but also to transfer and access such data in the memory [2]. This is even more crucial for battery-limited user devices, which requires novel energy-saving techniques for exploiting AI-enabled communications.

In such a context, this thesis aims at investigating the fundamental issues of wireless access design, namely user association and interference management, and developing new radio resource allocation optimization methods to enhance global network performances while satisfying individual user QoS constraints. Namely, we focus on the B5G use cases whereby each user device and AP are equipped by multiple wireless interfaces, and by deep learning functionalities. We investigate wireless access optimization methods for supporting multiple applications simultaneously with heterogeneous types of QoS and levels, requested by each user device. That is, depending on the QoS requirements and the state of the dynamically varying wireless environment, each application should be served on the most suitable wireless interface at a given time, to offer the utmost user satisfaction to the maximum number of users over the whole network.

In particular, we investigate the three following issues, then for each, the wireless access optimization methods are proposed. Firstly, we consider the issue of joint distributed user-to-AP association at user devices and optimization of user selection and beamforming at APs, where DNN-enabled user devices optimize their best sets of APs to request at any time, while based on these requests, APs optimize user selection and beamforming. The goal is to maximize the system throughput while the users' QoS requirements and APs' load constraints are satisfied. Then, given that user association selection is performed by DNN-enabled user devices, the energy consumption at these devices is investigated. Finally, we consider the issue of reliability enhancement for mMTC communications which will become paramount for B5G as mentioned above. Namely, APs optimize their interface selection and packet transmission through their AI functionalities to improve the reliability of the whole system, while satisfying the

PER requirement of each user device, given a stringent delay constraint.

1.2 Thesis contributions

Before giving the details of our thesis contributions, we provide the global view of the positioning of our work within the related literature. In Fig. 1.2, we illustrate the global view of our contributions concerning the user association and energy efficiency issues (Chapters 3, 4 and 5), while Fig. 1.3 illustrates our contributions related to reliability enhancement (Chapter 6). Firstly, as shown in Fig. 1.2, there are two major approaches for handling the user association problem, which are centralized or distributed approaches. In this thesis, we have developed a distributed approach which combines a multiple-APs selection phase at the user device side, and based on these requests, the user selection and beamforming optimization takes place at each AP, where all APs interfere mutually. In particular, we investigate ML-based methods, i.e., Reinforcement Learning (RL) and DL that are envisioned as key enabling technologies for future wireless networks. In addition, while existing DL methods based on DNN either ignore or consider only part of its energy consumption, we take into account the entire DNN energy consumption in the optimized design of our proposed user-to-multiple APs association method.

Next, Fig. 1.3 illustrates our contributions related to reliability enhancement for mMTC types of applications. So far, the reliability of URLLC and mMTC types of services in 5G was improved by means of packet transmission duplication, splitting, and redundant coding. In most cases, these methods also relied on the knowledge of perfect instantaneous Channel State Information (CSI) and/or the knowledge of channel statistics between APs and users, which may not be available in reality, especially in mMTC use case which involves simplistic IoT devices with scarce computation and battery capabilities. Therefore, the target in this thesis is to improve reliability for mMTC use case where only perfect average CSI knowledge could be leveraged, by increasing the number of IoT devices whose reliability QoS (namely, required packet loss rate (PLR) level) is satisfied.

In details, our main contributions in this thesis are listed as follows and presented in Fig. 1.4.

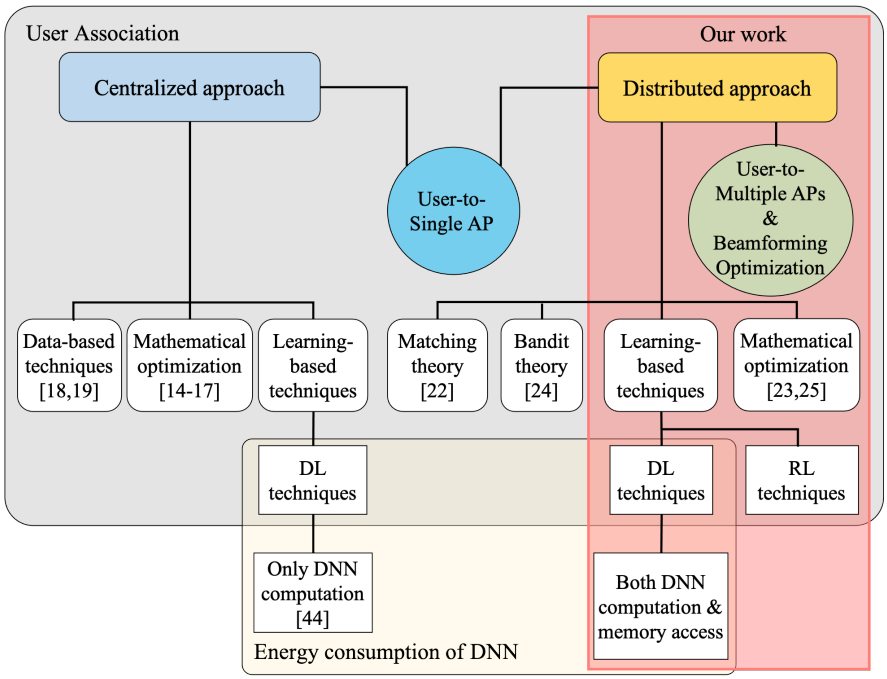


Figure 1.2: Positioning of our work within the literature, contributions related to ML-based user association and energy efficiency improvement (Chapters 3, 4 and 5)

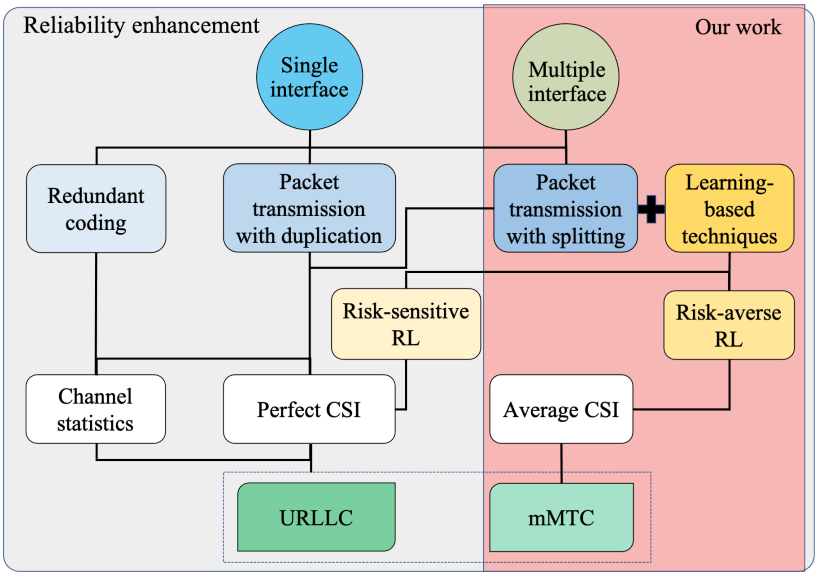


Figure 1.3: Positioning of our work within the literature, contributions related to reliability enhancement (Chapter 6) for mMTC use case, with partial CSI knowledge

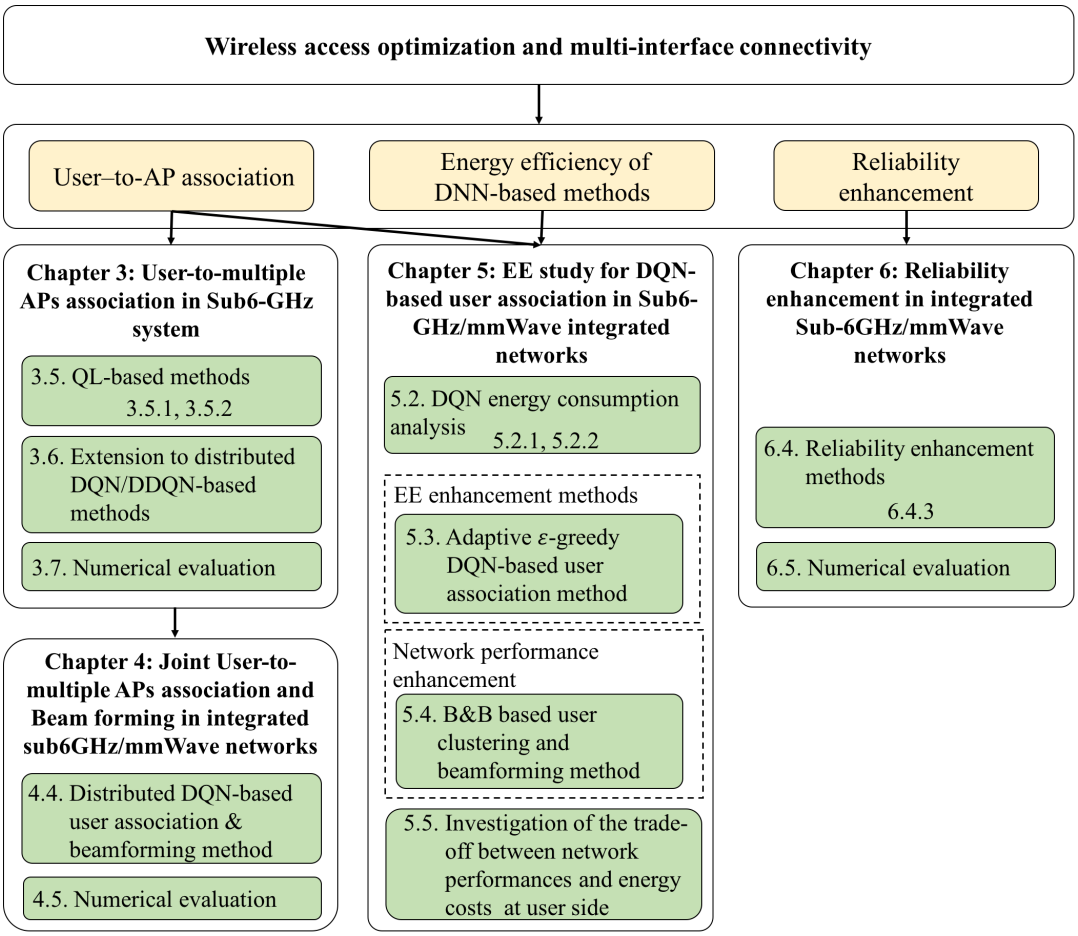


Figure 1.4: Thesis structure and contributions

1. To the best of our knowledge, this is one of the first research enabling user-to-*multiple* APs association where a user requiring several applications with different QoS requirements can be served by multiple APs and interfaces, instead of by only one AP as conventional association methods, simultaneously and without global Channel State Information (CSI). As a preliminary study, we investigate this issue in a wireless system which only operates on Sub-6GHz band and propose two *distributed* methods that leverage Reinforcement Learning (RL), namely Q-Learning (QL) at each user device. The proposed QL-based methods enable each user to optimize its own association decision while aiming at global network optimization. Then, to cope with large-scale networks, we extend this initial QL-based association method by making use of Deep RL (DRL) tools such as Deep Q-Network (DQN) and Double DQN (DDQN).

2. In Sub-6GHz/mmWave integrated systems envisioned for B5G/6G, the user-to-multiple APs association issue becomes more challenging as compared to single band networks, as the combination of users, APs and interfaces should be jointly optimized. In addition, we also take into account the features of high path loss and signal blockage sensitivity of mmWave band. For that, we propose *distributed* DQN-based algorithms for joint user-to-multiple APs association and beamforming in Sub-6GHz/mmWave integrated networks. Namely, at user side, each user device optimizes its own association requests (AP, interface) for each application through its own DQN. Then based on these requests, the beamforming and user selection are optimized at the AP side by our proposed greedy-like method with low complexity without incurring high costs in terms of signaling and CSI overheads.

3. Regarding the energy aspect of the future systems, we make a comprehensive evaluation of the energy efficiency (EE) metric at the user device for the proposed DQN-based methods of user-to-multiple APs association in the Sub-6GHz/mmWave integrated networks. Unlike existing works, the overall energy consumption at the user side is analyzed, including the energy required for operating the DQN, namely for DQN computation as well as memory data access. Based on that, to improve the energy efficiency and to better cope with the dynamics of the mobile environment, we enhance the proposed DQN-based user-

to-multiple APs association and beamforming methods by introducing an adaptive ε-greedy DQN policy at the user side for encouraging the online exploration of new potential APs and interfaces whenever a change of environment is detected. In particular, the beamforming implementation at the AP side is enhanced by the proposed method based on Branch-and-Bound algorithm.

4. For enhancing the reliability of future wireless systems, we investigate the issue of interface selection and packet scheduling at AP side, where the goal is to maximize the global successful packet delivery rate under device Packet Loss Rate (PLR) constraint in Sub-6GHz/mmWave networks. We design a method based on the Risk-Averse Averaged Q-Learning (RAQL) framework [11] for handling this issue, whereby each AP avoids to transmit on interfaces with high risks of violating devices' PLR targets, based on limited feedback from their associated devices.

5. We assess all proposed methods through numerical evaluations over various network settings. These results not only demonstrate the necessity of user-to-multiple APs/interfaces association in the context of B5G systems, but also show that the proposed approach enables all users to associate to multiple APs/interfaces distributively and efficiently, while satisfying their heterogeneous QoS requirements and enhancing the network performance. Moreover, the proposed algorithms are also shown to outperform benchmark methods, both in terms of global sum-rate, application outage probabilities and user fairness. In particular, the proposed methods enable to cope with dynamic environments and to strike a balanced trade-off between network sum-rate, QoS satisfaction of diverse applications, as well as user energy consumption. Concerning reliability improvement, the proposed methods based on RAQL can significantly improve network performance by increasing the global successful packet delivery rate while reducing individual PLRs as compared to baseline algorithms.

1.3 Thesis organization

The remaining chapters of this thesis are organized as follows.

In **Chapter** 2, a comprehensive survey of the state-of-the-art for wireless access optimization is presented.

In **Chapter** 3, the problem of user-to-multiple APs association is first investigated in current Sub-6GHz networks. An overview of studies on this issue in the literature is given. We then formulate the user-to-multiple APs association with various constraints mathematically. We first propose two distributed methods based on QL to handle the formulated problem. Then, we extend these proposed methods by using DQN and DDQN frameworks to further enhance the performances and to cope with the large-scale networks. All proposed methods are evaluated and discussed on several network settings and compared to the corresponding baseline algorithms.

Chapter 4 continues to study on the issue of user-to-multiple APs association but jointly with beamforming in Sub-6GHz/mmWave integrated networks. A DQN-based method of joint user-to-multiple APs association and beamforming is proposed and evaluated through numerical simulations.

Chapter 5 conducts a comprehensive analysis of the energy consumption for operating the DQN functionalities at the user side. Based on that, we enhance the proposed DQN-based algorithm in Chapter 4 for not only improving energy efficiency, but also better coping with the dynamic environment. Then, the trade-off between achievable network performances and energy costs at the user side of the proposed algorithm is investigated and evaluated through numerical simulations.

In **Chapter** 6, the issue of reliability enhancement for mMTC use case is investigated. This issue is formulated as an optimization problem of interface selection and packet scheduling under users' PLR constraints. The algorithms based on RAQL are presented in detail and its effectiveness are assessed through numerical results.

Chapter 7 summarizes the main points of the thesis and provides the research direction for future work.

2

Related Work

2.1 User-to-AP association methods

The problem of user-to-AP association has been identified as one of the key radio resource management issues governing the performance of future dense wireless networks [12]. Most existing user-to-AP association schemes allow each user to associate with the AP that provides the strongest received signal strength (RSSI) [13], resulting into intense congestion at this AP in hot spot scenarios, and hence high outage probability for these users, which requires more efficient user association methods for addressing not only this drawback but also other requirements of future wireless networks. So far, there have been many works handling this issue, which can be categorized into centralized and distributed approaches.

Centralized approaches

Reference [14] proposed a gradient descent-based user association method for maximizing the network utility while balancing load among APs. In [15], the association solution based on Nash bargaining was adopted for guaranteeing their fairness and load balance while the user minimum rate constraints are satisfied. In [16], users are associated to appropriate small cells for minimizing the latency, energy consumption and interference of network through a search algorithm based on Pareto optimality. Concerning the spectrum and energy efficiency, [17] addressed the joint user association and resource allocation through a framework composed of nonlinear fractional programming and dual decomposition techniques. However, these methods require global and perfect Channel State Information (CSI) knowledge of all the links at the centralized scheduler. Moreover, they incur prohibitively high costs in terms of computational complexity, power and signaling overhead, making them unsuitable for B5G networks.

More recently, a centralized data-driven approach was proposed in [18, 19], where a robust optimal user association map, pre-calculated at the BS, is used to determine the actual served users in real-time. However, this method is limited to the specific area whose optimal association data is available before hand, making it difficult to generalize to other regions.

Inspired by the success of the ML-based methods from other research fields, learning-based user association methods have been emerged. For instance, a method based on an actor-critic DL for efficient joint user association and bandwidth allocation in a dense downlink mobile network was proposed in [20]. Making use of a pre-trained DNN, reference [21] designed a centralized user association scheme that can provide a real-time solution through MEC. However, this scheme requires a long training phase, for gathering large amounts of historical data of the global network environment.

In general, the centralized methods requires a large number of data including perfect global CSI of the whole network, historical knowledge of the network area with wireless environment, which is not always available and consumes a lot of time for collecting and processing before being usable.

Distributed approaches

To overcome the drawbacks of these centralized methods, distributed approaches with partial knowledge of the network environment have been considered. For instance, [22] modelled the competitive behavior among users and APs as a dynamic matching game and then presented a distributed matching algorithm for optimizing user-to-AP association. In heterogeneous networks, [23] proposed a distributed strategy of energy-efficient and fair user association based on Lagrange dual method, in order to maximize a global network utility. The authors of [24] developed a distributed scheme where each user independently selects one AP to associate, given a success probability metric, based on bandit theory without any prior information at users. In [25], a semi-distributed method based on Alternating Direction Method of Multipliers (ADMM) was introduced to handle the joint issue of user association and user scheduling for load balancing in heterogeneous networks. However, these methods require significant time to converge as the network size grows, making them difficult to apply to large-scale networks and delay-stringent applications.

This is why more recently, distributed learning-methods are interested. Reference [26] designed a method using Deep Deterministic Policy Gradient in the context of online video streaming services with MEC. In [27], the authors provided an online DRL method using multiple DNNs to generate solutions for the training data set of the user association problem in heterogeneous systems. However, this method still requires the channel information of the whole network at the input of the DNNs. References [28–31] also designed learning-based methods for distributed user association, where it is generally assumed that each user has knowledge of the QoS satisfaction status of all other users in the network, which is unrealistic.

One major observation is that, all these above methods do not allow each user to be associated to multiple APs simultaneously, which is a fundamental limitation that hinders the joint satisfaction of heterogeneous types of applications and services, as will be required in future networks. To overcome this drawback and to achieve much higher user satisfaction, each user device should be able to connect to different radio interfaces across different APs for their various applications. Thus, developing new approaches and solutions tailored to the issue of user-to-*multiple* APs association is crucial.

Moreover, as envisioned, every entity in the network will be equipped with both Sub-6GHz and mmWave interfaces, thereby forming integrated networks. However, in the literature, most studies investigated user-to-AP association issue in either sub-6GHz [30] or mmWave systems [32, 33]. In spite of considering a system of mmWave APs and sub-6GHz APs, references [34, 35] proposed centralized methods to associate each user to one AP, under the ideal assumption of perfect knowledge of blockages, which is not applicable for B5G large-scale networks.

Hence, to the best of our knowledge, so far, no study investigating the issue of user-to-multiple APs association, especially in Sub-6GHz/mmWave integrated networks without any assumption of perfect CSI knowledge at both APs and users, which will be one of the objects of this thesis. In particular, this issue will be handled in a distributed manner at user devices by DL techniques which is envisioned for future wireless networks.

2.2 Energy efficiency of DNN-based access methods

As aforementioned discussion in Chapter 1, the energy efficiency (EE) will become one of the major KPIs in future networks. Meanwhile, AI-enabled network entities, mostly leveraging DL techniques based on DNN, are envisioned as one of supporting factors for enabling B5G/6G. Although DNNs are took advantage in many DL-based methods and provide high performance, it is expensive in energy cost. Therefore, energy efficiency in processing of DNNs is inevitable. To tackle this issue, approaches from both the algorithm design and hardware architecture have been investigated.

Several works of the hardware architecture for EE can be found in [36, 37] which is to handle compressed form of DNN, or in [38] for introducing a large MAC array which allow to reuse some weights and activation values instead of loading each time of using.

In the algorithm aspect, one main focus is to reduce the number of weights and Multiples and Accumulates (MAC) operations. For instance, [39] proposed a prunning technique to increase the weight sparsity level. However, [40] showed that this methods reduce confidence of DNN predictions, resulting in much lower performance in some tasks. More importantly, this approach only decreases the storing space in memory, but doesn't actually reduces the energy consumption as showed in [41]. Namely, the

DNN model SqueezeNet consumes more energy than AlexNetrequires which has more weights. This is because, as indicated in [2, 42], a large amount of energy is consumed not only for DNN computations using massive data, but also to data movement which includes accessing, reading and writing data in the device memory. Therefore, energy consumption of DNN should not be based only on the number of weights and MACs of DNN computation. However, so far, this crucial aspect had been discarded in most previous works. Although [43] proposed a machine learning-based method to improve EE, the actual energy consumed by DNN was not considered. Recently, [44] proposed an Artificial Neural Network (ANN)-based method to optimize the global energy efficiency of a single-cell system. Despite considering an energy efficient design of the DNN, the energy consumption for data movement was not accounted for, although this was shown in [2, 42] to be dominant as compared to the energy consumption for computation. Furthermore, the DNN of [44] is located at a centralized cloud server, not within user devices.

Therefore, a full study of the overall energy consumption required by DNNs is needed, which will be conducted in this thesis. Moreover, based on this study, we then enhance our proposed DQN-based user-to-multiple AP association method in order to obtain a higher EE for devices and the trade-off between trade-off between achievable network performances and energy costs of user devices will be also investigated.

2.3 Reliability improvement through multi-interface connectivity

As shown in Fig. 1.1, realizing extreme reliability is another of the major milestones paving the way towards B5G, especially for IoT/IoE communications such as autonomous driving, remote medecine.

In the literature, many approaches were proposed to enhance the reliability of wireless communications. One of the main approaches is to exploit multi-connectivity over multiple transmission paths, through, e.g., packet cloning, message splitting, or optimized path selection, among others [45–47]. In [47], a method for coding over multiple interfaces was proposed, whereby the packet splitting weights were optimized based on the statistical models of each interface, in order to improve the

latency-reliability trade-off.

However, these methods assumed perfect CSI knowledge which may not be available in reality. Moreover, multi-interface diversity was not exploited, which makes the current approaches unsuitable for B5G networks with joint exploitation both Sub-6GHz and mmWave. This motivates us to consider the issue of reliability improvement for packet transmission under such networks where every entity is equipped by both Sub-6GHz and mmWave interfaces.

3

Learning-based User-to-Multiple Access Points Association in Sub-6GHz Networks

3.1 Introduction

The issue of user-to-AP association was considered in many works as mentioned in section 2. However, these studies were not able to fulfil the requirements of future networks as each user is restricted to be served by only one AP at a time. In this chapter, we first investigate the issue of user-to-multiple APs association, where a user requiring several applications can be served by several APs simultaneously, in the current Sub-6GHz systems. In particular, unlike previous works [14, 17, 27], we do not make any assumption for global CSI knowledge at both user and AP sides. This problem is formulated as a network sum-rate maximization such that the required QoS constraints for each user and application, and AP load constraints are satisfied.

This issue is non-trivial even in current Sub-6GHz networks due to the dynamics and uncertainties of the wireless environment in the absence of CSI knowledge. We then first propose two QL-based distributed user association methods, where each user is able to learn its best set of APs to be connected to at any time, solely based on local knowledge of its surrounding wireless environment. It is worthwhile noting that the considered user-to-multiple APs association is fundamentally different to Coordinated Multipoint (CoMP)-like approaches, as in our case all APs are uncoordinated and take their allocation and association decisions independently.

Although QL guarantees convergence towards the optimal policy as long as all states and actions are visited often enough, this method is hardly applicable to scenarios with large state/action spaces. To address this essential problem, we hence propose user-to-multiple APs association methods exploiting DRL, and in particular, DQL based on Deep Q-Networks [48]. However, an intrinsic drawback of DQN is the overestimation issue of Q-values which introduces bias in the optimal action selection. As explained in [49], this problem can be efficiently tackled by Double DQN (DDQN) which makes use of different DQNs for Q-value estimation and action selection. Therefore, we also propose a method leveraging the DDQN technique to further enhance the network performance.

Particularly, for each proposed association algorithm, two types of distribution level are developed: the first, termed *Fully Distributed-QL (Fully Distributed-DQN/DDQN)* method, is based on minimal decision feedback from APs towards users, and the second, termed *Partially Distributed-QL (Partially Distributed-DQN/DDQN)* method, further improves the achievable network performance by letting each user acquire additional local information regarding its neighboring user requirements.

3.2 System model

We consider a downlink network composed of a set $\mathcal{B}$ of fixed APs and a set $\mathcal{K}$ of randomly located static users[1], as depicted in Fig. 3.1. All APs operate on the same bandwidth for sake of spectrum efficiency, but interfere among each other as in a realistic environment. In each scheduling frame t, each user $k \in \mathcal{K}$ requests a set of

[1]We assume user positions to be fixed under low user mobility scenarios.

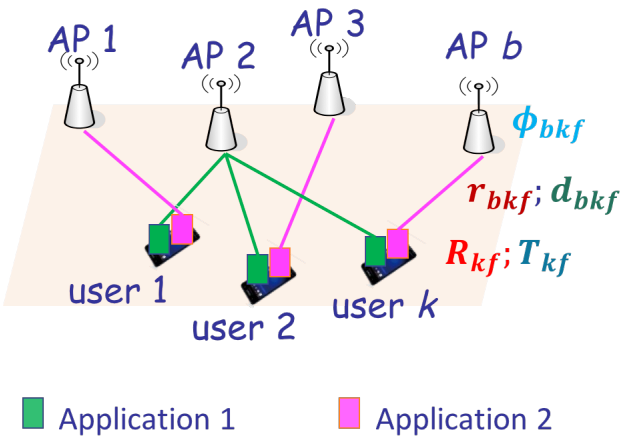


Figure 3.1: User-to-multiple APs association in Sub-6GHz networks

applications $\mathcal{F}_k$, in which each application f requests either a minimum rate constraint R_{kf} or a maximum delay time constraint T_{kf}. For convenience, we denote the set of applications requesting R_{kf} as $\mathcal{F}_{kr}$ and the set of applications requesting T_{kf} as $\mathcal{F}_{fd}$. Hence, $\mathcal{F}_{kr} \cup \mathcal{F}_{fd} = \mathcal{F}_k$ and $\mathcal{F}_{kr} \cap \mathcal{F}_{kd} = \emptyset$.

The achievable data rate $r_{bkf}(t)$ at user k for application f provided by AP b at frame t is computed by

$$r_{bkf}(t) = W \log(1 + \gamma_{bkf}(t)), \tag{3.1}$$

where W is the bandwidth. $\gamma_{bkf}(t)$ denotes the Signal to Interference-plus-Noise Ratio (SINR) at user k with application f for the transmit signal from AP b, which is given as

$$\gamma_{bkf}(t) = \frac{|h_{bk}(t)|^2 p_{bkf}(t)}{\sum\limits_{b' \in \mathcal{B} \setminus b} |h_{b'k}(t)|^2 p_{b'}(t) + W\sigma_n^2}. \tag{3.2}$$

Here $h_{bk}(t) \in \mathbb{C}$ denotes the complex channel coefficient between AP b and user k, including path loss and small-scale fading effects; $p_{bkf}(t) \in \mathbb{R}^+$ is the transmit power from AP b to user k for application f, which is assumed to be known and fixed. The term σ_n^2 denotes the Additive White Gaussian Noise (AWGN) power. Since they use the same bandwidth, all other APs $b' \in \mathcal{B} \setminus \{b\}$ cause interference to user k served by AP b, with full transmit power $p_{b'}$.

Next, the delay time d_{bkf} required for serving application f of user k by AP b at

frame t is given by

$$d_{bkf}(t) = \frac{s_f}{r_{bkf}(t)}, \tag{3.3}$$

where s_f denotes the file size of application f.

All APs are assumed to be able to serve any requested application unless their maximum load limit is violated. As in [28], the load of AP b for serving application f of user k is computed by

$$\phi_{bkf}(t) = \frac{m_{kf}}{\mu_{kf} r_{bkf}(t)}, \tag{3.4}$$

where m_{kf} and $\frac{1}{\mu_{kf}}$ denote the mean arrival rate in number of packets per seconds, and the mean packet size of application f in bits, respectively. Hence, assuming an orthogonal allocation of wireless resources in frequency or time for serving user applications as in [28], AP b will become overloaded if its total load exceeds the normalized value of 1, namely if

$$\Phi_b(t) = \sum_{k \in \mathcal{K}} \sum_{f \in \mathcal{F}_k} x_{bkf}(t) \phi_{bkf}(t) > 1, \tag{3.5}$$

where $x_{bkf}(t)$ is an association variable defined as,

$$x_{bkf}(t) = \begin{cases} 1, & \text{if AP } b \text{ serves application } f \text{ of user } k \text{ at frame } t, \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

3.3 Problem formulation

We consider the long-term global average sum-rate maximization problem, under user QoS constraints in terms of minimum data rate R_{kf} and maximum delay time T_{kf} for each user k, application f, and AP load constraints, which is formulated as in (3.7).

The objective function (3.7) expresses the long-term average sum-rate over all applications, users and APs in the network. Constraint (3.7a) sets the binary nature of association variables $x_{bkf}(t)$ defined in (3.6). Eq. (3.7b) constrains each application requested by each user to be served by a unique AP. The minimum data rate and the maximum delay time for each application are specified by (3.7c) and (3.7d), respectively.

Finally, the load constraint for each AP b is reflected in (3.7e).

$$\max_{x_{bkf}(t)} \quad \mathbf{E}_t \left[\sum_{b\in\mathcal{B}} \sum_{k\in\mathcal{K}} \sum_{f\in\mathcal{F}_k} x_{bkf}(t) r_{bkf}(t) \right], \tag{3.7}$$

$$\text{s.t. } x_{bkf}(t) \in \{0,1\}, \forall b \in \mathcal{B}, k \in \mathcal{K}, f \in \mathcal{F}_k, \tag{3.7a}$$

$$\sum_{b\in\mathcal{B}} x_{bkf}(t) = 1, \quad \forall k \in \mathcal{K}, \forall f \in \mathcal{F}_k, \tag{3.7b}$$

$$\sum_{b\in\mathcal{B}} x_{bkf}(t) r_{bkf}(t) \geq R_{kf}, \quad \forall k \in \mathcal{K}, \forall f \in \mathcal{F}_{kr}, \tag{3.7c}$$

$$\sum_{b\in\mathcal{B}} x_{bkf}(t) d_{bkf}(t) \leq T_{kf}, \quad \forall k \in \mathcal{K}, \forall f \in \mathcal{F}_{kd}, \tag{3.7d}$$

$$\Phi_b(t) = \sum_{k\in\mathcal{K}} \sum_{f\in\mathcal{F}_k} x_{bkf}(t) \phi_{bkf}(t) \leq 1, \quad \forall b \in \mathcal{B}. \tag{3.7e}$$

Problem (3.7) is a combinatorial optimization problem which cannot be solved in polynomial time. This becomes especially intricate in a B5G setting where a large number of users with conflicting QoS constraints and creating high interference levels, should be simultaneously satisfied. Furthermore, distributed association methods based on local network and channel state information, are deemed necessary. To meet these goals, we first propose to make use of reinforcement learning, in particular Q-learning [50]. Then to cope with large-scale networks, we propose to leverage self-learning and self-optimization by exploiting deep reinforcement learning (DRL) at the user side, as explained in the next sections.

3.4 Formulation as a Markov Decision Process (MDP)

We first formulate the considered distributed problem as an MDP. Based on that, the proposed distributed methods based on QL, DQN/DDQN can be devised, whereby each user learns the best set of APs to request in each time frame, so as to satisfy the heterogeneous QoS requirements of each application as explained in next sections.

An MDP is a discrete time stochastic control process defined by four elements (state space, action space, transition probability, reward function). As shown in Fig. 3.2, each user is an agent who takes its decision of requesting APs for its applications. At each scheduling frame t, the user knows its current state s_t, i.e., its current association

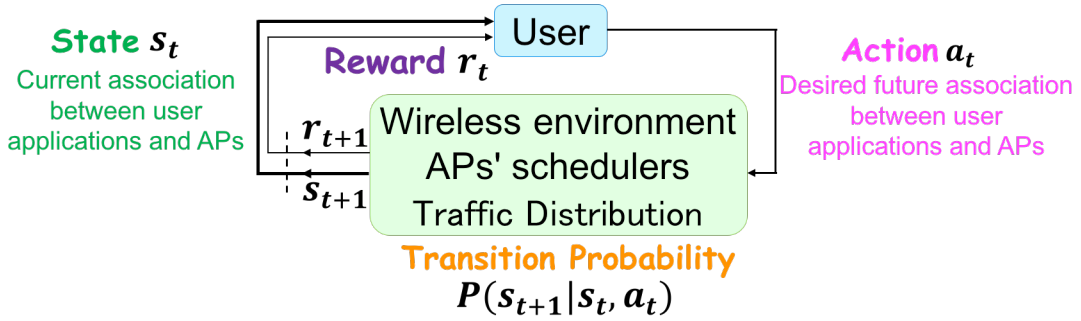


Figure 3.2: An MDP model of user-to-multiple APs association in Sub-6GHz networks

between its applications and APs, and takes action a_t, i.e., requesting for the next frame the same or different APs for each application. The user then moves to a new state s_{t+1} and receives an immediate reward r_t from the environment. One major observation here is that, the transition probability $P(s_{t+1}|a_t, s_t)$ is unknown to the user, since the association decisions for each application on each interface, though based on user requests, are taken by each AP given the current wireless environment and traffic distribution. Therefore, we propose to solve this problem by means of RL and DRL, in particular based on QL and DQN/DDQN due to its efficiency for handling similar issues in wireless systems as shown in [28, 30] and for coping with large state/action spaces.

3.5 Proposed QL-based user-to-multiple APs association methods

To make problem (3.7) tractable by DL, we consider optimizing the long-term average network sum-rate under (3.7b)-(3.7e), but keeping a short-term QoS contraint for (3.7c) and (3.7d). Namely, constraints (3.7c) and (3.7d) should still be satisfied respectively for short-term rates r_{bkf} and delay time d_{bkf} that are impacted by instantaneous channel fading effects. To this end, we propose two distributed QL approaches where each user in the system acts as an agent taking independent decisions (action) given its current state and based on its local knowledge of the environment. We define state and action space of user k as follows.

- **User State:** The state $\mathbf{s}_k(t)$ of agent k in state space $\mathcal{S}_k$ is the current association between required applications f of user k and each AP b at the beginning of

frame t, i.e,

$$\mathbf{s}_k(t) \in \mathcal{S}_k = \left\{ x_{bkf}(t), \forall b \in \mathcal{B}, \forall f \in \mathcal{F}_k \right\}. \tag{3.8}$$

Due to (3.7b) which constrains each application f of user k to be served by one AP, the maximum number of possible states of each agent is $(\mathbf{C}^1_{|\mathcal{B}|})^{|\mathcal{F}_k|} = B^{F_k}$. That is, instead of considering all $\{0, 1\}$ combinations of all APs and all applications, which results into $2^{B \times F_k}$ elements as in previous works, our state space definition allows it to be compactly represented by B^{F_k} elements, thereby reducing the number of rows required in the Q-tables in case of QL-based method and hence memory usage in user terminals.

- **User Action:** Given its current association state $\mathbf{s}_k(t)$ and immediate reward defined in the sequel, user k selects by action $\mathbf{a}_k(t)$ in its action space $\mathcal{A}_k$, its desired future APs to be associated to for frame $t + 1$, under the restriction (3.7b), namely

$$\mathbf{a}_k(t) \in \mathcal{A}_k = \left\{ a_{bkf}(t) \mid \sum_{b \in \mathcal{B}} a_{bkf}(t) = 1, \forall f \in \mathcal{F}_k \right\}. \tag{3.9}$$

Here, $a_{bkf}(t)$ are the binary variables of association requests, defined as

$$a_{bkf}(t) = \begin{cases} 1, & \text{if user } k \text{ requests AP } b \text{ for application } f \in \mathcal{F}_k \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}$$

Similarly, $\mathcal{A}_k$ has a maximum of $(\mathbf{C}^1_{|\mathcal{B}|})^{|\mathcal{F}_k|} = B^{F_k}$ possible actions. Note also that, by definition of this action space, constraint (3.7b) is guaranteed to hold.

- **Q-function:** For each selected action $\mathbf{a}_k(t)$ at state $\mathbf{s}_k(t)$, the corresponding Q-value is given by [50]

$$Q\Big(\mathbf{s}_k(t), \mathbf{a}_k(t)\Big) = Q\Big(\mathbf{s}_k(t), \mathbf{a}_k(t)\Big) + \alpha\Big(\Gamma_k(t) + \beta \max_{\mathbf{a}_k \in \mathcal{A}_k} Q\Big(\mathbf{s}_k(t+1), \mathbf{a}_k\Big) - Q\Big(\mathbf{s}_k(t), \mathbf{a}_k(t)\Big)\Big), \tag{3.11}$$

where $\Gamma_k(t)$ denotes the immediate reward achieved by action $\mathbf{a}_k$ selected by user k at frame t.

Based on the above definitions, we propose our QL-based user association methods described in Algorithm 3.1 and explained as follows.

Algorithm 3.1: Proposed QL-based User Association Algorithm

1 Decay factor λ, greedy policy factor ε, weights w_{1k}, w_{2k};
2 **for** *each user* $k \in \mathcal{K}$ **do**
3 Initialize Q-table Q with zero-values;
4 Random initial state $\mathbf{s}_k$;
5 **for** $t = 1, 2, \ldots, T$ **do**
6 **for** *each user* $k \in \mathcal{K}$ **do**
7 $\varepsilon \leftarrow \varepsilon \times \lambda$;
8 **if** *random number* $p < \varepsilon$ **then** Select action $\mathbf{a}_k$ randomly;
9 **else** Select action $\mathbf{a}_k$ with highest $Q(\mathbf{s}_k, \mathbf{a}_k)$;
10 **for** *each AP* $b \in \mathcal{B}$ **do**
11 Consider requests, select users/applications (if necessary) by greedy method ;
12 Feedback to users, data transmission;
13 **for** *each user* $k \in \mathcal{K}$ **do**
14 Calculate the reward of action $\mathbf{a}_k$ by (3.12) ;
15 Update $Q(\mathbf{s}_k, \mathbf{a}_k)$ by (3.11);
16 Move to the new state $\mathbf{s}_k \leftarrow \mathbf{s}_k^{'}$;

Step 1- At the user side, with probability 1-ε, each user selects action $\mathbf{a}_k(t)$ as in (3.9) with the highest Q-value from its own Q-table for its current state $\mathbf{s}_k(t)$, then sends its request $\mathbf{a}_k(t)$ to the desired AP for each application f in $\mathcal{F}_k$ (Lines 6 to 9).

Step 2- After receiving all user requests, each AP decides to accept these requests or not based on its current load, i.e., if $\Phi_b(t) \leq 1$, AP b will serve all requested applications. Otherwise, AP b drops some applications by a greedy manner, namely, the user/application requests $a_{bkf}(t)$ corresponding to the highest load $\phi_{bkf}(t)$ will be eliminated until the constraint $\Phi_b(t) \leq 1$ is satisfied. This is because the objective of this problem is to maximize the total system throughput while satisfying all user/application QoS requirements. In addition, for a given file size, lower achievable rates entail higher loads at the AP. After this phase, AP b sends its association decision $x_{bkf}(t)$ to its requested user through feedback. (Lines 10 to 12)

Step 3- At the user side, based on the feedback from APs, each user calculates its immediate reward $\Gamma_k(t)$ which is the weighted sum of two terms, c_{1k} and $(c_{2k}^r + c_{2k}^d)$

with corresponding weights w_{1k} and w_{2k}, namely

$$\Gamma_k(t) = w_{1k}c_{1k}(t) + w_{2k}\left(c^{\mathrm{r}}_{2k}(t) + c^{\mathrm{d}}_{2k}(t)\right). \tag{3.12}$$

Here c_{1k} is the reward for user k as its requested application $f \in \mathcal{F}_k$ was actually served by the selected AP, and satisfied the corresponding QoS, given by

$$c_{1k}(t) = \sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kr}} \mathcal{I}\left(x_{bkf}(t) = 1, r_{bkf}(t) \geq R_{kf}\right)a_{bkf}(t)r_{bkf}(t) + \sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kd}} \mathcal{I}\left(x_{bkf}(t) = 1, d_{bkf}(t) \leq T_{kf}\right)a_{bkf}(t)r_{bkf}(t), \tag{3.13}$$

with the indicator function $\mathcal{I}(.)$.

On the contrary, if user k's requested applications' QoS were not satisfied or if they were dropped by APs, user k calculates its penalties, given by c^{r}_{2k} if the application has a minimum rate requirement, i.e., $f \in \mathcal{F}_{kr}$ and by c^{d}_{2k} if the application has a maximum delay requirement, i.e., $f \in \mathcal{F}_{kd}$. The immediate reward depends upon the type of feedback from its requested APs, from which users may get more or less knowledge of their local wireless environment. Thus, we considered two types of AP feedback, based on which two methods are designed, namely, the *Proposed Fully Distributed QL Association* and the *Proposed Partially Distributed QL Association*, described in Sections 3.5.1 and 3.5.2, respectively. Finally, user k updates its own Q-table for the pair $(\mathbf{s}_k(t), \mathbf{a}_k(t))$ by (3.11) and move to its new state (Lines 13 to 16). This process is repeated over frames until convergence of the reward, or the maximum number of frames is reached.

3.5.1 Proposed fully distributed QL-based association

In this algorithm, each user receives the minimal feedback Ω_{bk} including only its desired APs' allocation decision, i.e,

$$\Omega_{bk}(t) = \{x_{bkf}(t) | a_{bkf}(t) = 1 \;\&\; f \in \mathcal{F}_k\}. \tag{3.14}$$

In other words, each user knows only about its own instantaneous channel information

with its serving APs, but has no knowledge about other users' channel states nor requested QoS. Therefore, the penalties c^{r}_{2k}, c^{d}_{2k} of user k are calculated solely using its local channel state information, following (3.15a) if user k is served by AP b but at a rate lower than R_{kf}, and by (3.16a) if user k is served by AP b but with a delay time larger than T_{kf}, respectively. If user k's application f is dropped by its selected AP b, the penalties c^{r}_{2k}, c^{d}_{2k}, which is to emphasize how bad this action is, are computed by (3.15b), (3.16b),

$$c^{r}_{2k}(t) = \begin{cases} -\sum\limits_{b\in\mathcal{B}}\sum\limits_{f\in\mathcal{F}_{kr}} \mathcal{I}\left(x_{bkf}(t)=1, r_{bkf}(t)<R_{kf}\right) a_{bkf}(t)\frac{R_{kf}}{r_{bkf}(t)} & \text{(a)},\\ -\sum\limits_{b\in\mathcal{B}}\sum\limits_{f\in\mathcal{F}_{kr}} \mathcal{I}\left(x_{bkf}(t)=0\right) a_{bkf}(t)\frac{R_{kf}}{r_{bkf}(t)} & \text{(b)}. \end{cases} \tag{3.15}$$

$$c^{d}_{2k}(t) = \begin{cases} -\sum\limits_{b\in\mathcal{B}}\sum\limits_{f\in\mathcal{F}_{kd}} \mathcal{I}\left(x_{bkf}(t)=1, d_{bkf}(t)>T_{kf}\right) a_{bkf}(t)\frac{d_{bkf}(t)}{T_{kf}} & \text{(a)},\\ -\sum\limits_{b\in\mathcal{B}}\sum\limits_{f\in\mathcal{F}_{kd}} \mathcal{I}\left(x_{bkf}(t)=0\right) a_{bkf}(t)\frac{d_{bkf}(t)}{T_{kf}} & \text{(b)}. \end{cases} \tag{3.16}$$

In (3.15) and (3.16), the ratios $\frac{R_{kf}}{r_{bkf}(t)}$ and $\frac{d_{bkf}}{T_{kf}(t)}$ enable to weight the impact of the incurred loss according to the actual rate and delay QoS, but also to the instantaneous channel quality: the higher the required rate R_{kf} (or the lower the required delay time T_{kf}) and the lower the channel quality (expressed by instantaneous rates $r_{bkf}(t)$ or by instantaneous serving time $d_{bkf}(t)$), the larger the penalty.

3.5.2 Proposed partially distributed QL-based association

In this algorithm, the users will receive additional information from the feedback of its desired APs. Namely, in addition to their own instantaneous channel information and requested APs' association decisions, each user will also gain knowledge about its own load relative to that of neighboring competitors who had requested the same APs. Hence, the feedback Ω_{bk} sent to user k, is now given as

$$\Omega_{bk}(t) = \{x_{bkf}(t), \phi_{bkf}(t) | a_{bkf}(t) = 1 \;\&\; \forall f \in \mathcal{F}_k; N^{\mathrm{r}}_b(t), N^{\mathrm{d}}_b(t)\}, \tag{3.17}$$

where $N_b^{\mathrm{r}}(t)$, $N_b^{\mathrm{d}}(t)$ are normalization factors for dropped applications with minimum rate and maximum delay time requirements respectively, given as

$$N_b^{\mathrm{r}}(t) = \sum_{\substack{k' \in \mathcal{K} \\ f' \in \mathcal{F}_{kr}}} \mathcal{I}\left(a_{bk'f'}(t) = 1, x_{bk'f'}(t) = 0\right) \frac{R_{k'f'}}{r_{bk'f'}(t)}. \tag{3.18}$$

$$N_b^{\mathrm{d}}(t) = \sum_{\substack{k' \in \mathcal{K} \\ f' \in \mathcal{F}_{kd}}} \mathcal{I}\left(a_{bk'f'}(t) = 1, x_{bk'f'}(t) = 0\right) \frac{d_{bk'f'}}{T_{k'f'}(t)}. \tag{3.19}$$

Taking advantage of this improved local environment knowledge, we now update the computation of penalty terms c_{2k}^{r}, c_{2k}^{d} as

$$c_{2k}^{\mathrm{r}}(t) = \begin{cases} -\sum\limits_{b \in \mathcal{B}} \sum\limits_{f \in \mathcal{F}_{kr}} \mathcal{I}\left(x_{bkf}(t) = 1, r_{bkf}(t) < R_{kf}\right) a_{bkf}(t) \frac{R_{kf}}{r_{bkf}(t)}, & \text{(a)} \\ -\sum\limits_{b \in \mathcal{B}} \sum\limits_{f \in \mathcal{F}_{kr}} \mathcal{I}\left(x_{bkf}(t) = 0\right) a_{bkf}(t) \phi_{bkf}(t) \frac{\frac{R_{kf}}{r_{bkf}(t)}}{N_b^{\mathrm{r}}(t)}. & \text{(b)} \end{cases} \tag{3.20}$$

$$c_{2k}^{\mathrm{d}}(t) = \begin{cases} -\sum\limits_{b \in \mathcal{B}} \sum\limits_{f \in \mathcal{F}_{kd}} \mathcal{I}\left(x_{bkf}(t) = 1, r_{bkf}(t) < R_{kf}\right) a_{bkf}(t) \frac{d_{bkf}}{T_{kf}(t)}, & \text{(a)} \\ -\sum\limits_{b \in \mathcal{B}} \sum\limits_{f \in \mathcal{F}_{kd}} \mathcal{I}\left(x_{bkf}(t) = 0\right) a_{bkf}(t) \phi_{bkf}(t) \frac{\frac{d_{bkf}}{T_{kf}(t)}}{N_b^{\mathrm{d}}(t)}. & \text{(b)} \end{cases} \tag{3.21}$$

In (3.20b) and (3.21b), the penalty terms in the case of overloaded APs, are improved by weighting each dropped application by its load contribution $\phi_{bkf}(t)$ upon its requested AP b, and also by normalizing the previous weight $\frac{R_{kf}}{r_{bkf}(t)}$, $\frac{d_{bkf}(t)}{T_{kf}}$ by the term $N_b^{\mathrm{r}}(t)$, $N_b^{\mathrm{d}}(t)$ respectively, which incorporates the rate as well as delay time requirements and instantaneous channel qualities of all other dropped users. These new definitions enable to set adequate penalties to each user, relatively to each other's loads, required rates, required serving time, and instantaneous channels, yet with only local information.

3.6 Extension to DQN/DDQN-based user-to-multiple APs association methods

We extend the proposed QL-based algorithms in section 3.5 by making use of DQN and DDQN techniques, which enables these methods to be applicable in large-scale

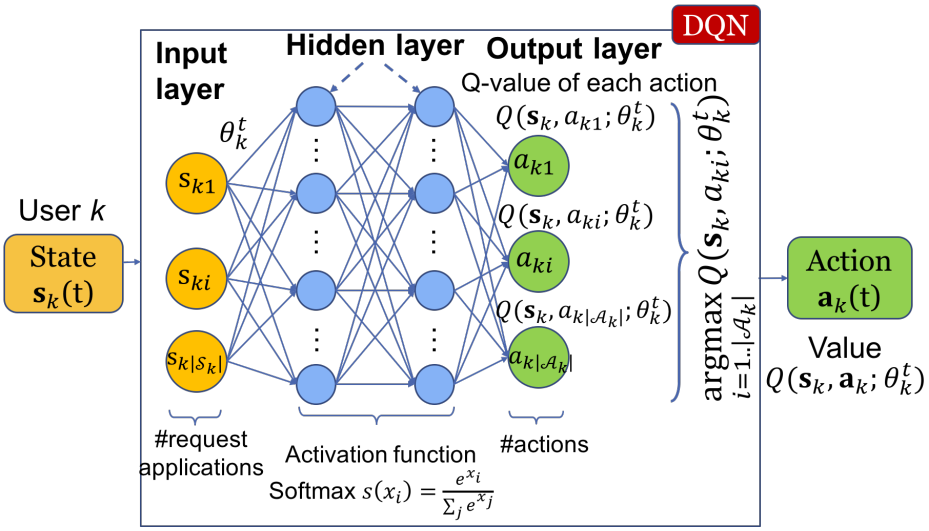


Figure 3.3: DQN structure for user-to-multiple APs association in Sub-6GHz networks

networks. The proposed DQN/DDQN-based methods also define the state and action spaces of agents by (3.8), (3.9), respectively. However, instead of selecting an action through a Q-table, each user now uses a DQN for making its selection.

Figure 3.3 depicts the DQN structure used in our proposed DQN/DDQN-based algorithms. The input layer represents the current state $\mathbf{s}_k(t)$ of user k including variables $x_{bkf}(t)$, and the output layer gives the approximated Q-values for each available action in (3.9), calculated through the hidden layers based on the set of DNN weight parameters θ_k^t. It is worth noting that, here the number of available APs and requested applications are fixed and determine the size of the DQN's input/output layers. This is reasonable as those parameters can be assumed to vary slowly in the case of static or low mobility users, as confirmed by our simulation results in Section 3.7.3. In general, those parameters should undergo slow and smooth variations as compared to users' learning time, and hence users will be able to learn them online without having to re-train their DQN/DDQN from scratch.

In the conventional Q-Learning algorithm presented in Section 3.5, only the Q-value $Q(\mathbf{s}_k(t), \mathbf{a}_k(t))$ for the current state and selected action $\mathbf{a}_k(t)$ is calculated at each iteration and memorized in a Q-table, as in [50]. By contrast, in the DQL approach taken here, the goal of the DNN at each user device is to learn an approximated Q-value function. At each iteration, this Q-function approximation is updated for all available

actions at the same time based on the DQN [48] or DDQN [49] techniques, making Q-Learning applicable to large state/action spaces, and hence, to large-scale networks.

After performing its selected action, user k receives its immediate reward $\Gamma_k(t)$, which is used to update the set of DNN parameters θ_k^t. In the case of DQN, this update is made such that the loss function below is minimized through stochastic gradient descent, given the discount factor γ,

$$\mathcal{L}_k^{\text{DQN}} = \left[Q(\mathbf{s}_k(t), \mathbf{a}_k(t); \theta_k^t) - \left(\Gamma_k(t) + \gamma \max_{\mathbf{a}'_k} Q(\mathbf{s}_k(t+1), \mathbf{a}'_k; \theta_k^t) \right) \right]^2. \tag{3.22}$$

From (3.22), it can be observed that DQN Q is used both for action selection and evaluation, which is useful for saving the computational burden and memory storage of user devices, but may suffer from substantial Q-value overestimation issues [49]. On the contrary, by using two DQNs, one for action selection and the other for Q-value estimation, the DDQN is expected to overcome this drawback but at the cost of higher memory space consumption and increased computational complexity, which may be quite detrimental for computation and battery-limited user devices. Namely, a DDQN combines two different DQNs: in addition to the first DQN Q, a second DQN Q' is built with the same structure, however its set of parameters ${\theta'}_k^t$ is copied from the first only DQN periodically, i.e., every l frames. Then, DQN Q serves for action selection, while DQN Q' serves for state/action evaluation. In this case, the set of parameters ${\theta'}_k^t$ is updated by minimizing the following loss function,

$$\mathcal{L}_k^{\text{DDQN}} = \left[Q(\mathbf{s}_k(t), \mathbf{a}_k(t); \theta_k^t) - \left(\Gamma_k(t) + \gamma Q' \left(\mathbf{s}_k(t+1), \arg\max_{\mathbf{a}'_k} Q(\mathbf{s}_k(t+1), \mathbf{a}'_k; \theta_k^t); {\theta'}_k^t \right) \right) \right]^2. \tag{3.23}$$

In both Eqs. (3.22) and (3.23), the immediate reward $\Gamma_k(t)$ is the same as (3.12) defined in the proposed QL-based method. Based on that, we also designe two methods, namely, the *Proposed Fully Distributed DQN (DDQN) Association* and the *Proposed Partially Distributed DQN (DDQN) Association* as the same way with those in case of QL-based algortithm. Namely, *Proposed Fully Distributed DQN (DDQN) Association* methods calculate the terms c_{2k}^r, c_{2k}^d of the reward (3.12) by (3.15) and (3.16), respectively,

Algorithm 3.2: Proposed DQN/DDQN-based User Association Algorithms

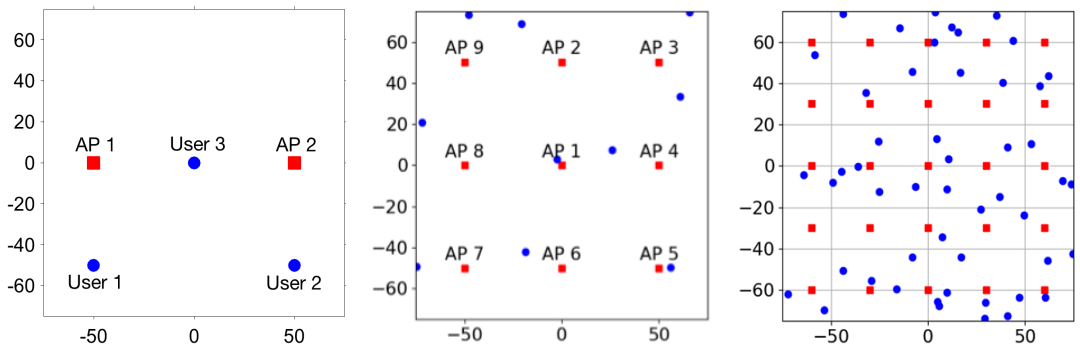
Decay factor λ, greedy factor ε, weights w_{1k}, w_{2k};
for *each user* $k \in \mathcal{K}$ **do**
    Initialize DQN/DDQN Q with random weight values θ_k;
    Random initial state $\mathbf{s}_k$;
for $t = 1, 2, \ldots, T$ **do**
    for *each user* $k \in \mathcal{K}$ **do**
        $\varepsilon \leftarrow \varepsilon \times \lambda$;
        if *random number* $p < \varepsilon$ **then** Select action $\mathbf{a}_k$ randomly;
        else Select action $\mathbf{a}_k$ with highest $Q(\mathbf{s}_k, \mathbf{a}_k; \theta_k^t)$;
    for *each AP* $b \in \mathcal{B}$ **do**
        Consider requests, select users/applications (if necessary) by greedy method ;
        Feedback to users, data transmission;
    for *each user* $k \in \mathcal{K}$ **do**
        Calculate the reward of action $\mathbf{a}_k$ by (3.12) ;
        Update θ_k^t by (3.22) or (3.23);
        Move to the new state $\mathbf{s}_k \leftarrow \mathbf{s}_k^{'}$;

whereas *Proposed Partially Distributed DQN (DDQN) Association* algorithms use (3.20) and (3.21). The pseudo-code of the proposed DQN/DDQB-based algorithm are then given by Algorithm 3.2.

3.7 Numerical Evaluation

3.7.1 Simulation settings

We assess our proposed methods on three scenarios by varying the number of APs and users. Namely, with a small scale, scenario 1 is composed of 2 APs and 3 fixed users, whereas scenario 2 is composed of 9 APs and 10 users uniformly distributed over the network area as shown in Figs. 3.4a and 3.4b, respectively. In Fig. 3.4c, scenario 3 with a larger-scale network is composed of 25 APs and 50 uniformly distributed users. Scenario 1 with a few of APs and users is deployed in order to analyze in detail the basic performance of the proposed methods, whereas scenarios 2 and 3 would



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 3.4: Simulation scenarios

correspond to applications in, e.g., smart factories, where associating devices with multiple APs simultaneously would help achieving multiple applications with various QoS constraints.

First, the proposed QL-based methods are evaluated assuming that each user requires two applications with different rate requirements $R_{k,1} = 6$ Mpbs, $R_{k,2} = 3$ Mbps in small-scale scenarios 1 and 2.

Next, we evaluate the proposed DQN/DDQN-based algorithms in both small and larger-scale networks. Namely, in scenario 2 (Fig. 3.4b), we consider two cases: first is the same as above, i.e., each user requires rate requirements $R_{k1} = 6$ Mbps, $R_{2k} = 3$ Mbps for two different applications, and second, each user requests three applications among which two applications have the same minimum rate requirements as in the first case, while the third application has a maximum delay time requirement of $T_{3k} = 1$ ms. In the larger-scale network (scenario 3 - Fig. 3.4c), each user requires two applications with minimum rate requirements $R_{k1} = 6$ Mbps and maximum delay time $T_{3k} = 1$ ms.

In all evaluations, block Rayleigh fading channels are assumed, where each user channel coefficient remains fixed during a frame, but changes randomly across frames. Users are assumed fixed during each episode of 2500 frames in scenario 1 and of 10000 frames in scenarios 2 and 3. It is reasonable to assume that user positions will remain fixed during the users' learning time under low user mobility scenarios, as will be confirmed by the simulations results. However, results are averaged over 100 random user positions. Remaining parameters values are given in Table 6.1. For fair comparison, reward weights (w_1, w_2) have been manually tuned to yield the "best"

Table 3.1: Simulation Parameters

Parameter	Description
Transmit Power P_{bkf}	5 dBm
Noise power σ_n^2	-169 dBm/Hz
Bandwidth	10 MHz
Channel fading model	Rayleigh fading
Path loss model	$140.7 + 37.6\log_{10}(d)$ (d: AP-user distance [Km])
Mean packet arrival rate ($\lambda_{bkf}\frac{1}{\mu_{bkf}}$)	0.1 Mbps
Learning rate α	0.8
Discount factor γ	0.9
Epsilon ε	0.5
Decay factor λ	0.995
Weight (w_1, w_2)	(0.8, 0.2)

performance for each algorithm. That is, preliminary evaluations over varying values of (w_1, w_2) have shown that the best setting for both proposed and baseline algorithms was (0.8, 0.2), though the algorithms do not exhibit large performance variations for $(w_1, w_2) \in [0.1, 0.9]^2$. Therefore, in the sequel, (w_1, w_2) will be fixed to (0.8, 0.2), though marginal performance gains may be achieved by an exhaustive search over (w_1, w_2), at the cost of higher computational complexity.

The DQN/DDQN is built with two hidden fully connected layers using Softmax activation function. The number of neural nodes per hidden layer is 16, the memory size is set to 100. The period l for updating the weights of the DDQN target network is set to 5. Finally, it is worth mentioning that in our proposed algorithms, the training phase is performed online, which allows to assess their performance when users need to learn the mobile environment from scratch. This is possible thanks to the simple DQN structure with reduced number of nodes and layers that guarantees good convergence behaviors, as shown in the sequel.

3.7.2 Benchmark schemes

First, to evaluate the effectiveness of the distributed approaches, we compare the proposed schemes with a centralized exhaustive search giving the optimal solution and

a centralized QL method. We then also compare our proposed methods with reference distributed algorithms for a comprehensive evaluation. These baseline methods are described as follows.

1) Centralized Exhaustive Search (*Ref. ES*): In each frame, the best association is selected for each user and application through exhaustive search. First, a list of possible associations for each user and application to a certain AP, i.e., satisfying all requirements without overloading any AP, is issued. Then, the association providing the largest instantaneous sum-rate is selected. If no possible association exists, i.e., some QoS or AP load constraints are impossible to satisfy, the association yielding the highest sum-rate without those constraints is chosen. Given the intractably high complexity of such exhaustive search, this algorithm can be only evaluated in scenario 1 (Fig. 3.4a).

2) Centralized Q-learning (*Ref. centralized QL*): In this centralized algorithm, we consider a genie who decides the associations for all users and applications simultaneously, for maximizing the total sum-rate of the system and satisfying user QoS requirements. The state, action space and the cost function in the centralized Q-learning model for problem (3.7) are defined as follows.

- **State:** The state variable $\mathbf{s}(t)$ is now defined as the current association of all users and all required applications to all APs, which is written as

$$\mathbf{s}(t) \in \mathcal{S} = \left\{ x_{bkf}(t) \in \{0,1\},\ \forall b \in \mathcal{B}, \forall k \in \mathcal{K}, \forall f \in \mathcal{F}_k \right\}. \tag{3.24}$$

  Given the constraint that each application may be served by only one AP, the size of this state space is $(\mathbf{C}^1_{|\mathcal{B}|})^{|\mathcal{K}|\times|\mathcal{F}|} = B^{KF}$, which obviously grows exponentially with the number of users and applications.

- **Action:** Similarly, action $\mathbf{a}(t)$ is the set of association requests towards APs, for each user and application, i.e.,

$$\mathbf{a}(t) \in \mathcal{A} = \left\{ a_{bkf}(t) \in \{0,1\} \mid \sum_{b\in\mathcal{B}} a_{bkf}(t) = 1,\ \forall b \in \mathcal{B}, \forall k \in \mathcal{K}, \forall f \in \mathcal{F}_k \right\}. \tag{3.25}$$

  The maximum number of candidates in the action space is hence $(\mathbf{C}^1_{|\mathcal{B}|})^{|\mathcal{K}|\times|\mathcal{F}|} =$

B^{KF}.

- **Reward:** Given the centralized structure of this model, the reward function sums up all user individual rewards, as follows

$$\Gamma(t) = \sum_{k \in \mathcal{K}} w_1 c_{1k}(t) + w_2 (c^r_{2k}(t) + c^d_{2k}), \tag{3.26}$$

where $c_{1k}(t), c^r_{2k}(t), c^d_{2k}(t)$ are given as in Eqs. (3.13), (3.20) and (3.21), respectively.

- **Q-value:** The Q-value of current state $\mathbf{s}(t)$ and selected action $\mathbf{a}(t)$ is now updated by

$$Q\Big(\mathbf{s}(t), \mathbf{a}(t)\Big) = (1-\alpha) Q\Big(\mathbf{s}(t), \mathbf{a}(t)\Big) + \alpha \left(\Gamma(t) + \beta \max_{\mathbf{a} \in \mathcal{A}} Q\Big(\mathbf{s}(t+1), \mathbf{a}\Big)\right). \tag{3.27}$$

The pseudo-code for the centralized Q-learning algorithm is given in Algorithm 3.3. Similar to the proposed distributed methods, an ε-greedy QL-based strategy is applied by the genie, as shown in Algorithm 3.3.

Algorithm 3.3: Centralized QL-based User Association Algorithm

```
Learning rate α, discount factor γ, decay factor λ, greedy policy factor ϵ, weight
  w_1, w_2;
Initialize Q-table Q with zero-values;
Random an initial state s;
while true do
    ϵ ← ϵ * λ;
    if random a number p < ϵ then
        Select action a randomly;
    else
        Select action a corresponding to the highest Q(s, a);
    Calculate the reward of action a by (3.26);
    Update Q(s, a) by (3.27);
    s ← s′          ▷ s′ is the new state after making selected action a;
```

3) Reference distributed methods: Since no distributed algorithms so far enable the association of a user to multiple APs, we consider the baseline algorithms which

follows the structure of proposed schemes. However, the difference lies in the definition of the reward function. Namely, the reward function for an action selected by user k is given by (3.12), but where c_{1k} and c^r_{2k}, c^d_{2k} are defined similarly to the rewards of [30] under QoS constraints,

$$c_{1k} = \sum_{b\in\mathcal{B}} \sum_{f\in\mathcal{F}_k} a_{bkf}(t)\mathcal{I}\left(x_{bkf}(t) = 1\right)(r_{bkf}(t) - R_{kf}), \tag{3.28}$$

$$c^r_{2k} = \sum_{b\in\mathcal{B}} \sum_{f\in\mathcal{F}_{kr}} a_{bkf}(t)\mathcal{I}\left(x_{bkf}(t) = 0\right)(r_{bkf}(t) - R_{kf}). \tag{3.29}$$

$$c^d_{2k} = \sum_{b\in\mathcal{B}} \sum_{f\in\mathcal{F}_{kd}} a_{bkf}(t)\mathcal{I}\left(x_{bkf}(t) = 0\right)(T_{kf} - d_{bkf}(t)). \tag{3.30}$$

By calculating the difference between instantaneous rate $r_{bkf}(t)$ and the required rate R_{kf}, Eq. (3.28) expresses how much gain user k obtains if it is served by AP b for f, whereas Eqs. (3.29) and (3.30) expresses the penalty incurred to user k if application f is dropped.

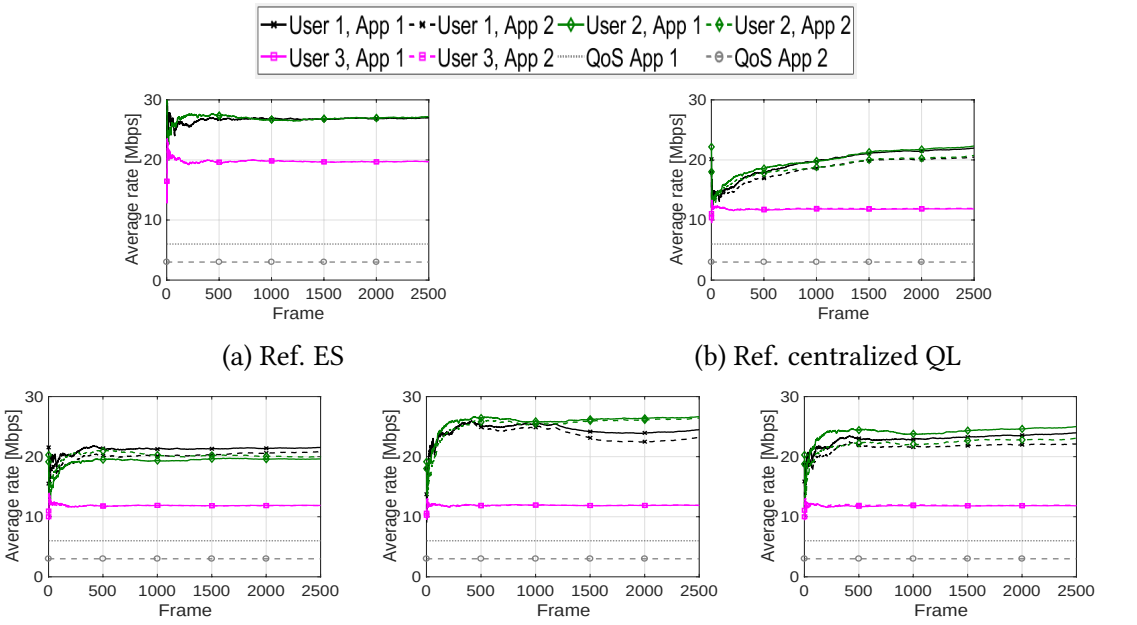
Based on that, the following distributed benchmarks are compared to evaluate the proposed algorithms:

- **Reference distributed Q-Learning** (*Ref. distributed QL*): This method is similar to the proposed fully distributed QL algorithm, but calculates the rewards by Eqs. (3.28), (3.29) and (3.30).

- **Reference Basic DQN (DDQN)** (*Ref. basic DQN (DDQN)*): This method is similar to *Ref. distributed QL*, but uses DQN (DDQN).

- **Reference Basic DQN (DDQN)-Single AP** (*Ref. basic DQN (DDQN)-Single AP*): This method is similar to *Ref. basic DQN (DDQN)*, but constrains each user to request only one AP for all its applications at each scheduling frame as the most of existing studies.

- **Reference greedy** (*Ref. Greedy*): At each scheduling frame, the APs providing best instantaneous Signal-to-Noise Ratios (SNR) will be requested by each user. Namely, the user will send its association request to the AP with higher SNRs,



(a) Ref. ES (b) Ref. centralized QL

(c) Ref. distributed QL (d) Prop. fully distributed QL (e) Prop. partially distributed QL

Figure 3.5: The average achieved data rate per user per application, scenario 1, two applications per user

according to the order of QoS priorities. Then the number of requested APs is equal to the number of required applications of each user.

For convenience, we denote the *Proposed Fully Distributed QL (DQN,DDQN) Association* and *Proposed Partially Distributed QL (DQN,DDQN) Association* algorithms as *Prop. fully distributed QL (DQN,DDQN)* and *Prop. partially distributed QL (DQN,DDQN)*, respectively.

3.7.3 Simulation results

3.7.3.1 Scenario 1, two applications, R_{k1} = 6 Mbps, R_{k2} = 3 Mbps

This small-scale network scenario is considered so as to compare the performances of distributed QL-based algorithms with centralized methods including the high-complexity exhaustive search and centralized QL algorithms.

Fig. 3.5 shows the achieved data rates averaged over frames for each application and each user. It can be observed that user 3 with the worst interference environment

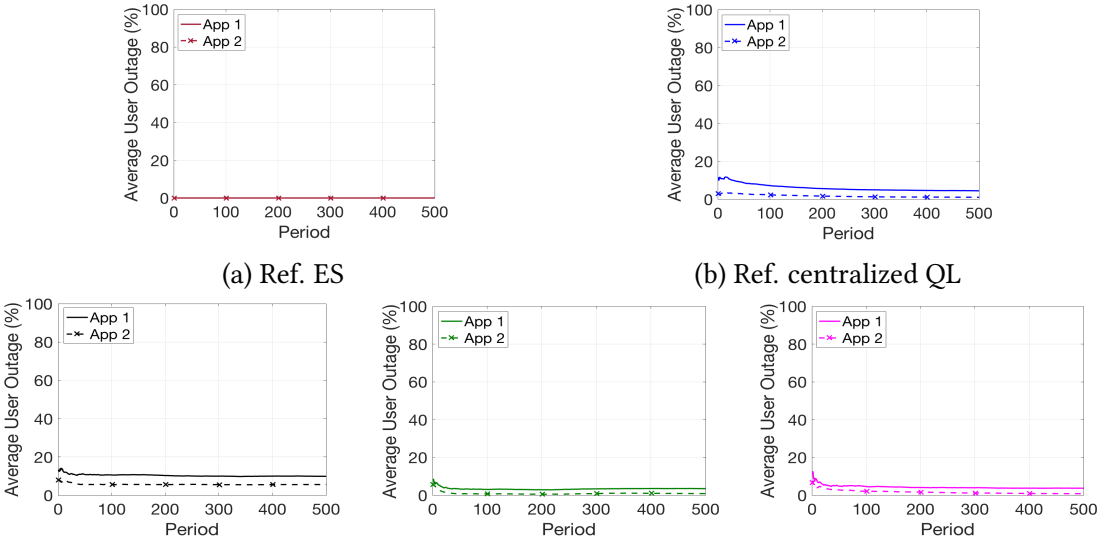
gets a lower rate compared to users 1 and 2, in all algorithms. For users 1 and 2 with better wireless environment, the performances of the proposed fully distributed and partially distributed algorithms outperform that of the reference distributed and even the centralized Q-learning algorithm, which requires much more frames to converge given the much larger state space. In addition, the proposed algorithms do not degrade too much the performance achieved by exhaustive search. Moreover, the partially distributed and centralized approaches show the most effective learning trend where the application 1 with higher R_{kf} obtains a higher rate, unlike the reference distributed and proposed fully distributed schemes. This suggests that the reward function design used for the proposed partially distributed algorithm is well suited for handling heterogeneous QoS requirements.

Table 3.2 presents the percentage of outage events averaged over users and for each application, where an outage occurs whenever the short term average user rate $\bar{r}_{bkf}(t) = \sum_{i=t}^{t+T} x_{bkf}(i) r_{bkf}(t)$ over a small number of frames $T = 5$ (5 ms for 1ms-frame) falls below the required application rate $R_{k,f}$. Fig. 3.6 shows the corresponding evolution of these user outages over frames. Again, we can see that the proposed fully distributed and proposed partially distributed algorithms outperform the performance of the reference basic distributed one, while performing close to that achieved by centralized Q-learning. Also, it can be observed that the application 1 with larger $R_{k,f}$ has a higher outage ratio compared to application 2 in all learning-based algorithms.

Table 3.2: Average outage ratio (%) per application (App), scenario 1, two applications

Scenario 1	Ref. ES	Ref. Centralized QL	Ref. distributed QL	Prop. fully distributed QL	Prop. partially distributed QL
App 1	0.0	4.46	9.90	3.52	3.74
App 2	0.0	1.07	5.61	0.90	0.84

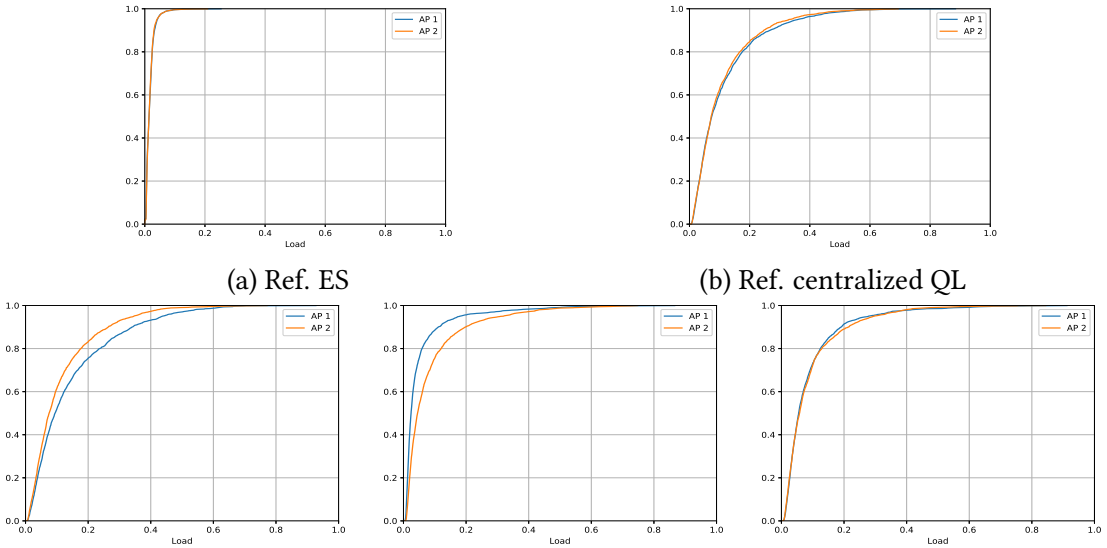
Next, the cumulative distribution function (CDF) of the APs' load is shown in Fig. 3.7. Given the symmetry of the situations of AP 1 and AP 2, their loads should be shared equally. Clearly, a better load balancing is achieved by the exhaustive, centralized and proposed partially distributed algorithm as compared to the reference and proposed fully distributed algorithms.



(a) Ref. ES (b) Ref. centralized QL

(c) Ref. distributed QL (d) Prop. fully distributed QL (e) Prop. partially distributed QL

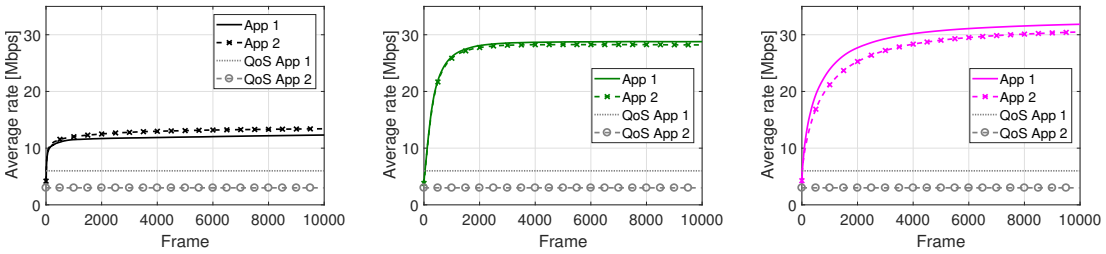
Figure 3.6: Percentage of user outage per application averaged over frames, Scenario 1, two applications per user



(a) Ref. ES (b) Ref. centralized QL

(c) Ref. distributed QL (d) Prop. fully distributed QL (e) Prop. partially distributed QL

Figure 3.7: CDF of the load per AP, Scenario 1, two applications per user



(a) Ref. distributed QL (b) Prop. fully distributed QL (c) Prop. partially distributed QL

Figure 3.8: The average data rate per application, scenario 2, two applications per user

3.7.3.2 Scenario 2, two applications, R_{k1} = 6 Mbps, R_{k2} = 3 Mbps

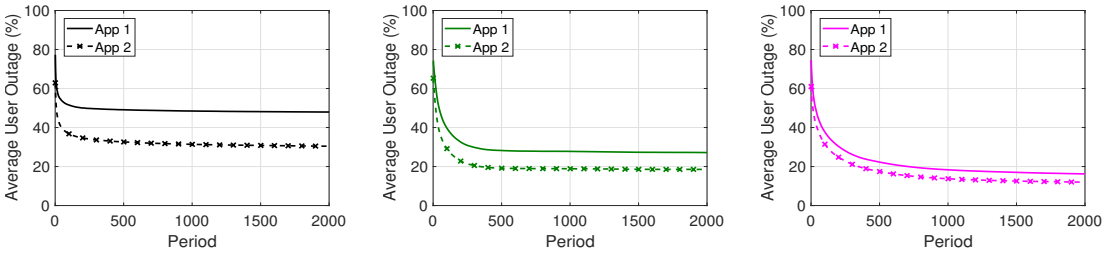
In this scenario, due to the exploding number of states and actions ($9^2 0$ each), computing solutions with exhaustive search and centralized Q-Learning becomes infeasible. Therefore, the proposed algorithms are compared with the reference basic distributed algorithm.

QL-based methods:

We first compare the QL-based methods, namely *Prop. fully distributed QL*, *Prop. partially distributed QL* and *Ref. distributed QL*.

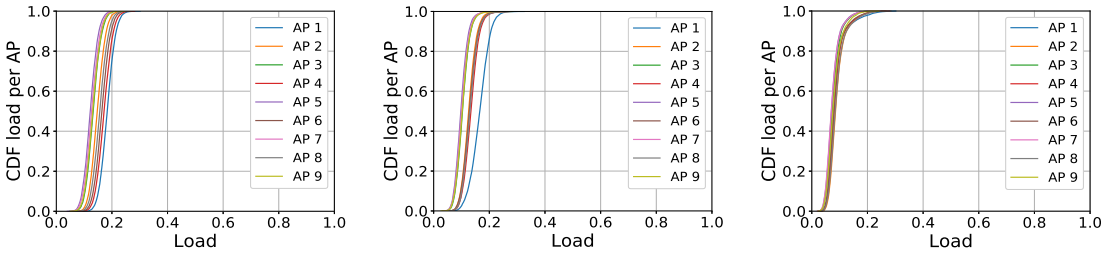
Figure 3.8 shows the achievable data rate per application averaged over all users and positions. It can be observed that all considered algorithms converge well. Compared to the baseline method, both proposed algorithms significantly improve the rates of each application. Namely, the rates of *Prop. fully distributed QL* and *proposed partially distributed QL* achieve 100% and 130% higher gains compared to the reference scheme, respectively. In addition, the proposed partially distributed method clearly provides a higher rate to the application with larger requirement R_{kf}, by contrast to the baseline one. This shows that our algorithm successfully adapts its allocated rates to the specific QoS requirements.

Next, we consider the evolution of user outage across frames for each application, averaged over users and positions. This averaged user outage is evaluated based on short-term served user rate as present in Section 3.7.3.1. As shown in Fig. 3.9, the proposed algorithms outperform the baseline one with 43% (fully distributed) and 66% (partially distributed) lower outage probabilities for application 1 and 39% (fully



(a) Baseline distributed QL (b) Prop. fully distributed QL (c) Prop. partially distributed QL

Figure 3.9: The average user outage per application, scenario 2, two applications per user



(a) Baseline distributed QL (b) Prop. fully distributed QL (c) Prop. partially distributed QL

Figure 3.10: CDF of the load per AP, scenario 2, two applications per user

distributed) and 61% (partially distributed) lower outage probabilities for application 2. Although all algorithms reduce their outage occurrences through learning, our proposed schemes, especially the partially distributed one, not only reduce outage events, but also improve fairness between applications as shown by the tightening of the two curves in Figs. (3.9b) and (3.9c).

Finally, the cumulative distribution function (CDF) of the load for each AP is presented in Fig. 3.10. We observe that the *proposed partially distributed QL* achieves the best fairness in terms of load among APs, while even decreasing the burden per AP.

Hence, our proposed learning-based distributed user to multiple APs association methods enable to jointly enhance global network throughput and application QoS fulfillment, while improving the load balancing among interfering APs and reducing the load per AP.

DQN/DDQN-based methods:

In this part, we evaluate DQN/DDQN-based methods and also *Ref. Greedy*.

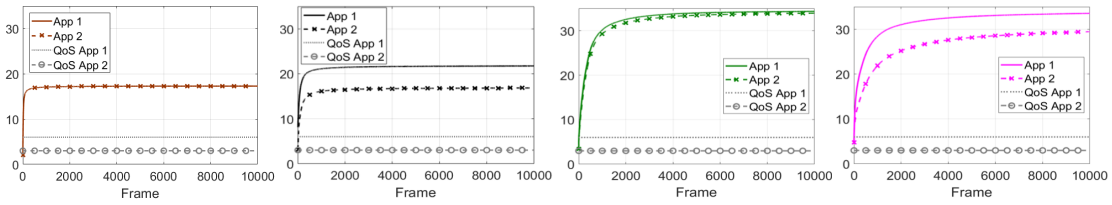
We show the evolution of data rates over scheduling time frames, as well as of outage given by the DQN/DDQN-based algorithms in Figs. 3.11 and 3.12, respectively, then the average sum-rate and outage of all algorithms including *Ref. Greedy* are given in Table 3.3.

Firstly, Fig. 3.11 shows the achievable data rate per application averaged over all users and positions by DQN/DDQN-based algorithms. We can observe that all algorithms converge well. Namely, *Ref. Basic DQN/DDQN-Single AP* converge after about 500 frames, whereas *Ref. Basic DQN/DDQN* and the proposed algorithms need 1000 and 2000 frames to converge, respectively[2]. This is because in the reference basic DQN/DDQN-Single AP algorithms, each user requests the same APs for all of its applications, thereby reducing the action space and taking a shorter time to converge.

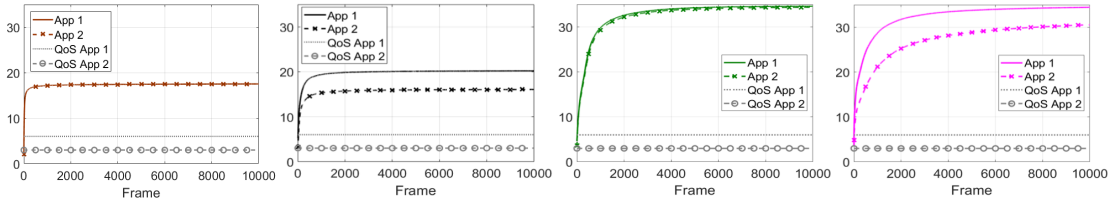
In general, the DQN-based methods achieve similar average rates as their corresponding DDQN-based algorithms. It can be observed that all user-to-multiple AP association algorithms achieve higher sum-rate than *Ref. basic DQN/DDQN-Single AP*. This proves the aforementioned necessity of allowing users to associate with several APs in the context of B5G systems. Compared to *Ref. basic DQN/DDQN*, all proposed algorithms significantly improve the rates of each application. Namely *Prop. fully distributed DQN/DDQN* achieve a 69% rate increase for application 1 and 106% for application 2, whereas *Prop. partially distributed DQN/DDQN* achieve 68% and 85% rate increase for applications 1 and 2, respectively. In addition, Fig. 3.11 also shows that all algorithms can allocate a higher rate to the application with higher requirement R_{kf}, especially for *Ref. Basic-DQN/DDQN* and *Prop. partially distributed DQN/DDQN*.

Next, we consider the evolution of user outage, averaged over all users and positions. Interestingly, although reference user-to-single AP association methods provide the same data rate for all applications, their outage are much different as shown in Figs. 3.11 (a), (e) since they have different QoS requirements. This again indicates the advantage of proposed user-to-multiple APs association methods. As shown in Fig. 3.12, both DQN and DDQN-based reference algorithms are outperformed by proposed algorithms. Namely, *Prop. fully distributed DQN/DDQN* provide 76% and 85% lower outage probabilities for applications 1, 2 respectively, whereas 83% and 86% lower outage levels are observed for applications 1 and 2 by *Prop. partially distributed*

[2]If we consider each frame to be of 1 ms, the proposed algorithms can converge after 2s for the small network Scenario 1.

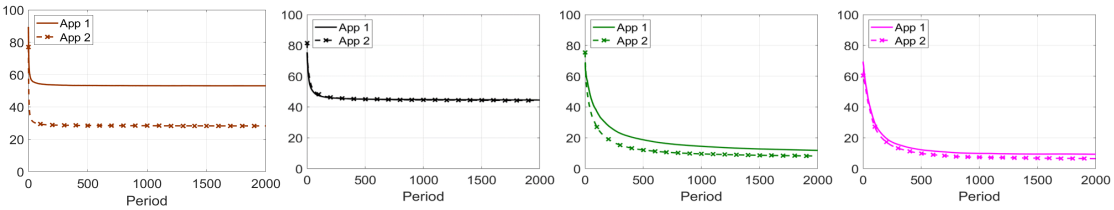


(a) Ref. Basic DQN -Single AP (b) Ref. Basic DQN (c) Prop. fully distributed DQN (d) Prop. partially distributed DQN

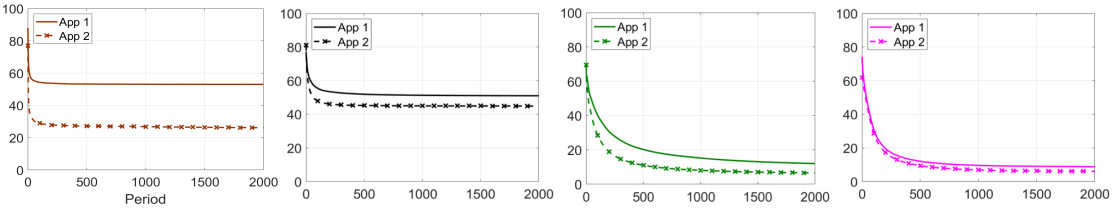


(e) Ref. Basic DDQN-Single AP (f) Ref. Basic DDQN (g) Prop. fully distributed DDQN (h) Prop. partially distributed DDQN

Figure 3.11: Average data rates [Mbps] per application, scenario 2, two applications per user



(a) Ref. Basic DQN-Single AP (b) Ref. Basic DQN (c) Prop. fully distributed DQN (d) Prop. partially distributed DQN



(e) Ref. Basic DDQN-Single AP (f) Ref. Basic DDQN (g) Prop. fully distributed DDQN (h) Prop. partially distributed DDQN

Figure 3.12: Average user outage [%] per application, scenario 2, two applications per user

DQN/DDQN. Compared to their DQN-based counterparts, the DDQN-based proposed methods provide slightly better outage performance, especially for application 2.
In addition, compared to *Prop. fully distributed DQN/DDQN*, *Prop. partially distributed*

Table 3.3: Average sum-rate [Mbps] and outage [%] after convergence, scenario 2, two applications per user

Algorithms	Sum-rate	Outage	
		App 1	App 2
Ref. Greedy	37.9	0.39	73.5
Ref. Basic DQN-Single AP	34.6	53.1	28.3
Ref. Basic DDQN-Single AP	35.1	53.0	26.2
Ref. Basic DQN	38.6	44.6	44.3
Ref. Basic DDQN	36.3	51.0	44.8
Prop. Fully Distributed DQN	68.3	11.8	8.2
Prop. Fully Distributed DDQN	**69.0**	12.0	6.54
Prop. Partially Distributed DQN	64.7	9.35	6.49
Prop. Partially Distributed DDQN	65.0	**8.80**	**6.06**

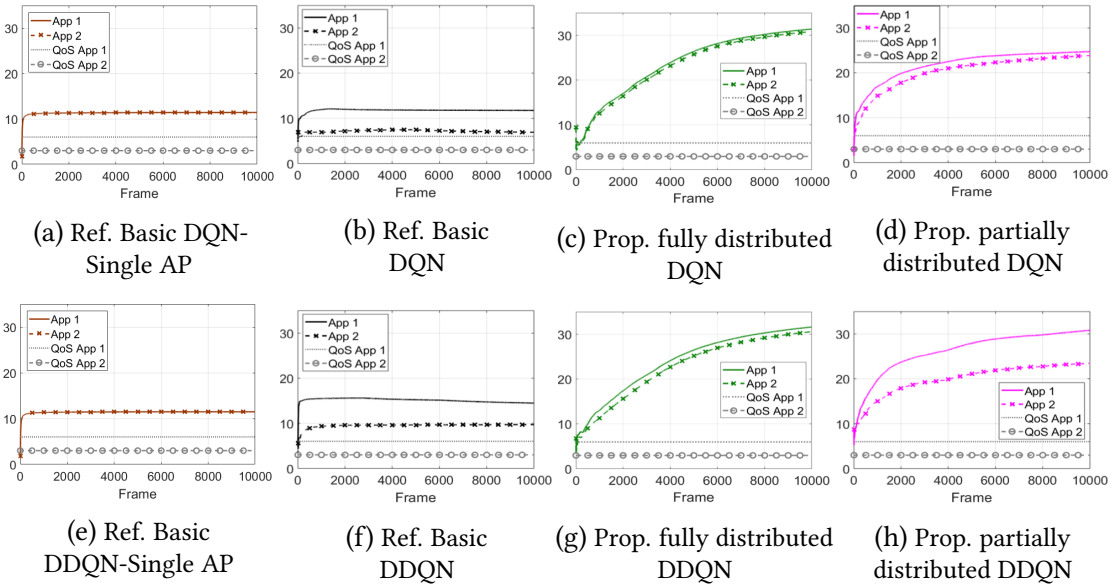
DQN/DDQN achieve better outage fairness between applications, as seen by the smaller gaps between each outage curve as shown in Figs. 3.12.

From Table 3.3, we can observe that *Ref. Greedy* achieves the lowest outage level for application 1 because users always request the best AP for this application. However, application 2 suffers much higher outage, up to 73.5% compared to 0.39% of application 1. Therefore, this algorithm cannot guarantee the fairness among different QoS applications. Moreover, this algorithm is also outperformed by proposed algorithms in terms of sum-rate.

3.7.3.3 Scenario 2, three applications, R_{k1} = 6 Mbps, R_{k2} = 3 Mbps, T_{k3} = 1 ms

In this scenario where each user requests 3 applications, we evaluate DQN/DDQN-based methods in comparison with *Ref. Greedy*.

Fig. 3.13 presents the average user rates for application 1 and 2 for DQN/DDQN-based algorithms. Similarly to the previous case, *Ref. Basic DQN/DDQN-Single AP* provides the same data rate for all applications, while user-to-multiple AP association methods provide higher data rate for application 1 and lower data rate for application 2. Again, among algorithms enabling user-to-multiple APs association, it can be observed that proposed algorithms outperform reference ones. Meanwhile, for the delay-stringent application 3, Fig. 3.14 shows that all DQN/DDQN-based algorithms provide the same performance. However, as shown in Fig. 3.15, the proposed algorithms provide



(a) Ref. Basic DQN-Single AP

(b) Ref. Basic DQN

(c) Prop. fully distributed DQN

(d) Prop. partially distributed DQN

(e) Ref. Basic DDQN-Single AP

(f) Ref. Basic DDQN

(g) Prop. fully distributed DDQN

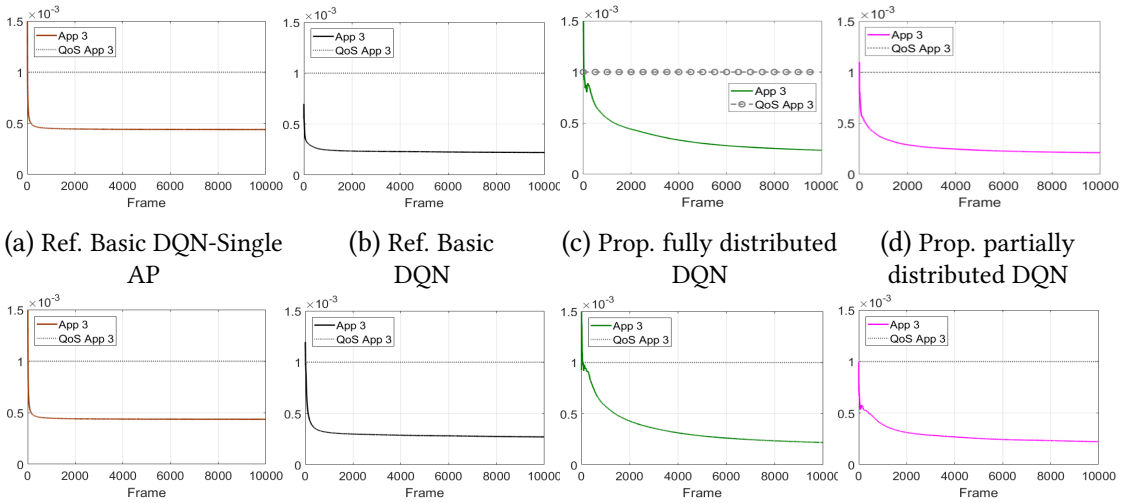
(h) Prop. partially distributed DDQN

Figure 3.13: Average data rates [Mbps] per application, scenario 2, three applications per user

lower outage probabilities not only for applications 1 and 2, but also for application 3 as compared to *Ref. Basic DQN/DDQN-Single AP* and *Ref. Basic-DQN/DDQN*. This proves that our proposed algorithms significantly improve the association decisions as well as rate allocation compared to the benchmarks.

It can be also observed in Fig. 3.13 that the algorithms based on DDQN achieve slightly higher average rates compared to their DQN-based counterparts, in particular for application 1 regarding *Prop. partially distributed DDQN*, and for application 2 regarding *Ref. Basic-DDQN*. The rate gap between the two application curves is also more pronounced for DDQN-based algorithms than that for DQN-based ones.

In addition, it is interesting to observe that *Prop. partially distributed DQN/DDQN* provide the best fairness among applications as the three curves in Fig. 3.15 can be hardly distinguished, whereas *Ref. Basic-DQN* shows the largest gap between the two rate-constrained applications and the delay-constrained application (Fig. 3.15(b)). In conclusion, the reference algorithms cannot handle well the QoS diversity of the multiple applications required by each user, for instance, rate versus delay in this case. This can be explained as follows: *Ref. Basic-DQN/DDQN* calculate the penalties $c_{2k}^{\mathrm{r}}, c_{2k}^{\mathrm{d}}$ by the difference between the served rate/delay and the corresponding QoS



(a) Ref. Basic DQN-Single AP (b) Ref. Basic DQN (c) Prop. fully distributed DQN (d) Prop. partially distributed DQN

(e) Ref. Basic DDQN-Single AP (f) Ref. Basic DDQN (g) Prop. fully distributed DDQN (h) Prop. partially distributed DDQN

Figure 3.14: Average delays [s] per application, scenario 2, three applications per user

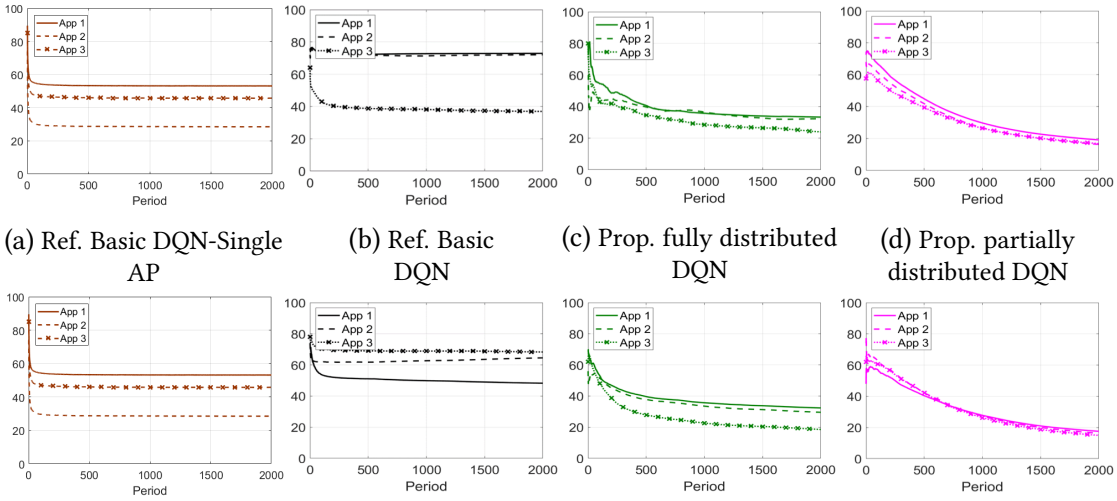
requirement, however these raw differences are hardly comparable among themselves. By contrast, by using the ratio between each QoS requirement and the served rate or delay, the proposed algorithms can alleviate this drawback and hence gaurantee better fairness among applications, as observed in Fig. 3.15.

The average sum-rate and outage after convergence per application of all algorithms for Scenario 1 with three required applications are given in Table 3.4. Again, we can see that the greedy method cannot work well in the context of various QoS constraints, as only application 1 with highest QoS requirement is served well while the other applications undergo high outage. In addition, *Prop. Fully Distributed DQN/DDQN* obtains highest sum-rate, but at the cost of higher outage probability for all applications as compared to *Prop. Partially Distributed DQN/DDQN*.

3.7.3.4 Scenario 3, two applications, R_{k1} = 6 Mbps, T_{k2} = 1 ms

Similar to the previous case, the DQN/DDQN-based methods are compared with *Ref. Greedy* in this scenario.

Considering a dense and large-scale interfering network, Figs. 3.16 and 3.17 show



(a) Ref. Basic DQN-Single AP (b) Ref. Basic DQN (c) Prop. fully distributed DQN (d) Prop. partially distributed DQN

(e) Ref. Basic DDQN-Single AP (f) Ref. Basic DDQN (g) Prop. fully distributed DDQN (h) Prop. partially distributed DDQN

Figure 3.15: Average user outage [%] per application, scenario 2, three applications per user

the average user rate of application 1 and the average delay of application 2 for all DQN/DDQN-based algorithms. It can be observed that even in the large-scale network, *Ref. Basic DQN/DDQN-Single AP* still converge quickly after 500 frames, while *Ref. Basic DQN/DDQN* and the proposed algorithms take more time, though limited to 2000 and 4000 frames, respectively. In spite of a slightly slower convergence compared to the reference schemes, the proposed methods provide higher rate for application 1 which requires the minimum rate QoS, and the same delay for application 2 as compared to *Ref. Basic-DQN/DDQN*, which is lower than that of *Ref. Basic DQN/DDQN-Single AP*. Like previous scenarios, *Ref. Basic-DQN/DDQN* also result into the largest gaps between both outage probability curves as shown in Fig. 3.18(b), (f).

In particular, the average outage probability of application 1 amounts to 76%, and to 25% for application 2. *Prop. fully distributed DQN/DDQN* significantly reduce this gap with 34% outage for application 1 and 25% outage for application 2. As expected, *Prop. partially distributed DQN/DDQN* still provide the best fairness and lowest outage for both applications with 23% and 19% outage probabilities for applications 1 and 2, respectively.

Table 3.5 summarizes the average sum-rate and outage after convergence per

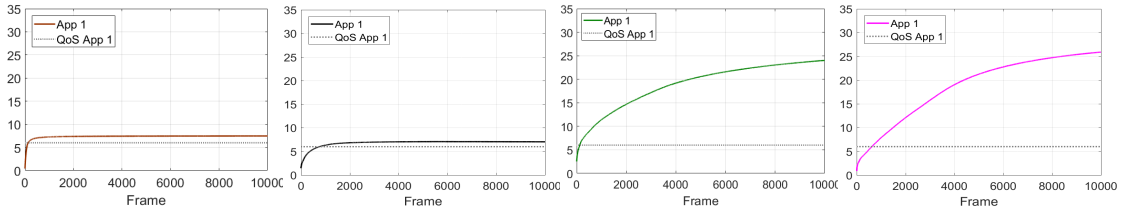
Table 3.4: Average sum-rate [Mbps] and outage [%] after convergence, scenario 2, three applications per user

Algorithms	Sum-rate	Outage		
		App 1	App 2	App 3
Ref. Greedy	37.0	**0.40**	99.9	97.4
Ref. Basic DQN-Single AP	34.4	53.3	30. 0	46.2
Ref. Basic DDQN-Single AP	34.5	53.1	28.5	45.8
Ref. Basic DQN	41.3	73.0	72.1	37.0
Ref. Basic DDQN	42.6	48.3	64.5	68.3
Prop. Fully Distributed DQN	80.6	33.3	32.3	23.9
Prop. Fully Distributed DDQN	**85.1**	32.4	29.6	18.5
Prop. Partially Distributed DQN	72.2	19.1	16.3	16.8
Prop. Partially Distributed DDQN	76.7	**17.6**	**16.3**	**15.0**

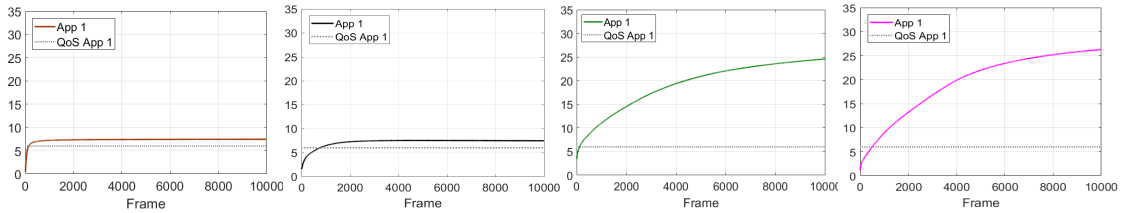
application given by all algorithms for Scenario 2. Similarly to previous scenarios, *Ref. Basic DQN/DDQN-Single AP* achieves the lowest average sum-rate, whereas the greedy method gets the most unfair outage between two applications. In spite of allowing users to associate with multiple APs, *Ref. Basic DQN/DDQN* still obtain low average sum-rate, only about 28 Mbps while getting very high outage for application 1.

Finally, the cumulative distribution function (CDF) of the load per AP is presented in Fig. 3.19. We observe that all algorithms satisfy the load constraint of each AP. In particular, *Prop. partially distributed DQN/DDQN* achieve the best fairness in terms of load among APs, as shown by the more compact distribution of the CDF curves of all APs, as compared to *Ref. Basic-DQN/DDQN-Single AP*, *Ref. Basic-DQN/DDQN* and to *Prop. fully distributed DQN/DDQN*. Furthermore, *Prop. partially distributed DQN/DDQN* achieves the lowest burden per AP as well.

Interestingly, we can observe from our simulation results that, although the DDQN-based proposed methods perform generally better compared to their DQN-based counterparts, the performance gains are rather limited. Therefore, the DQN-based approach may be more appropriate for computation and battery-limited user devices, whereas DDQN-based methods can be useful if there are no such constraints. Hence, the most appropriate proposed method may be chosen according to the specific needs in terms of performance levels, and depending on the user devices' processing, memory

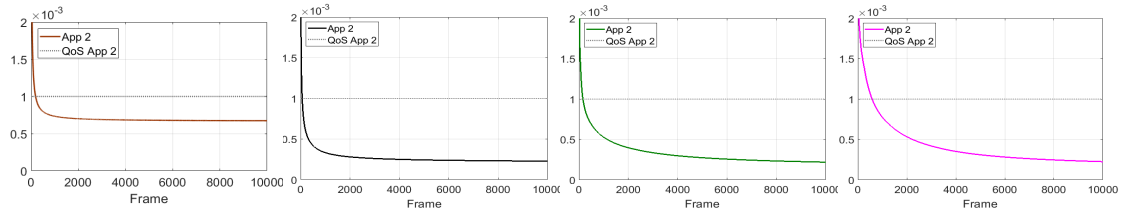


(a) Ref. Basic DQN-Single AP (b) Ref. Basic DQN (c) Prop. fully distributed DQN (d) Prop. partially distributed DQN

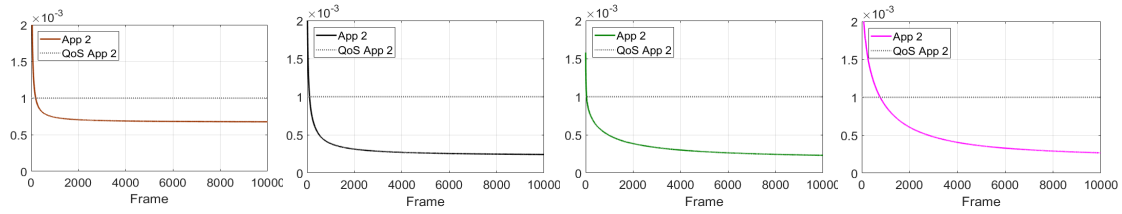


(e) Ref. Basic DDQN-Single AP (f) Ref. Basic DDQN (g) Prop. fully distributed DDQN (h) Prop. partially distributed DDQN

Figure 3.16: The average data rate [Mbps] per application, scenario 3, two applications per user



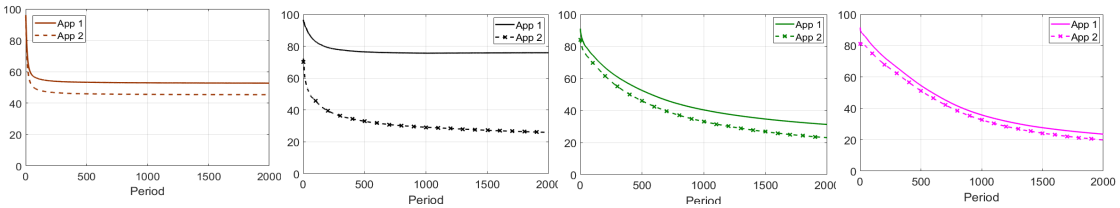
(a) Ref. Basic DQN-Single AP (b) Ref. Basic DQN (c) Prop. fully distributed DQN (d) Prop. partially distributed DQN



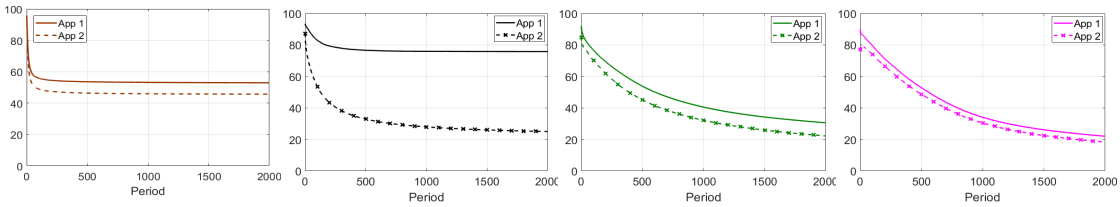
(e) Ref. Basic DDQN-Single AP (f) Ref. Basic DDQN (g) Prop. fully distributed DDQN (h) Prop. partially distributed DDQN

Figure 3.17: Average delay [s] per application, scenario 2, two applications per user

and battery capabilities.

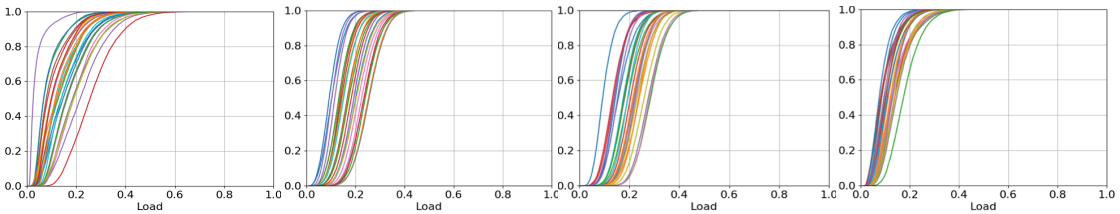


(a) Ref. Basic DQN-Single AP (b) Ref. Basic DQN (c) Prop. fully distributed DQN (d) Prop. partially distributed DQN

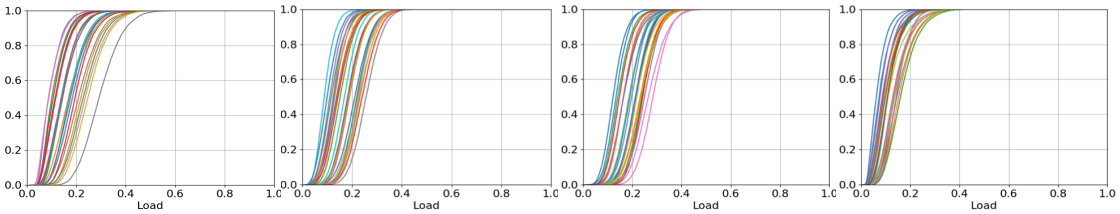


(e) Ref. Basic DDQN-Single AP (f) Ref. Basic DDQN (g) Prop. fully distributed DDQN (h) Prop. partially distributed DDQN

Figure 3.18: Average user outage [%] per application, scenario 3, two applications per user



(a) Ref. Basic DQN-Single AP (b) Ref. Basic DQN (c) Prop. fully distributed DQN (d) Prop. partially distributed DQN



(e) Ref. Basic DDQN-Single AP (f) Ref. Basic DDQN (g) Prop. fully distributed DDQN (h) Prop. partially distributed DDQN

Figure 3.19: CDF load per AP, scenario 3, two applications per user

3.8 Summary

We have investigated the issue of user-to-multiple AP association, where a user requiring various applications with different QoS may be served by multiple APs

Table 3.5: Average sum-rate [Mbps] and outage [%] after convergence, scenario 3 network, two applications per user

Algorithms	Sum-rate	Outage	
		App 1	App 2
Ref. Greedy	34.8	0.12	97.7
Ref. Basic DQN-Single AP	14.9	52.7	45.4
Ref. Basic DDQN-Single AP	15.0	53.0	45.7
Ref. Basic DQN	28.0	75.8	25.4
Ref. Basic DDQN	29.2	75.7	25.0
Prop. Fully Distributed DQN	45.8	33.6	25.4
Prop. Fully Distributed DDQN	46.2	33.3	25.0
Prop. Partially Distributed DQN	45.0	23.5	19.8
Prop. Partially Distributed DDQN	**48.2**	**22.2**	**18.3**

simultaneously, in the current Sub-6GHz system. As a a preliminary study to address this novel issue, we proposed two distributed QL-based methods at user devices with different amounts of local feedback from requested APs. Then, to cope with a large-scale envisioned B5G/6G, we extend these two proposed methods by exploiting the DQN and DDQN-based deep reinforcement learning frameworks. Numerical results show the effectiveness of the proposed methods against reference methods by not only improving multiple objectives such as sum-rate and QoS satisfaction levels, but also enhancing outage fairness among applications, as well as AP load balancing. Unlike reference schemes, the proposed methods are particularly well suited for handling heterogeneous types of QoS requirements.

4

Deep Q-Network based Joint User Association and Beamforming in Integrated Sub-6GHz/mmWave Network

4.1 Introduction

To cope with the stringent requirements of B5G/6G applications, deep interests are turned towards the integration of mmWave and Sub-6GHz interfaces, aiming at jointly exploiting the high reliability of Sub-6GHz, and high capacity and massive spectrum availability of mmWave despite their severe propagation characteristics of high path loss and signal blockage. Although various studies have been conducted so far, many research issues are yet to be solved for enabling a seamless integration of mmWave and Sub-6GHz technologies, in particular the issues of user-to-AP association and performance optimization as mentioned in section 2.

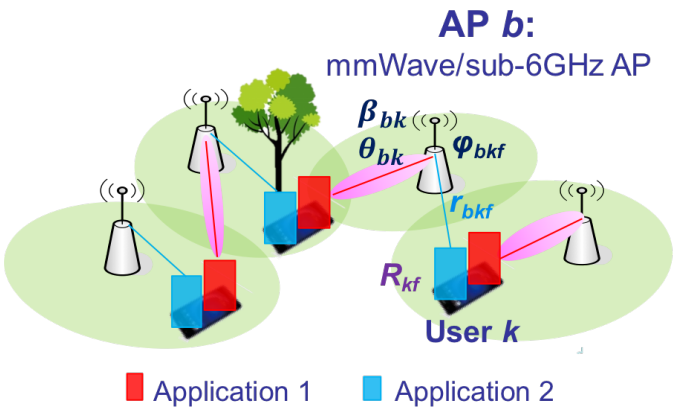


Figure 4.1: An integrated mmWave/Sub-6GHz system

Therefore, in this chapter, we continue to investigate the problem of user-to-multiple APs association, but now in Sub-6GHz/mmWave integrated systems. In such a system, each user may request several applications, and may be supported by multiple APs and *interfaces* simultaneously, making this issue is more complicated compared to the one in the Sub-6GHz systems of the previous chapter. Based on results of Chapter 3, we propose to make use of a DRL technique based on DQN in order to solve this challenging problem. Namely, the mobile users autonomously learn, through their own DQN, the best APs and interface, mmWave or Sub-6GHz, to be requested for each of their applications, by using only locally available information and such that the global utility of the system is optimized. Based on these user and application requests, each AP decides which user/application to serve on each interface, depending on its load capability. In addition, for the mmWave interface, a user clustering and beamforming algorithm is devised, which optimizes the beam direction and beamwidth serving each user cluster at every time frame, so as to maximize the utility of the overall system.

4.2 System Model

We consider the downlink of a wireless network consisting of a set $\mathcal{B}$ of fixed APs and a set $\mathcal{K}$ of randomly located users as shown in Fig. 4.1. Each AP $b \in \mathcal{B}$ and user $k \in \mathcal{K}$ are assumed to operate on both Sub-6GHz and mmWave bands simultaneously.

In each scheduling frame t, each user k requests multiple applications $f \in \mathcal{F}_k$, where $\mathcal{F}_k$ is the set of applications of user k. Each application f requested by user k has a minimum rate constraint[1] R_{kf}. In each scheduling frame and interface $\nu \in \mathcal{I} = \{\text{sub}, \text{mW}\}$, every AP $b \in \mathcal{B}$ has a power budget of p_b^ν and may serve any application, provided its maximum load is not attained.

For Sub-6GHz band, no beamforming is assumed, i.e., the Signal-to-Interference plus Noise Ratio (SINR) from AP b to user k for application f is thus given by

$$\gamma_{bkf}^{\text{sub}}(t) = \frac{p_{bkf}^{\text{sub}}(t) h_{bk}^{\text{sub}}(t)}{\sum_{b' \in \mathcal{B} \setminus \{b\}} p_{b'}^{\text{sub}}(t) h_{b'k}^{\text{sub}}(t) + W_{\text{sub}} \sigma_n^2}, \tag{4.1}$$

where p_{bkf}^{sub} is the transmit power between AP b and user k for application f on Sub-6GHz interface, h_{bk}^{sub} denotes the channel power (i.e., $h_{bk}^{\text{sub}} = |\tilde{h}_{bk}^{\text{sub}}|^2$, with $\tilde{h}_{bk}^{\text{sub}}$ the complex channel coefficient) between AP b and user k, and σ_n^2 is the Additive White Gaussian Noise (AWGN) power. The interference term toward user k served by AP b is composed of all non-serving APs $b' \in \mathcal{B} \setminus \{b\}$ with total transmit power $p_{b'}^{\text{sub}}$.

For mmWave band, the tuple $(\theta_{bk}, \beta_{bk})$ defines parameters of a transmit beam from AP b to user k, where θ_{bk}, β_{bk} are the width and direction[2] of the AP beam, respectively. Every θ_{bk} takes values from the discrete set of beamwidths $\mathbb{D}_\theta$, whereas β_{bk} takes continuous values in $[0, 2\pi]$. Hence, the SINR is given as

$$\gamma_{bkf}^{\text{mW}} = \frac{p_{bkf}^{\text{mW}} h_{bk}^{\text{mW}}(\theta_{bk}, \beta_{bk}, \tilde{\theta}_{bk}, \tilde{\beta}_{bk})}{I_{bkf}^{\text{mW}} + W_{\text{mW}} \sigma_n^2}, \tag{4.2}$$

where p_{bkf}^{mW} is the transmit power between AP b and user k for application f on mmWave interface. $h_{bk}^{\text{mW}}(\theta_{bk}, \beta_{bk}, \tilde{\theta}_{bk}, \tilde{\beta}_{bk})$ is the power of the channel between AP b and user k and is a function of the width and direction of AP b's transmit beam $(\theta_{bk}, \beta_{bk})$ and of user k's receive beam $(\tilde{\theta}_{bk}, \tilde{\beta}_{bk})$ given by [51],

$$h_{bk}(\theta_{bk}, \beta_{bk}, \tilde{\theta}_{bk}, \tilde{\beta}_{bk}) = G_b^{\text{Tx}}(\theta_{bk}, \beta_{bk}) \tilde{G}_k^{\text{Rx}}(\tilde{\theta}_{bk}, \tilde{\beta}_{bk}) \text{PL}_{bk}, \tag{4.3}$$

where PL_{bk} denotes the path loss between AP b and user k. $G_b^{\text{Tx}}(\theta_{bk}, \beta_{bk})$, $\tilde{G}_k^{\text{Rx}}(\tilde{\theta}_{bk}, \tilde{\beta}_{bk})$

[1] Our method can be easily extended to other constraints such as latency as did in Chapter 3

[2] Only azimuthal angle is considered here, however, the elevation angle may be included as well.

are the transmit and receive beam gains of AP b and user k, defined as

$$G_b^{\mathrm{Tx}}(\theta_{bk}, \beta_{bk}) = \begin{cases} G^{\mathrm{main}}(\varphi) & \text{, if } 0 < \varphi = |\beta_{bk}^{LoS} - \beta_{bk}| < \frac{\theta_{bk}}{2} \\ \epsilon & \text{, otherwise} \end{cases} \tag{4.4}$$

$$\tilde{G}_k^{\mathrm{Rx}}(\tilde{\theta}_{bk}, \tilde{\beta}_{bk}) = \begin{cases} G^{\mathrm{main}}(\varphi) & \text{, if } 0 < \varphi = |\beta_{bk}^{LoS} - \tilde{\beta}_{bk}| < \frac{\tilde{\theta}_{bk}}{2} \\ \epsilon & \text{, otherwise} \end{cases} \tag{4.5}$$

Here, $\beta_{bk}^{\mathrm{LoS}}$ is the line-of-sight (LoS) angle between AP b and user k, G^{main} and ϵ are the gains of the main lobe and side lobe beams, respectively, where G^{main} is given by

$$G^{\mathrm{main}}(\varphi) = \frac{2\pi - (2\pi - \theta)\epsilon}{\theta} \quad \text{if } |\varphi| \leq \frac{\theta}{2}. \tag{4.6}$$

In (4.2), I_{bkf}^{mW} denotes the interference power at user k served by AP b,

$$I_{bkf}^{\mathrm{mW}} = \sum_{b' \in \mathcal{B} \setminus \{b\}} \sum_{k', f'} p_{b'k'f'}^{\mathrm{mW}} h_{b'k}^{\mathrm{mW}}(\theta_{b'k'}, \beta_{b'k'}, \tilde{\theta}_{bk}, \tilde{\beta}_{bk}), \tag{4.7}$$

where $k' \in \mathcal{K}$ is served by other APs b' for $f' \in \mathcal{F}_{k'}$.

At the user side, for simplicity, the beamwidth and beam direction are assumed to be fixed as in [51], namely, $\tilde{\theta}_{bk} = 90^\circ$ and $\tilde{\beta}_{bk} = \beta_{bk}^{\mathrm{LoS}}$ if AP b serves user k on mmWave interface. Also, the power from each AP on each interface is assumed to be equally divided among its served users and applications [52].

In case of blockage, i.e., non-LoS (NLoS) situation, mmWave transmissions will result into zero rate [53]. Finally, the rate of user k associated to AP b for application f can be modeled as

$$r_{bkf}(t) = \begin{cases} W_{\mathrm{sub}} \log_2(1 + \gamma_{bkf}^{\mathrm{sub}}(t)), & \text{if user } k \text{ is served on Sub-6GHz band,} \\ W_{\mathrm{mW}} \log_2(1 + \gamma_{bkf}^{\mathrm{mW}}(t)), & \text{if user } k \text{ is served on mmWave band in LoS,} \\ 0, & \text{if user } k \text{ is served on mmWave band in NLoS.} \end{cases} \tag{4.8}$$

AP b's load for serving user k, application f, is set as [28]

$$\phi_{bkf}(t) = \frac{\lambda_{kf}}{\mu_{kf} r_{bkf}(t)}, \tag{4.9}$$

where λ_{kf} is the mean arrival rate in number of packets per seconds, and $\frac{1}{\mu_{kf}}$ the mean packet size of application f in bits. Hence, as in [28], assuming an orthogonal allocation of wireless resources in time or frequency for its own allocated user applications, AP b is overloaded on each interface $v \in \mathcal{I} = \{\text{sub}, \text{mW}\}$ if the sum of all its served applications' loads exceeds 1, namely if

$$\Phi_b^v(t) = \sum_{k \in \mathcal{K}} \sum_{f \in \mathcal{F}_k} x_{bkf}^v(t) \phi_{bkf}(t) > 1, \forall v \in \mathcal{I}, \tag{4.10}$$

where $x_{bkf}^v(t)$ is the association binary variable defined as

$$x_{bkf}^v(t) = \begin{cases} 1, & \text{if AP } b \text{ serves user } k \text{ for application} f \text{ at frame } t \\ & \text{with interface } v, \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

4.3 Problem formulation

We formulate the average network sum-rate maximization problem, under minimum rate constraints R_{kf} for each user k, application f and the APs' load constraints, as follows:

$$\max_{\substack{x_{bkf}^v(t), \\ \theta_{bk}(t), \beta_{bk}(t)}} \mathbf{E}_t \left[\sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} \sum_{f \in \mathcal{F}_k} \sum_{v \in \mathcal{I}} x_{bkf}^v(t) r_{bkf}(t) \right] \tag{4.12}$$

$$\text{s.t. } x_{bkf}^v(t) \in \{0, 1\}, \forall b \in \mathcal{B}, k \in \mathcal{K}, f \in \mathcal{F}_k, \tag{4.12a}$$

$$\theta_{bk}(t) \in \mathbb{D}_\theta, \beta_{bk} \in [0, 2\pi], \forall b \in \mathcal{B}, k \in \mathcal{K}, \tag{4.12b}$$

$$\sum_{b \in \mathcal{B}} \sum_{v \in \mathcal{I}} x_{bkf}^v(t) = 1, \forall k \in \mathcal{K}, \forall f \in \mathcal{F}_k, \tag{4.12c}$$

$$\sum_{b \in \mathcal{B}} \sum_{v \in \mathcal{I}} x_{bkf}^v(t) r_{bkf}(t) \geq R_{kf}, \forall k \in \mathcal{K}, \forall f \in \mathcal{F}_k, \tag{4.12d}$$

$$\Phi_b^v(t) = \sum_{k \in \mathcal{K}} \sum_{f \in \mathcal{F}_k} x_{bkf}^v(t) \phi_{bkf}(t) \leq 1, \forall b \in \mathcal{B}. \tag{4.12e}$$

The objective function (4.12) is the long-term average sum-rate over all APs, users and applications. Eqs. (4.12a) and (4.12b) give the domains of definition for each variable. Eq. (4.12c) constrains each application requested by a user to be served by a unique

AP and one interface. The minimum rate constraint for each application is given by (4.12d). Finally, (4.12e) reflects the AP load constraint for each interface.

This problem is a mixed-integer non-convex optimization problem which cannot be solved in polynomial time. Compared to the initial work of [54], the problem in integrated mmWave and Sub-6GHz networks becomes even more intricate, as we have to optimize not only the user-to-APs association per application, but also the interface allocation and the beamforming parameters. Thus, to solve this problem efficiently and in a distributed manner, the proposed method leverages the powerful capabilities of DRL as explained next.

4.4 Proposed Distributed Algorithm

Similar as in the previous chapter, we first formulate the considered distributed problem as an MDP. Then, the proposed distributed method based on DQN is devised, whereby each user learns the best set of (AP, interface) to request in each time frame, so as to satisfy the heterogeneous QoS requirements of each application.

4.4.1 Formulation as an MDP

As shown in Fig. 4.2, each user is an agent who takes its decision of requesting APs/interfaces for its applications. At each scheduling frame t, the user knows its current state s_t, i.e., its current association between its applications and APs/interfaces, and takes action a_t, i.e., requesting for the next frame the same or different APs/interfaces for each application. The user then moves to a new state s_{t+1} and receives an immediate reward r_t from the environment. One major observation here is that, the transition probability $P(s_{t+1}|a_t, s_t)$ is unknown to the user, since the association decisions for each application on each interface, though based on user requests, are taken by each AP given the current wireless environment and traffic distribution. Therefore, we propose to solve this problem by means of DRL, in particular based on DQN due to its efficiency for handling similar issues in wireless systems and for coping with large state/action spaces.

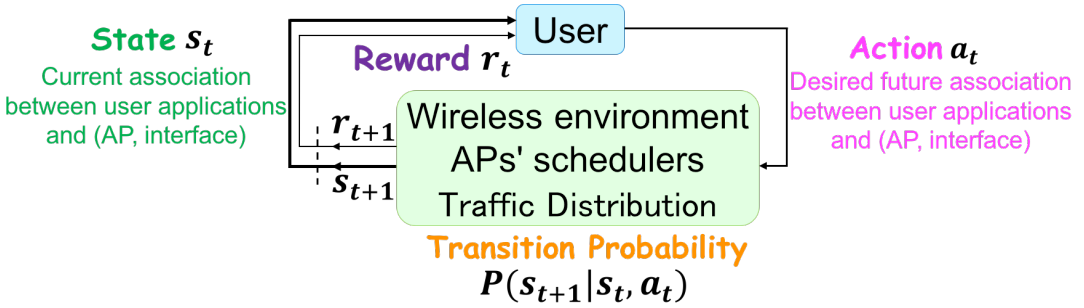


Figure 4.2: MDP model of the considered problem (4.12)

4.4.2 Proposed Distributed DQN-based Method

We define below the state and action spaces of each user.

- **User State:** Unlike the previous definition of state in Chapter 3 where each user state was defined as the association between each user/application and each AP, we now also consider the blockage status between users and APs. Therefore, the state of each user k at time t, denoted by $s_k(t)$ in the state space $\mathcal{S}_k$, is defined as the actual association between user k for its required applications/interfaces and each AP b, and the current estimated blockage status o_{bk} with AP b,

$$s_k(t) \in \mathcal{S}_k = \left\{ x^v_{bkf}(t) \in \{0,1\}, o_{bk}(t) \in \{0,1\}, \forall b \in \mathcal{B}, \forall f \in \mathcal{F}_k, \forall v \in \mathcal{I} \right\}, \quad (4.13)$$

where o_{bk} is equal to 1 if user k's LoS direction towards AP b is blocked, otherwise it is set to 0.

Here, we consider a simple blockage estimation method as follows. Note that the goal of this work is not to propose a new blockage estimation method, hence any other blockage estimation method applies. User k calculates the short-term average SNR over X consecutive scheduling frames based on pilot signals from AP b, i.e.,

$$\bar{\xi}_{bk}(t) = \frac{1}{X} \sum_{\tau=t-X+1}^{t} \xi^{\mathrm{mW}}_{bk}(\tau),$$

where $\xi^{\mathrm{mW}}_{bk}(\tau)$ is the instantaneous SNR on mmWave interface at time τ, then

compares it with the long-term moving average SNR up to time t, defined by

$$\hat{\xi}_{bk}(t) = \frac{(t-1)\hat{\xi}_{bk}(t-1) + \xi_{bk}^{\mathrm{mW}}(t)}{t}.$$

Given a pre-defined threshold Δ_{ξ}, if the gap between short-term and long-term moving average SNRs becomes $|\bar{\xi}_{bk}(t) - \hat{\xi}_{bk}(t)| > \Delta_{\xi}$, the user will estimate that it has changed its blockage status (from unblocked to blocked or vice versa), otherwise that it has not changed. Results show that even with this simple, imperfect blockage status prediction, the proposed method achieves high performances.

- **User Action:** $a_k(t)$ is the desired future association of user k with APs and interfaces for its applications at time t. Users only choose among valid actions, i.e, they request a unique AP/interface per application, and do not request mmWave interfaces estimated to be blocked. Action a_k is hence given by

$$a_k(t) \in \mathcal{A}_k = \Big\{ a_{bkf}^{v}(t) \in \{0,1\} | \sum_{b \in \mathcal{B}} a_{bkf}^{v} = 1, \forall b \in \mathcal{B}, \\ \forall f \in \mathcal{F}_k, \forall v \in \mathcal{I} \text{ and } a_{bkf}^{\mathrm{mW}}(t) = 1 \text{ iff } o_{bk} = 0 \Big\}. \quad (4.14)$$

The DQN designed in Fig. 4.3 enables to approximate the Q-values of each state-action pair, and hence to select the optimal action for each state. The input layer is the state $s_k(t)$ of user k, which includes $x_{bkf}^{v}(t)$ and $o_{bk}(t)$ of (4.13), and the output layer gives the Q-values for each available action as in (4.14). In order to minimize the inherent model complexity, we limit the number of hidden layers as well as of neural nodes as shown in simulation settings. Next, we propose the DQN-based User Association and Beamforming method described in Algorithm 4.1.

Step 1- At the user side, given the probability of exploration $\varepsilon \in (0,1)$, with probability 1-ε, each user selects action $a_k(t)$ as in (4.14) from the output of its DQN based on its current state, then sends its request $a_k(t)$ to the desired (AP, interface) for each application f in $\mathcal{F}_k$ (Lines 5 to 8).

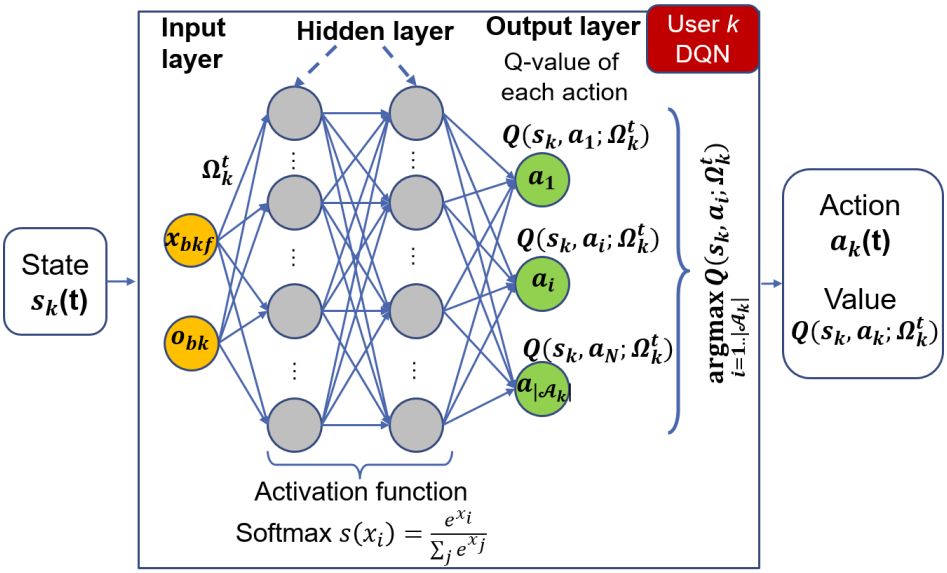


Figure 4.3: DQN structure for user-to-multiple APs association in Sub-6GHz/mmWave integrated network

Algorithm 4.1: Proposed DQN-based user association and Beamforming (*Prop. DQN-UABF*) method

for *each user* $k \in \mathcal{K}$ **do**
    Initialize DQN Q with random weight values Ω_k;
    Random initial state s_k;
for $t = 1, 2, \ldots, T$ **do**
    for *each user* $k \in \mathcal{K}$ **do**
        $\varepsilon \leftarrow \varepsilon \times \lambda$;
        if *random number* $p < \varepsilon$ **then** Select action a_k randomly;
        else Select action a_k with max $Q(s_k, a_k; \Omega_k^t)$;
    for *each AP* $b \in \mathcal{B}$ **do**
        Select users for Sub-6GHz by greedy method [54];
        Select users for mmWave by Algorithm 4.2;
        Feedback to users, downlink transmission;
    for *each user* $k \in \mathcal{K}$ **do**
        Calculate reward of action a_k by (4.18) ;
        Update Ω_k^t by (4.21);
        Move to the new state $s_k \leftarrow s'_k$;

Step 2- After receiving all user requests, APs perform different user association strategies for each interface. Denoting $\mathcal{R}_b^\nu, \mathcal{K}_b^\nu$ as the set of users requesting AP b with interface ν and the set of users accepted by AP b with interface ν, respectively,
• For ν = Sub-6GHz, each AP selects users based on its utility function $\mathcal{U}_b(\mathcal{K}_b^{\text{sub}})$, i.e., such that the sum-rate of selected users/applications satisfying the load constraint (4.12e) is maximized,

$$\mathcal{U}_b(\mathcal{K}_b^{\text{sub}}) = \sum_{k \in \mathcal{K}_b^{\text{sub}}} \sum_{f \in \mathcal{F}_k^{\text{sub}}} r_{bkf}(t) \mathbb{I}(\Phi_b^{\text{sub}}(t) \leq 1), \tag{4.15}$$

where $\mathbb{I}(.)$ is the indicator function. To solve this, we make use of the greedy method of [54], where APs select their best users with lowest load in (4.9), until they are overloaded.
• For ν = mmWave, given the narrow beam characteristics, each AP needs to consider the relative positions of requesting users for beamforming optimization. We propose Algorithm 4.2 that clusters requesting users and optimizes beamforming parameters. The utility function of each user k is thus set as the sum-rate of its applications satisfying (4.12d),

$$\mathcal{U}_k(b) = \sum_{f \in \mathcal{F}_k} a_{bkf}^{\text{mW}}(t) x_{bkf}^{\text{mW}}(t) r_{bkf}(t) \mathbb{I}(r_{bkf}(t) \geq R_{kf}). \tag{4.16}$$

At AP side, due to the lack of information about other APs nor the interference experienced by user k, the AP can not exactly know r_{bkf}. Instead, it makes use of $\tilde{r}_{bkf}(t) = W_{\text{mW}} \log_2(1 + \tilde{\gamma}_{bkf}^{\text{mW}}(t))$, where SINR $\tilde{\gamma}_{bkf}^{\text{mW}}$ is estimated under the assumption of side lobe beam interference and maximum transmit power (i.e., the power budget) from all other APs. Then the user with highest utility (4.16) estimated by $\tilde{r}_{bkf}$ is selected as the seed of the cluster C_b. From this seed, the AP considers its neighboring users to be added to the cluster in a greedy manner. The beamwidth is defined as the smallest value in set $\mathbb{D}_\theta$ that covers all users in the cluster, and the beam direction is taken as the center of the beamwidth. One candidate is added to the cluster if, given the new beamwidth and direction, this adjunction does not decrease AP b's utility given as the

estimated sum-rate of selected users/applications satisfying load constraint (4.12e),

$$\mathcal{U}_b(C_b, \theta_b) = \sum_{k \in \mathcal{K}_b^{\mathrm{mW}}} \sum_{f \in \mathcal{F}_k^{\mathrm{mW}}} \tilde{r}_{bkf}(t) \mathbb{I}(\Phi_b^{\mathrm{mW}}(t) \leq 1). \tag{4.17}$$

The algorithm 4.2 stops when (4.17) stops increasing.

Algorithm 4.2: User Clustering and AP beamforming

1 Cluster $C_b \leftarrow \emptyset$, $\theta_b \leftarrow \theta^{\min}$;
2 **for** *each user* $k \in \mathcal{R}_b^{\mathrm{mW}}$ **do**
3 Compute the user utility of user k by (4.16);
4 Choose the best user k^* with highest utility, then
  $C_b \leftarrow C_b \cup \{k^*\}$,
  $\mathcal{R}_b \leftarrow \mathcal{R}_b \backslash \{k^*\}$;
5 **while** $\mathcal{R}_b \neq \emptyset$ **do**
6 *Candidates* $\leftarrow \emptyset$;
7 **for** *each user* $k' \in \mathcal{R}_b$ **do**
8 **if** $\exists \theta \in \mathbb{D}_\theta$ *covers* $\{C_b \cup k'\}$ **then**
9 *Candidates* $\leftarrow$ *Candidates* $\cup \{k'\}$;
10 **if** *Candidates* $== \emptyset$ **then** break;
11 **for** $k' \in$ *Candidates* **do**
12 $\theta' = \min \{\theta \in \mathbb{D}_\theta | \theta$ covers $\{C_b \cup k'\}\}$;
13 **if** $\mathcal{U}_b(C_b \cup k' | \theta') \geq \mathcal{U}_b(C_b | \theta_b)$ **then**
14 $C_b \leftarrow C_b \cup k'$;
15 $\theta_b \leftarrow \theta'$;
16 $\mathcal{R}_b \leftarrow \mathcal{R}_b \backslash \{k'\}$;
17 break;
18 $\beta_b \leftarrow$ center of the beamwidth θ_b ;
19 return C_b, θ_b, β_b

The association decision $x_{bkf}^{\nu}(t)$ is then sent to users through feedback (Lines 9 to 12).

Step 3- At the user side, based on the feedback from APs, each user calculates its immediate reward $\Gamma_k(t)$ by

$$\Gamma_k(t) = w_{1k} \sum_{\nu \in \mathcal{I}} c_{1k}^{\nu}(t) + w_{2k} \sum_{\nu \in \mathcal{I}} c_{2k}^{\nu}(t), \tag{4.18}$$

where c^v_{1k}, c^v_{2k} are the rewards for QoS-satisfied and QoS-unsatisfied applications of user k, respectively,

$$c^v_{1k}(t) = \sum_{b\in\mathcal{B}} \sum_{f\in\mathcal{F}_k} \mathbb{I}(x^v_{bkf} = 1, r_{bkf}(t) > R_{kf}) a^v_{bkf}(t) \frac{r_{bkf}(t)}{W_v}, \tag{4.19}$$

$$c^v_{2k}(t) = \begin{cases} -\sum_{b\in\mathcal{B}} \sum_{f\in\mathcal{F}_k} \mathbb{I}(x^v_{bkf}(t) = 1, r_{bkf}(t) < R_{kf}) \times a^v_{bkf}(t) \frac{R_{kf}}{r_{bkf}(t)}, & \text{(4.20a)} \\ -\sum_{b\in\mathcal{B}} \sum_{f\in\mathcal{F}_k} \mathbb{I}(x^v_{bkf}(t) = 0) a^v_{bkf}(t) \frac{R_{kf}}{\hat{r}_{bkf}(t)}. & \text{(4.20b)} \end{cases}$$

In (4.19), the reward is the sum-rate of QoS-satisfied applications for each interface, normalized by each bandwidth W_v. The purpose of this normalization is to guarantee reward-fairness among the two interfaces, as they provide drastically different rate levels. If user k is served by AP b on interface v but at lower rate than R_{kf}, c^v_{2k} is calculated by (4.20a). But if this application is dropped or blocked (in the case of mmWave interface), then c^v_{2k} is calculated by (4.20b), where $\hat{r}_{bkf}(t)$ is estimated by user k. On Sub-6GHz interface, the user can estimate $\hat{r}_{bkf}(t)$ by using (4.1). However, on mmWave interface, as the AP's beamwidth and direction is unknown to the dropped user, we propose to estimate $\hat{r}_{bkf}(t)$ assuming the narrowest beamwidth and LoS direction, and only side-lobe interference ϵ from other APs. Note that, as in [54], users whose applications are dropped, blocked or whose served rates are lower than R_{kf}, will be in outage.

Then based on these rewards, each user k updates the weights Ω^t_k of its DQN at each scheduling time frame t, such that its loss function $\mathcal{L}_k$ is minimized through stochastic gradient descent as in [55], then moves to its new state (actual association/application/interface), with

$$\mathcal{L}_k = \left(Q(s_k, a_k; \Omega^t_k) - (\Gamma_k + \alpha \max_{a'_k} Q(s'_k, a'_k; \Omega^t_k))\right)^2, \tag{4.21}$$

where α is the discount factor (Lines 14 to 16).

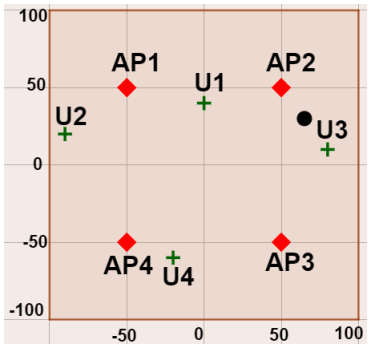


Figure 4.4: Small network

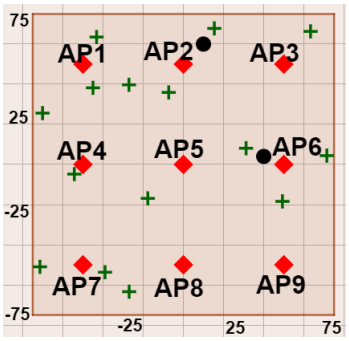


Figure 4.5: Larger network

4.5 Numerical Evaluation

The proposed algorithms are evaluated in two types of networks: firstly, a small network (Fig. 4.4) composed of 4 APs and 4 fixed users, and secondly, a larger network (Fig. 4.5) composed of 9 APs and 15 users uniformly distributed over the whole network area. In both scenarios, each user requires two applications with minimum rate requirements R_{k1} = 100 Mbps, R_{k2} = 1 Mbps. This means that, while the small network is a reasonable scenario for an initial evaluation of the proposed methods, the large network, where 30 applications should be supported at anytime by only 9 APs simultaneously, represents a heavy and challenging environment as in urban hot spot scenarios.. To assess the influence of obstacles, one or two obstructions (black circles) are placed as shown in Figs. 4.4 and 4.5, then removed during two periods, namely from frames 2000 – 4000 and from frames 6000 – 8000.

Block Rayleigh fading channels are assumed, where each channel coefficient remains fixed during a frame, but changes randomly across frames. Users are assumed fixed during each episode T of 10000 frames. In the larger network case, results are averaged over 50 random user positions. We have built our own simulator using Python 2.7. Detailed simulation settings, including path loss models, are presented in Table 4.1.

The DQN (Fig. 4.3) is built with two hidden fully connected layers using Softmax activation function. The number of neural nodes per hidden layer is 16, the memory size is set to 100 and the batch size is 20. Then, learning parameters are set as [30] with α = 0.9, ε = 0.5, and decay factor λ = 0.995. Weights (w_{1k}, w_{2k}) in (4.18) are (0.8, 0.2) as

Table 4.1: Simulation Parameters

Parameter	Description
Transmit Power p_b	5 dBm
Noise power σ_n^2	-169 dBm/Hz
Bandwidth	10 MHz
Channel fading model	Rayleigh fading
Path loss model - Sub-6GHz LoS Path loss model - mmWave NLoS Path loss model - mmWave [56]	$38.5 + 30\log_{10}(d)$ $61.4 + 20\log_{10}(d) + X_\sigma$, $X_\sigma \sim$ Gauss(0; 5.8 dB) $72 + 29.2\log_{10}(d) + X_{\sigma'}$, $X_{\sigma'} \sim$ Gauss(0; 8.7 dB) (d: AP-user distance [m])
Mean packet arrival rate ($\lambda_{kf}\frac{1}{\mu_{kf}}$)	0.1 Mbps
X	3

in Chapter 3.

4.5.1 Benchmark schemes

The following two benchmarks are evaluated:

- **Reference Basic-DQN** (*Ref. Basic-DQN*): This method is similar to the reference QL-based scheme in [54], but translated to integrated networks and using DQN. User state and action are respectively the current association and its desired association between its applications and APs/interfaces. After receiving all requests, APs select users for each interface as in our proposed algorithm, but only a unique user with narrowest beam is supported by each mmWave.

- **Reference Action Elimination (AE)-DQN** (*Ref. AE-DQN*): In this method proposed in [57], the DQN is combined with an Action Elimination Network (AEN) based on linear contextual bandit model, that eliminates invalid actions (such as APs in NLoS) given a specified state. An action is valid if its value given by the AEN with its current state is within a pre-defined confidence ellipsoid. Among a set of valid actions provided by the AEN, with probability $1 - \varepsilon$, the action with highest Q-value given by the DQN is chosen. Further details can be found in [57]. After receiving all requests, procedure at APs is the same as in our proposed algorithm and the users also calculate their rewards by (4.18). Due to

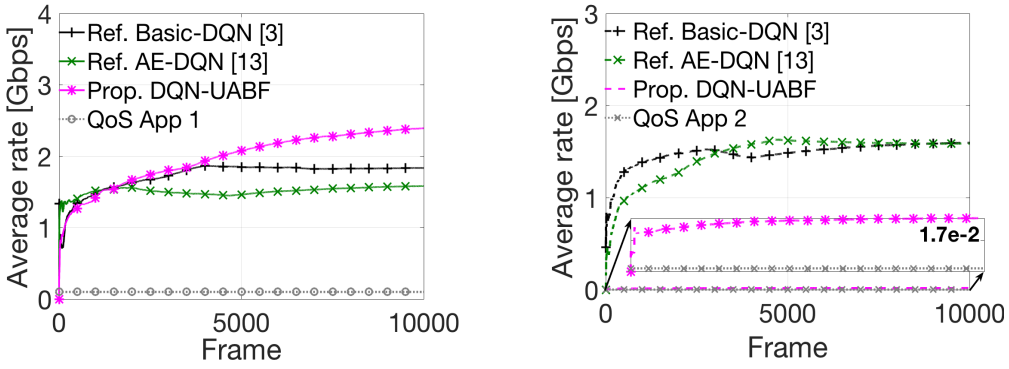
its high computational complexity, this scheme is only evaluated in the small network.

4.5.2 Simulation Results

4.5.2.1 Small network

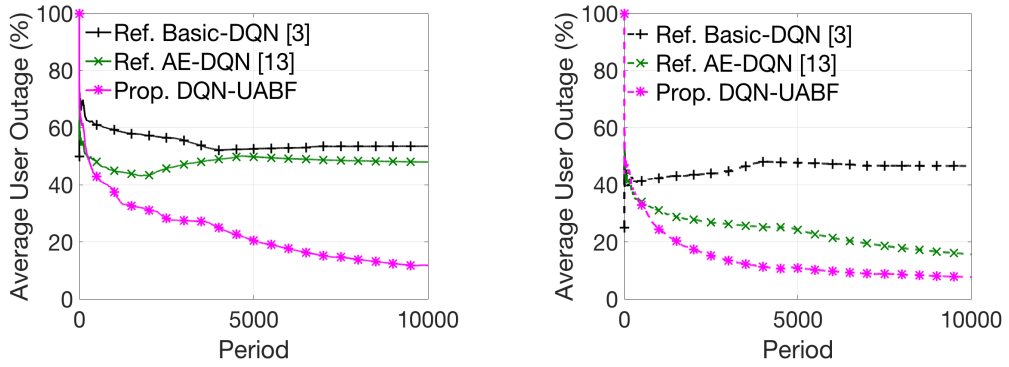
Fig. 4.6 shows the achievable data rate per application averaged over frames. We can see that all algorithms converge well and satisfy QoS requirements. We observe that, while our proposed algorithm outperforms both benchmarks for application 1, they provide much higher data rates for application 2, despite its much lower requirement ($R_{k2} << R_{k1}$). *Prop. DQN-UABF* hence adapts better its allocated rates to each of the QoS levels.

Moreover, we can observe in Fig. 4.7 that *Prop. DQN-UABF* achieves lower user outage probabilities for both applications, i.e., the probability that the QoS target is not met. Indeed, the proposed method achieves a reduction of 80% and 83% for applications 1 and 2 vs. *Ref. Basic-DQN*, and a reduction of 75% and 51% for applications 1 and 2 vs. *Ref. AE-DQN*. In addition, despite obtaining high data rates, *Ref. Basic-DQN* still gets high outage probabilities for both applications, as only a small number of users is allocated with extremely high data rates while the others are dropped. That is, *Ref. Basic-DQN* tends to allocate mmWave interface for both applications, but only a unique user with narrowest beam is supported by each AP. The users in *Ref. AE-DQN* also prefer to request mmWave interface for both applications, but more users are satisfied thanks to the clustering algorithm. However, *Ref. AE-DQN* cannot guarantee a fair QoS satisfaction between applications 1 and 2, with 48% and 16% outage levels, respectively. By contrast, *Prop. DQN-UABF* not only achieves lower outage, but also guarantees outage fairness between both applications, having 12% and 8% outage levels, respectively. This is because the proposed algorithm successfully learns to allocate mmWave band for application 1 and Sub-6GHz band for application 2, thanks to its specific state space and reward design.



(a) Application 1 (b) Application 2

Figure 4.6: The average data rate per application in small network



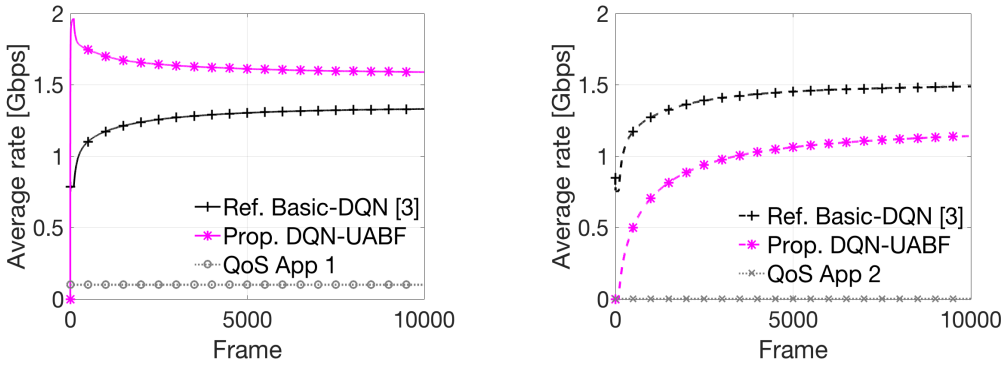
(a) Application 1 (b) Application 2

Figure 4.7: The average user outage per application in small network

4.5.2.2 Larger network

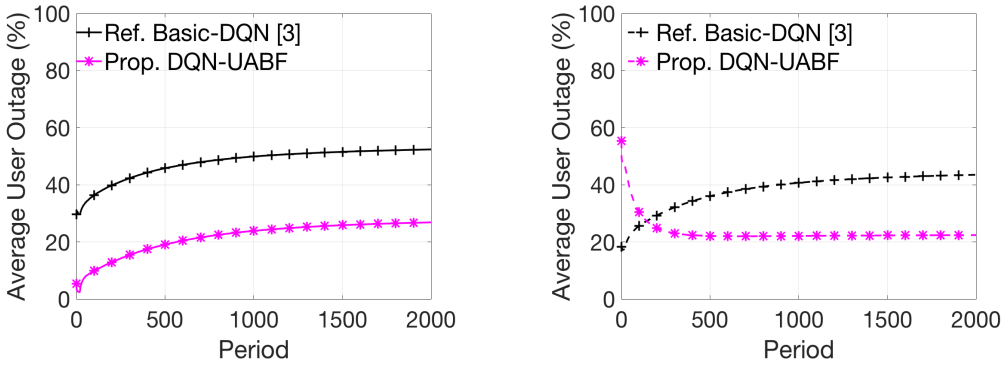
As explained above, in the larger network, the proposed algorithm can be only compared to *Ref. Basic-DQN*. Figure 4.8 shows the data rate for each application averaged over all users and positions. Interestingly, *Ref. Basic-DQN* provides a higher rate for application 2 with lower R_{kf} than for application 1, whereas *Prop. DQN-UABF* enables a tailored rate provision given the level of R_{kf}.

Moreover, although *Ref. Basic-DQN* improves the average rate of both applications over time, its outage probabilities also increase as shown in Fig. 4.9. This is also because *Ref. Basic-DQN* tends to allocate mmWave interface for both applications, and only users with highest utilities are supported. Furthermore, the proposed algorithm



(a) Application 1 (b) Application 2

Figure 4.8: The average data rate per application in larger network

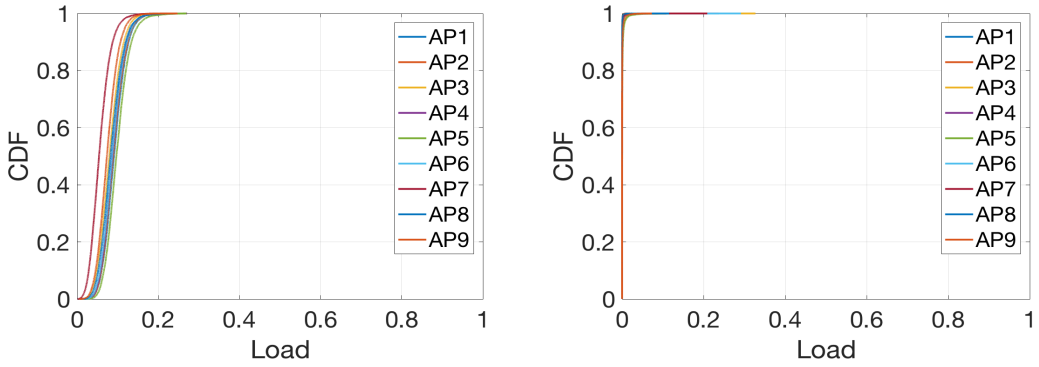


(a) Application 1 (b) Application 2

Figure 4.9: The average user outage per application in larger network

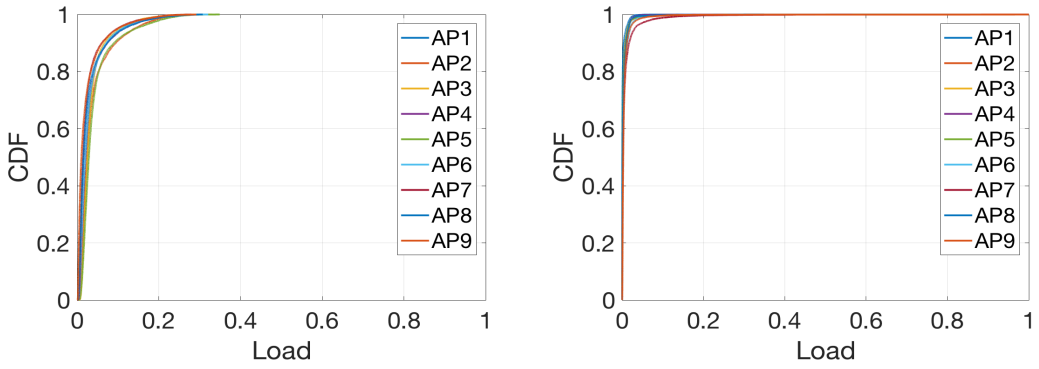
provides lower outage compared to *Ref. Basic-DQN*, namely 45% and 51% reductions for applications 1 and 2, respectively. In addition, the outage probabilities for each application are 45% and 36% for *Ref. Basic-DQN*, vs. 18% and 15% for *Prof. DQN-UABF*. This again proves that our proposed method greatly enhances outage fairness among applications.

Finally, we show the cumulative distribution function (CDF) of the load for each AP/interface in Figs. 4.10 and 4.11. It can be observed that all algorithms largely satisfy the load constraint of each AP and each interface. In particular, for Sub-6GHz interface, the proposed algorithm achieves lower load as well as better load fairness among APs compared to *Ref. Basic-DQN*. On the mmWave interface, the AP load given by *Ref.*



(a) Sub-6GHz interface (b) mmW interface

Figure 4.10: The CDF of AP load by *Ref. Basic DQN*



(a) Sub-6GHz interface (b) mmW interface

Figure 4.11: The CDF of AP load by *Prop. DQN-UABF*

Basic-DQN is a bit lower than that of *Prop. DQN-UABF*. This is due to the fact that *Ref. Basic-DQN* only serves an unique user with narrowest beam, whereas the proposed algorithm can serve more users efficiently, while guaranteeing the load constraint.

4.6 Summary

We have investigated the issue of joint user-to-multiple AP association and beamforming in Sub-6GHz/mmWave integrated networks, whereby a user requiring several applications may be served by several APs and interfaces simultaneously. To address this intricate issue, we have proposed the DQN-based user association and

beamforming method, that enables distributed learning by each user with minimal feedback from local APs. Simulation results have shown the effectiveness of the proposed method against benchmarks, by notably reducing user outage, while improving outage fairness among applications.

5 Energy Efficiency Study of Deep Q-Network based User Association Methods in Sub-6GHz/mmWave Integrated Networks

5.1 Introduction

In this chapter, we focus on the energy efficiency study of the DQN-based user-to-multiple APs association method proposed in the previous chapter, for B5G Sub-6GHz/mmWave integrated networks. We first conduct a comprehensive analysis of the energy consumption for both computation and data access for operating each user's DQN, unlike in previous works.

Based on that, we enhance the proposed DQN-based user association and beam-

forming method in Chapter 4, which enables not only higher energy efficiency, but also to better cope with dynamic environments. Namely, the adaptive ε-greedy strategy is proposed to allow user devices to favor exploration whenever notable changes of its surrounding environment are detected. In addition, to further improve the network performance, instead of making use of the greedy algorithm 4.2 as in Chapter 4, we design a Branch-and-Bound-based algorithm [58] to cope with the features of mmWave bands, for solving user clustering and beamforming at AP side. Finally, the trade-off between achievable network performances and energy costs at user side is also investigated through numerical evaluations.

5.2 Analysis of DQN energy consumption

We analyze the energy consumed for running the proposed DQN-based user association and beamforming (*Prop.DQN-UABF* described in Algorithm 4.1) method at every user device. Power consumption arises from signaling and data transmissions between each user and their requested/serving APs, as well as the power consumed at the user DQN, including:

- P_k^c: the hardware power required to keep user k's device activated,
- P_k^{Tx}: wireless power transmission for downlink/uplink signaling, data transmission and CSI feedback,
- P_k^{proc}: processing power for DQN computation tasks,
- P_k^{data}: power required for data movement, i.e., for accessing user k's memory and to read/write the data required for operating its DQN, such as the DQN weight updates, and experience replay updates.

Here, P_k^c and P_k^{Tx} are fixed. In the sequel, P_k^{proc} and P_k^{data} are analyzed in detail.

5.2.1 Energy consumption for user DQN processing P_k^{proc}

DQN processing includes two major tasks: firstly, forward propagation for obtaining the best action given the current state, and secondly, DQN weight updates through

experience replay [48], which comprises forward and backward propagation, as well as the derivation operation of stochastic gradient descent. These actions require substantial power to perform basic Multiplication-and-Accumulation (MAC) operations depending on the number of neurons and links in the DNN.

We denote by L the number of hidden layers, N_l ($l = 1..N$) the number of neurons in layer l and N_0, N_{L+1} the number of input and output nodes, respectively. S_{batch} is the size of a mini-batch in the experience replay technique. Then, similarly to [44], we express the power consumed for DQN processing as a function of the number of layers and neurons, as

$$P_k^{\text{proc}} = \left(\sum_{l=1}^{L+1} N_{l-1}N_l + N_l\right) P_{\text{unit}} + \left(\sum_{l=1}^{L+1} 2N_{l-1}N_l + 3N_l\right) S_{\text{batch}} P_{\text{unit}}, \tag{5.1}$$

where P_{unit} is the power required for a basic MAC operation, similarly to [2, 42]. In (5.1), the first term is the power consumed for computing the best action which requires a total of $\sum_{l=1}^{L+1} N_{l-1}N_l$ multiplications to feed data from the input layer with N_0 neurons through L hidden layers, producing N_{L+1} values at the output layer. During this process, N_l values at each hidden layer and at the output layer go through the activation function. The second term in (5.1) is the power required for updating the DQN weight, where the DQN needs to replay a mini-batch with S_{batch} experiences in memory. Similarly to [44], each experience requires a forward propagation given its stored state, a backward propagation from the output layer to the input layer, which also performs $\sum_{l=1}^{L+1} N_{l-1}N_l$ multiplications, and calculating the derivative of the loss function (4.21) through N_l real multiplications for each layer l with $l = 1..N + 1$.

5.2.2 Energy consumption for user DQN data movement P_k^{data}

This major energy consuming task had not been considered in previous works such as [44]. The power for data movement is consumed whenever the DQN accesses its stored weights in order to calculate the Q-values for all its output nodes, and to store the data of this new experience into its memory. Similarly, the target DQN used for computing the target Q-values, needs to read and write its experiences and weights in the memory. Given the multi-level memory hierarchy in modern hardwares in order to reduce delay and energy consumption, energy for data movement is not only a

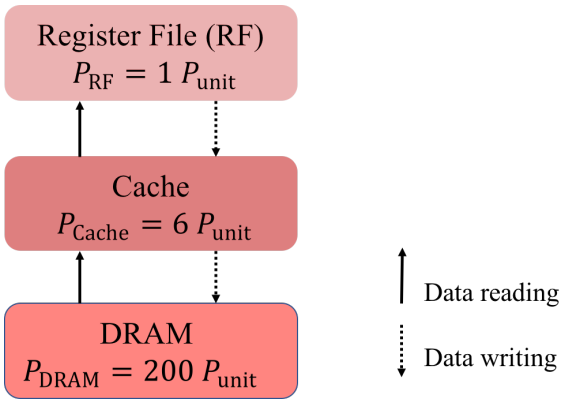


Figure 5.1: A memory hierarchy and data movement flow [2]

function of the number of layers and neurons in the DQN, but also depends on the data storage location as well as the memory hierarchy.

Without loss of generality, we here consider a memory hierarchy with three levels as in [2] and illustrated in Fig. 5.1: Register File (RF), Cache (Buffer), and Dynamic Random Access Memory (DRAM), where the energy costs to access data in the DRAM and in the Cache are respectively 200 times and 6 times more than that in the RF, which represents the unit energy cost P_{unit} in terms of the energy for a MAC on a commercial 65nm process [2]. That is, DRAM can store much larger amounts of data compared to Cache and RF, but at the cost of much higher energy consumption. With the current hardware technology, the amounts of data that can be stored at each level are estimated to be 0.5-1 kilobytes (kB) in RF, 100-500 kB in Cache [2]. Hence, any data of more than 500 kB needs to be stored in DRAM. In the sequel, we consider two typical scenarios, depending on the amounts of data to be stored: the "heavy" case where the weights of both DQN and target DQN are stored in DRAM, and experiences in Cache, and the second "light" case, where all weights and experiences are stored in Cache.

In the first scenario, during forward propagation, the DQN loads the weights from the DRAM and the input node values from the Cache to perform matrix multiplications, and stores its experience in the Cache. During DQN's weight updating process, a mini-batch of experiences is loaded from the Cache, then the weights of both DQN and target DQN in DRAM, as well as the input node values in Cache are also fed for matrix multiplication. The derivation results are stored in Cache, before writing the new weights to DRAM. Namely, to calculate a matrix multiplication in forward and

backward propagation between input nodes at layer $l-1$ and weights of size $N_{l-1}N_l$ at layer l ($l = 1..L+1$), the DQN needs to access the DRAM $N_{l-1}N_l$ times to load these weights to the Cache. Then, these weights and the input of layer $l-1$ are loaded to Arithmetic Logic Units (ALU) through Cache and RF, each requiring $N_{l-1}N_l + N_{l-1}$ accesses. At ALU, $N_{l-1}N_l$ MAC operations are performed and their results are written in the RF. Next, the RF is accessed again to read the accumulation results for computing N_l final results, which are written in Cache. In addition, during backward propagation, after the N_l final results at each layer l are obtained, they are stored and then loaded from the RF to calculate derivatives by N_l real multiplications before being stored in Cache. Finally, these derivatives and corresponding weights are again fed to the RF, then to ALU to update the weights, before saving the new weights into DRAM. From the above, the consumed power for DQN data movement P_k^{data} in this scenario is given by

$$P_k^{\text{data}} = \underbrace{\sum_{l=1}^{L+1} P_{\text{MM}^l} + P_{\text{Cache}}}_{\text{Power in forward propagation}} + \Bigg(\underbrace{S_{\text{batch}} \times P_{\text{Cache}}}_{\substack{\text{Power for} \\ \text{loading experiences}}} + \underbrace{2\sum_{l=1}^{L+1} P_{\text{MM}}^l}_{\substack{\text{Power in forward} \\ \text{\& backward propagation}}} + \underbrace{2N_l \times P_{\text{RF}}}_{\text{Power for derivatives}} + \underbrace{(P_{\text{Cache}} + P_{\text{RF}} + P_{\text{DRAM}}) \sum_{l=1}^{L+1} N_{l-1}N_l}_{\text{Power for updating weights}} \Bigg), \quad (5.2)$$

where P_{DRAM}, P_{Cache} and P_{RF} are the consumed energy for accessing DRAM, Cache and RF, respectively. Their units are normalized by P_{unit}, with $P_{\text{RF}} = P_{\text{unit}}$ as in [2]. P_{MM}^l is the total power consumed for a matrix multiplication at layer l, which, from the above explanation, is computed by

$$P_{\text{MM}}^l = (N_{l-1}N_l)P_{\text{DRAM}} + (N_{l-1}N_l + N_{l-1} + N_l)P_{\text{Cache}} + (2N_{l-1}N_l + N_{l-1})P_{\text{RF}}. \quad (5.3)$$

In the second "light" scenario of data storage, all processes relating to DRAM are, instead, performed in the Cache. Therefore, similarly as above, we can compute P_k^{data} by

$$P_k^{\text{data}} = \underbrace{\sum_{l=1}^{L+1} P_{\text{MM}^l} + P_{\text{Cache}}}_{\text{Power in forward propagation}} + \Bigg(\underbrace{S_{\text{batch}} \times P_{\text{Cache}}}_{\substack{\text{Power for} \\ \text{loading experiences}}} + \underbrace{2 \sum_{l=1}^{L+1} P_{\text{MM}}^l}_{\substack{\text{Power in forward} \\ \& \text{ backward propagation}}} + \underbrace{2N_l \times P_{\text{RF}}}_{\text{Power for derivatives}} + \underbrace{(2P_{\text{Cache}} + P_{\text{RF}}) \sum_{l=1}^{L+1} N_{l-1} N_l}_{\text{Power for updating weights}} \Bigg), \quad (5.4)$$

where

$$P_{\text{MM}}^l = (N_{l-1}N_l)P_{\text{Cache}} + (N_{l-1}N_l + N_{l-1} + N_l)P_{\text{Cache}} + (2N_{l-1}N_l + N_{l-1})P_{\text{RF}}. \quad (5.5)$$

Based on the system and power consumption models above, we define the energy efficiency for each user k as the ratio between the sum of converged data rates for all its requested applications and its total power consumption, given by

$$EE_k = \frac{\sum_{b \in \mathcal{B}} \sum_{f \in \mathcal{F}_k} r_{bkf}}{P_k^c + P_k^{\text{Tx}} + P_k^{\text{proc}} + P_k^{\text{data}}}. \quad (5.6)$$

As such, in order to enhance the EE, each user needs to improve its achieved rate over frames while minimizing its energy consumption. Given a specific structure of DQN with a fixed number of hidden layers and hidden neural nodes, as well as a fixed number of possible actions (which can be kept in a low value for energy reduction as in our proposed scheme explained in next section), the consumed power of user can be considered as a constant. Therefore, improving EE means achieving higher data rates over frames, thereby also reducing the user outage probability. In addition, by the latency definition as in Eq. (3.3) which is a ratio between the file size of the application and its data rate, EE enhancement also reduce the delay time of serving the required application of the user.

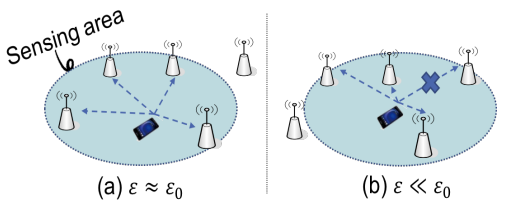


Figure 5.2: Original ε-greedy)
(a) $\varepsilon \approx \varepsilon_0$: explore all APs in sensing area,
(b) $\varepsilon \ll \varepsilon_0$: may not explore new APs in sensing area

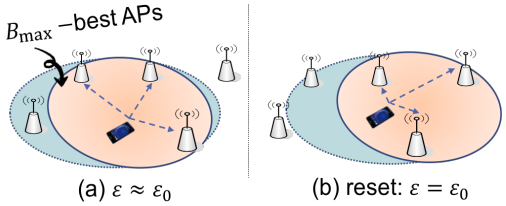


Figure 5.3: Adaptive ε-greedy
(a) $\varepsilon \approx \varepsilon_0$: explore all APs in $B_{\max}$-best APs,
(b) Detect new APs $B_{\max}$-best APs: reset $\varepsilon = \varepsilon_0$

5.3 The proposed adaptive ε-greedy DQN

Considering the same problem defined in section 4.4, we here enhance the proposed DQN-based user association and beamforming (*Prop. DQN-UABF*) method described in Chapter 4, which enables to cope with the dynamic network environment and most importantly, to improve energy efficiency.

Namely, in this enhancement, instead of applying the original ε-greedy method of reference [48] as in *Prop. DQN-UABF*, we design a new strategy coined as adaptive ε-greedy for controlling the ratios of exploration and exploitation, which enables to better cope with dynamic environments. This is because in the original ε-greedy method (depicted in Fig. 5.2), mobile users firstly explore the environment with a high probability ε (Fig. 5.2(a)) and exploit the actions maximizing their approximated Q-function with a small probability 1-ε. As ε decreases over the following iterations, only exploitation is performed even when the environment changes drastically (Fig. 5.2(b), in which case the best actions computed by their DQNs may not be suitable anymore. The proposed adaptive ε-greedy DQN method is hence designed to allow users to perform ad-hoc explorations as depicted in Fig. 5.3, whenever significant changes are detected in their surrounding environments, in particular in terms of wireless channel quality and blockage occurrences. The details of the proposed DQN-based User Association with the adaptive ε-greedy method are given in Algorithm 5.1, which can be described as follows.

Step 1- Whenever a change is detected, user k updates its own list of $B_{\max}$-best APs among all APs in its sensing area, based on estimated channel qualities and blockages

Algorithm 5.1: Proposed DQN-based Association and Beamforming with Adaptive ε-greedy Method

1 **for** *each user* $k \in \mathcal{K}$ **do**
2 $bestAPSet_k[0] \leftarrow B_{\max}$-best APs at time $t = 0$;
3 Initialize DQN Q with random weight values Ω_k;
4 Initialize state s_k based on $bestAPSet_k$;
5 $\varepsilon_k \leftarrow \varepsilon_0$;
6 **for** $t = 1, 2, \ldots, T$ **do**
7 **for** *each user* $k \in \mathcal{K}$ **do**
8 $bestAPSet_k[t] \leftarrow B_{\max}$-best APs at time t;
9 **if** $bestAPSet_k[t] \neq bestAPSet_k[t-1]$ **then** $\varepsilon_k \leftarrow \varepsilon_0$;
10 **else** $\varepsilon_k \leftarrow \varepsilon_k \times \lambda$;
11 **if** *random number* $p < \varepsilon_k$ **then** Select random action a_k;
12 **else** Select action a_k with max $Q(s_k, a_k; \Omega_k^t)$;
13 **for** *each AP* $b \in \mathcal{B}$ **do**
14 Select users for Sub-6GHz, mmWave by Branch-and Bound algorithm and feed back to users;
15 **for** *each user* $k \in \mathcal{K}$ **do**
16 Calculate reward of action a_k by (4.18) ;
17 Update Ω_k^t by (4.21);
18 Move to the new state $s_k \leftarrow s'_k$;

over its links to those APs, and resets its probability of exploration ε to the initial value (Fig. 5.3(b)), allowing exploration of those new APs and interfaces. Otherwise, ε is decreased by a decay factor λ (Fig. 5.3(a)). Then, with probability 1-ε, user k selects action $a_k(t)$ from the output of its DQN based on its current state, and sends its request $a_k(t)$ to the desired (AP, interface) for each application f in $\mathcal{F}_k$ (Lines 7–14).

Step 2- After receiving all user requests, APs perform different user association and beamforming strategies for each interface (Lines 15–17).

We denote by $\mathcal{K}_b^v$ the set of users requesting association to AP b with interface v. Given interface v, the following problem needs to be handled by AP b, i.e, AP b's sum-rate maximization under its load constraint,

$$\max_{x^{\nu}_{bkf}(t),\theta_{bk}(t)} \sum_{k\in\mathcal{K}^{\nu}_{b}} \sum_{f\in\mathcal{F}_k} x^{\nu}_{bkf}(t) r_{bkf}(t) \tag{5.7}$$

$$\text{s.t. } \Phi^{\nu}_{b}(t) = \sum_{k\in\mathcal{K}^{\nu}_{b}} \sum_{f\in\mathcal{F}_k} x^{\nu}_{bkf}(t)\phi_{bkf}(t) \leq 1. \tag{5.7a}$$

It may be observed that regarding variable x^{ν}_{bkf}, (5.7) is a 0-1 Knapsack problem, but with real valued weight items. Instead of applying a greedy method as in *Prop. DQN-UABF* to solve (5.7), we here design a method based on Branch-and-Bound [59], but where the upper bound value at each branching node is calculated based on the greedy solution in *Prop. DQN-UABF*, thereby guaranteeing a better solution for x^{ν}_{bkf} and θ_{bk} which enables higher network performances. The psedou code of this Branch-and-Bound-based method for solving (5.7) are given in Algorithm 5.2 and explained in details in the next section.

After solving (5.7) for each interface ν, APs send their association decision to their users through feedback.

Step 3- This step is the same as Step 3 in *Prop. DQN-UABF* explained in Chapter 4.

5.4 Proposed Branch-and-Bound-based algorithm for solving user clustering and beamforming

The idea of the B&B algorithm is to partition all feasible solutions into sub-classes with their corresponding upper bound performance values. Then, the partitioning process for each sub-class, called branching, is pursued until each feasible solution belongs to exactly one smaller sub-class, and stopped if its upper bound value is smaller than that of other sub-classes. A feasible solution is reached if the value of its objective function is greater than those of all sub-classes, and than those of all previously obtained solutions.

Based on this, the details of the proposed B&B-based optimization method for solving (5.7) are given as follows.

Node definition: Each node of the B&B tree is given as a tuple $(\mathcal{A}, \mathcal{R}, C, U)$, where $\mathcal{A}, \mathcal{R}, C$ are the sets of accepted devices, rejected devices and candidate devices,

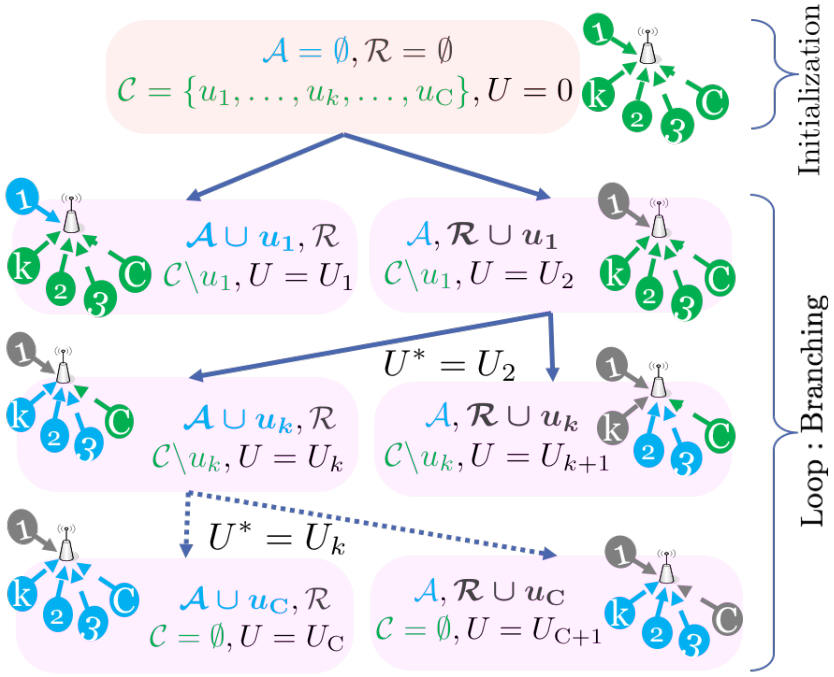


Figure 5.4: Illustration of the proposed Branch-and-Bound based algorithm for solving (5.7)

respectively. The term U gives the upper bound value of the node.

Upper bound calculation: The key step of the B&B algorithm is to calculate U for each node. In this study, we apply the greedy algorithm described in Chapter 4 for upper bound calculation, which guarantees the low complexity.

Next, the pseudo-code of the proposed B&B-based algorithm for solving (5.7) is presented in Algorithm 5.2 and illustrated in Fig. 5.4. Firstly, we calculate rate r_{bk} and load contribution ϕ_{bk} of each device $k \in \mathcal{K}$ given the smallest beamwidth and the closest beam direction in $\mathbb{D}_\beta$ to the LoS direction of AP b and device k. Next, we sort the devices in decreasing order of $\frac{r_{bk}}{\phi_{bk}}$ and initialize the first node n_0 with empty sets $\mathcal{A}$, $\mathcal{R}$, and the candidates' set $\mathcal{C}$ with all ordered devices, while the node's upper bound U is set to 0 (Lines 1–4). Then, the branching process is performed as follows. Among the nodes that are not branched yet, we select the one having the highest U, denoted as n^*, and branch it. Namely, we branch n^* on the first device in its candidate set $\mathcal{C}^*$ into two nodes (sub-classes): one is to accept this device, the other is to reject it, and calculate their respective U values by the greedy algorithm in *Prop. DQN-UABF* (Lines 6–9). This

process is iterated until no candidate remains in the selected node, i.e., all devices are already considered to be accepted or rejected (Line 11). Finally, the node for which $C = \emptyset$, and with the highest U value provides the solution to Problem (5.7). We set $x_{bk} = 1$ for all devices $k \in \mathcal{A}^*$, β_b to the direction closest to the bisection direction of angle $\theta_{\text{RL}}^{(\mathcal{A}^*)}$ and θ_b to the smallest beamwidth covering all devices in $\mathcal{A}^*$ (Lines 13-15).

Algorithm 5.2: Proposed Branch-and-Bound based Algorithm for solving (5.7)

// Initialization
1 **for** *device* $k \in \mathcal{K}$ **do**
2 Calculate r_{bk}, ϕ_{bk} by (4.8), (4.9) respectively with $\theta_b = \theta_{\min}$, $\beta_b = \arg\min_{\beta \in \mathbb{D}_\beta}\{|\beta - \beta_{bk}^{\text{LoS}}|\}$;
3 $C_0 \leftarrow$ Order devices in decreasing order of $\frac{r_{bk}}{\phi_{bk}}$;
4 Make first node $n_0 \leftarrow (\mathcal{A}_0 = \emptyset, \mathcal{R}_0 = \emptyset, C_0, U_0 = 0)$;
// Do branching
5 **while** *true* **do**
6 Find unbranched node $n_j = (\mathcal{A}_j, \mathcal{R}_j, C_j, U_j = U^*)$ with highest U^*;
7 **if** $C_j \neq \emptyset$ **then**
8 Branch n_j on the first device k in C_j into two sub-nodes: $n_{j1} \leftarrow (\mathcal{A}_j \cup k, \mathcal{R}_j, C_j \backslash k, U_{j1})$, $n_{j2} \leftarrow (\mathcal{A}_j, \mathcal{R}_j \cup k, C_j \backslash k, U_{j2})$;
9 Calculate U_{j1}, U_{j2} for n_{j1}, n_{j2} respectively by the greedy algorithm in *Prop. DQN-UABF*;
10 **else**
11 break // Solution found
12 Optimal node $n^* \leftarrow (\mathcal{A}^*, \mathcal{R}^*, C^* = \emptyset, U^*)$: the node with highest U;
13 Set $x_{bk} = 1, \forall k \in \mathcal{A}^*$;
14 $\beta_b^* \leftarrow \arg\min_{\beta \in \mathbb{D}_\beta}\{|\beta - \text{direction of bisection of } \theta_{\text{RL}}^{(\mathcal{A}^*)}|\}$;
15 $\theta_b^* \leftarrow \arg\min_\theta\{\theta \in \mathbb{D}_\theta \text{ covering all devices } \in \mathcal{A}^*\}$;
16 Return $x_{bk}, \beta_b^*, \theta_b^*$;

5.5 Investigation of the trade-off between network performances and energy costs at user side

5.5.1 Simulation settings

We investigate the trade-off between the achievable network performances and energy costs at user side for the proposed algorithms through numerical simulations. Namely, the proposed methods are evaluated in a network composed of 9 APs and 5 fixed users as depicted in Fig. 5.5, where several users such as 1, 2 and 5 are in the center of the network, whereas others are in the edge. This scenario allows us to evaluate the effect of users' channel qualities given their positions, on the proposed algorithm's performance. Among them, user 3's link to AP 1 is obstructed by an obstacle (black circle) which is removed during two periods, namely from frames 2000 – 4000 and from frames 6000 – 8000, which is to assess the behavior of the ε-greedy strategy in exploring new APs as the environment changes. Furthermore, all APs located in the LoS direction of any user-AP link are considered as obstacles, resulting into NLoS links in the mmWave band. Then, to assess the effectiveness of the proposed algorithm in dynamic environment, user 3 will be moving following the green path with Pedestrian speed (i.e., 1.15m/s [58]), while other users will remain fixed as shown in Fig. 5.6.

In both static and dynamic scenarios, each user requires four applications: two downlink (DL) applications with different minimum rate QoS: R_{DL1} = 100Mbps, R_{DL2} = 1Mbps; and two uplink (UL) applications with minimum rate QoS R_{UL1} = 10Mbps, R_{UL2} = 0.1Mbps.

For the fixed users, block Rayleigh fading channels are assumed, where each channel coefficient remains fixed during each scheduling frame of 1 ms, but changes randomly across frames. For the moving user 3, large scale fading parameters are assumed fixed over each period of 10 frames, while small scale fading still changes randomly every frame. The total number of simulated frames is 10000. We built our own simulator using Python 2.7. Detailed simulation settings are given in Table 5.1 where the transmit powers are considered to be fixed regardless to AP-user distance, which will be optimized in the future work.

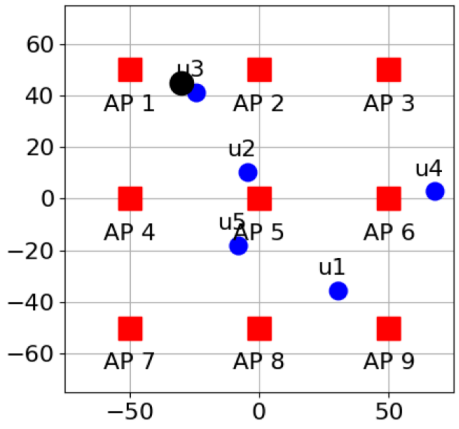


Figure 5.5: Static scenario

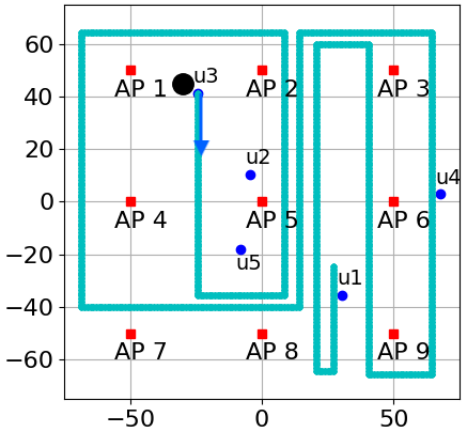


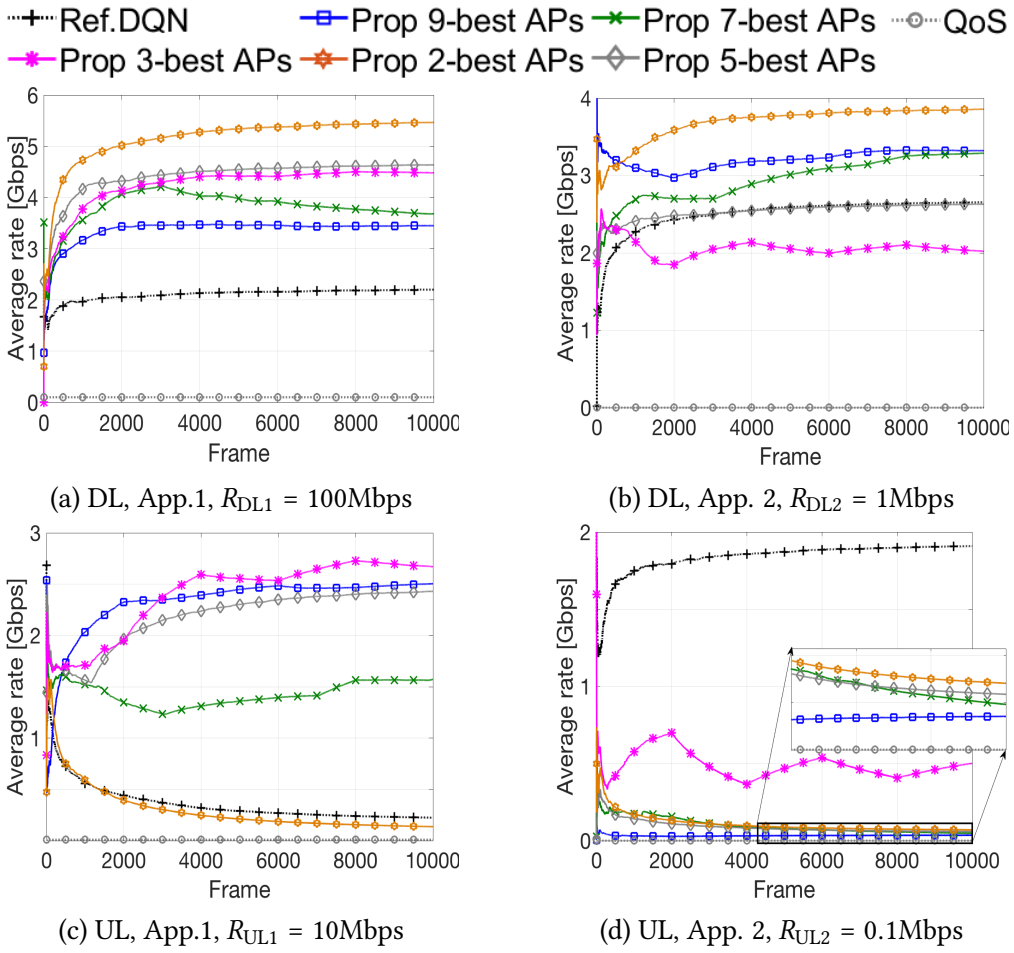
Figure 5.6: Dynamic scenario

Table 5.1: Simulation Parameters

Parameter	Description
DL/UL transmit power ($p_{bkf}^{\mathrm{DL},\nu}$, $p_{bkf}^{\mathrm{UL},\nu}$)	(5, 1.5) dBm
Noise power at user, AP sides	-169 dBm/Hz, -174 dBm/Hz
Sub-6GHz, mmWave Bandwidth	10 MHz, 1 GHz
Small-scale fading model	Rayleigh fading
Path loss model - Sub-6GHz LoS Path loss model - mmWave NLoS Path loss model - mmWave	$38.5 + 30\log_{10}(d)$ $61.4 + 20\log_{10}(d) + X_{\sigma}$, $X_{\sigma} \sim$ Gauss(0; 5.8 dB) $72 + 29.2\log_{10}(d) + X_{\sigma'}$, $X_{\sigma'} \sim$ Gauss(0; 8.7 dB) (d: AP-user distance [m])
Mean packet arrival rate ($m_{kf}\frac{1}{\mu_{kf}}$)	0.1 Mbps

Since user devices are fundamentally battery-limited, we assume the simplest yet effective DQN which is built with two fully connected hidden layers using Softmax activation function. The number of neural nodes per hidden layer is 16, the memory size is set to 100 and the batch size is 20. Then, learning parameters are set as the same proposed DQN-based methods in previous chapters with $\gamma = 0.9$, $\varepsilon = 0.5$, and decay factor $\lambda = 0.995$. Weights (w_{1k}, w_{2k}) in (4.18) are (0.5, 0.5). Moreover, P_k^c and P_k^{Tx} are fixed to 10 mW and 3 mW, while P_{unit} is set to 1 mW as in [44].

We compare the proposed algorithm with adaptive ε value to the reference algorithm *Ref. DQN* where each user exploits a similar DQN to explore all APs, i.e., $B_{\max}$ is fixed to 9, but can only be served by the interfaces of a unique AP simultaneously, as in



(a) DL, App.1, R_{DL1} = 100Mbps

(b) DL, App. 2, R_{DL2} = 1Mbps

(c) UL, App.1, R_{UL1} = 10Mbps

(d) UL, App. 2, R_{UL2} = 0.1Mbps

Figure 5.7: Average data rate over time frames, per application, static scenario

existing systems. To simplify, the proposed algorithm with adaptive ε-greedy will be denoted as *Prop* $B_{\max}$*-best APs*, where $B_{\max}$ refers to the number of candidate APs that are explored simultaneously. That is, by focusing on a small number of APs which provide the best SNR at each frame, users are expected to reduce their DQN power consumption, but should also guarantee a good performance. To clarify this trade-off, we investigate the cases of $B_{\max}$ = 9, 7, 5, 3 and 2 best candidate APs.

5.5.2 Static Scenario

In Fig. 5.7, we present the average data rate evolution over frames per application for the proposed and reference algorithms. The total sum-rates of all applications and

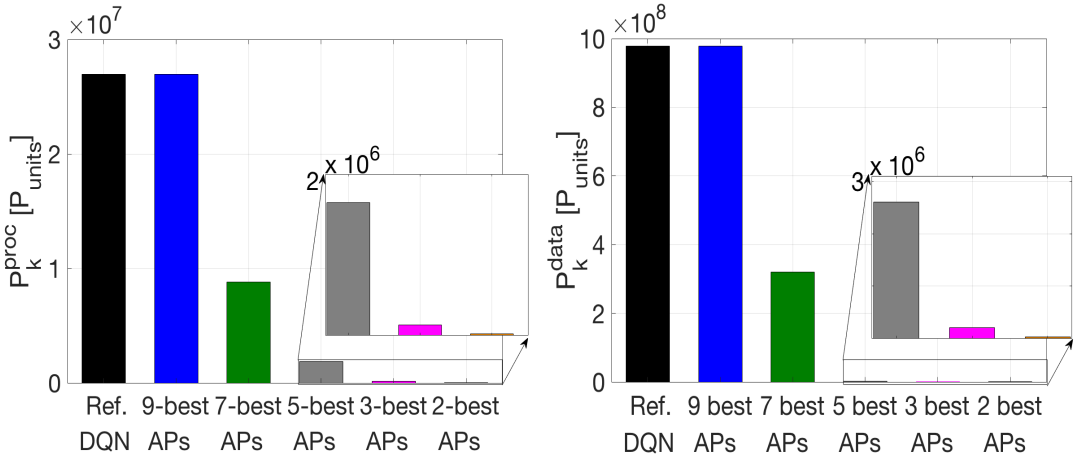
Table 5.2: Average sum-rate [Gbps], outage [%], total power consumption [W] and EE [Mbits/J], Static scenario

Algorithms	Sum-rate	Outage	Total consumed power	EE
Ref. DQN	6.99	25.8	1005.1	0.007
Prop 9-best APs	9.31	3.09	1005.1	0.009
Prop 7-best APs	8.59	3.17	329.0	0.026
Prop 5-best APs	**9.76**	7.72	4.5	2.17
Prop 3-best APs	9.67	**1.65**	0.3	32.2
Prop 2-best APs	9.53	1.98	**0.04**	**238.3**

average outage are given in Table 5.2. We can observe that all algorithms converge well and satisfy the QoS requirement for each application. All proposed methods outperform the reference one in terms of average sum-rate and outage. Namely, despite exploring the same number of APs, *Prop* 9*-best APs* achieves 1.3 times higher sum-rate while providing 8.3 times lower outage probability compared to *Ref. DQN*.

Furthermore, we show the power consumed for user DQN processing and data movement given by *Ref. DQN* and the proposed algorithms with all considered values of $B_{\max}$ in Fig. 5.8. Then, their total consumed power, including the four elements described in Section 5.2, and energy efficiency are also given in Table 5.2. Given $B_{\max}$-best APs and two interfaces (Sub-6GHz and mmWave) per AP, each user may explore a total number of 4 (applications)-permutations of $2B_{\max}$ APs/interfaces. Then for *Ref. DQN*, *Prop* 9*-best APs* and *Prop* 7*-best APs*, the number of possible actions becomes huge, resulting in storing the weights of both DQN and target DQN in DRAM and thereby consuming the data movement power in Eq. (5.2). This is why their consumed energy is extremely high as shown in Fig. 5.8. On the contrary, with $B_{\max} = 5, 3$ or 2-best APs, each user needs to explore only 4-permutations of 10, 6 or 4 APs/interfaces respectively, which enables both DQN and target DQN weights to be stored in the Cache, for the data movement power in (5.4).

In more details, compared to *Prop* 7*-best APs*, *Prop* 9*-best APs* gets higher data rate for the same outage level, but at the cost of higher energy consumption as two more APs are explored every frame. On the contrary, when users request only 5 best APs, the outage level is increased as more conflicts occur among neighboring users. However, *Prop* 3*-best APs* results into a lower outage despite the reduced number of



(a) Power for DQN processing (b) Power for data movement

Figure 5.8: Power consumption of user DQN processing and data movement

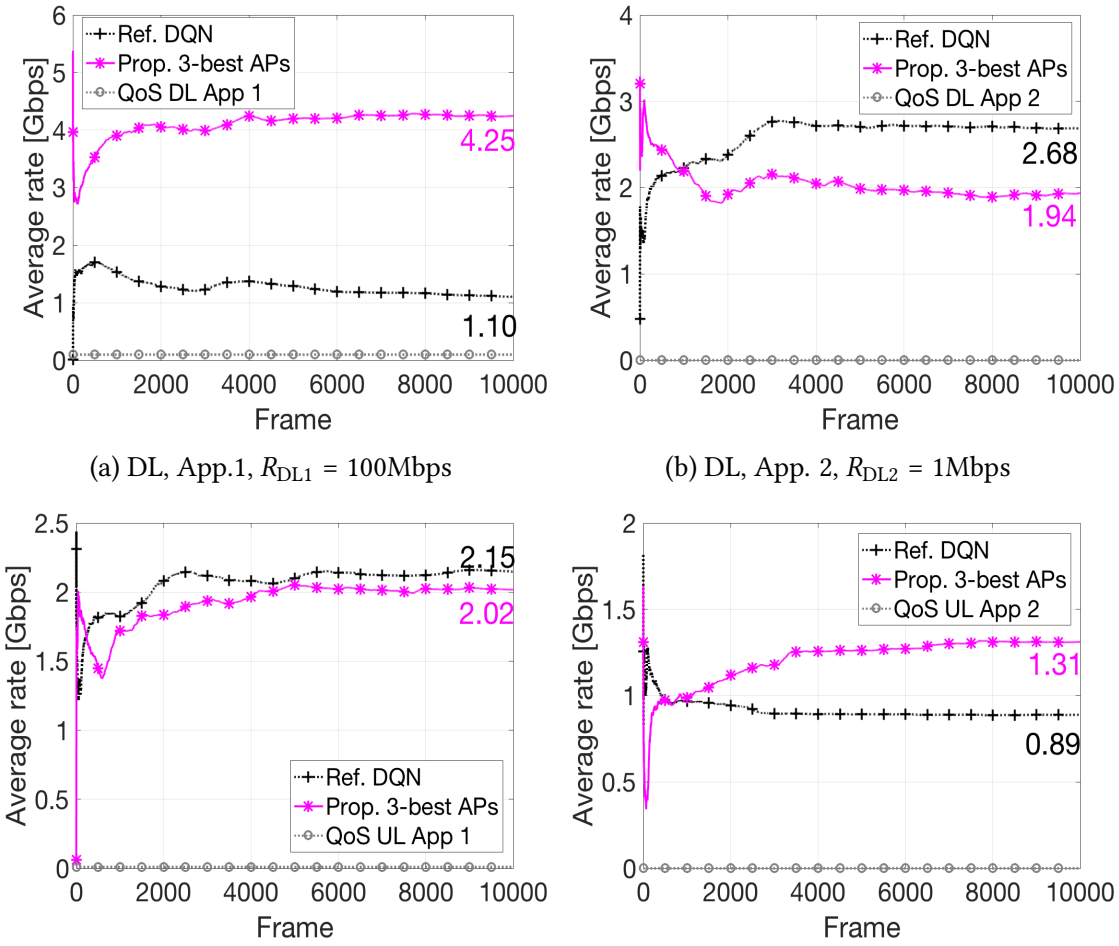
requested APs. This is because, by exploring only the 3 best APs, each user tends to request different sets of APs, thereby decreasing their conflicts. However, with *Prop 2-best APs*, each user needs to systematically request those 4 interfaces for their 4 applications, leaving no room for improvement. That is why *Prop 2-best APs* suffers from higher outage compared to *Prop 3-best APs*.

Overall, the proposed method with $B_{\max} = 2$ or 3 provide the lowest outage levels and approach the highest sum-rate, for a very low total power consumption as shown in Table 5.2. Namely, under this scenario, $B_{\max} = 3$ or 2 result into the best trade-offs between system performance and energy costs, the former for outage minimization, and the latter, for energy efficiency maximization due to its minimal power consumption. Based on this observation, we will evaluate the proposed algorithm with $B_{\max} = 3$ best APs in the dynamic scenario.

5.5.3 Dynamic Scenario

In Fig. 5.9, we show the average data rate evolution over frames per application for *Prop. 3-best APs* and *Ref. DQN*. It can be observed that both algorithms converge well even in the dynamic case. However, after convergence at about frame 3000, the reference method discloses a decreasing tendency for most applications, except for UL

App. 1, whereas *Prop. 3-best APs* keeps increasing the average data rate for DL App. 1 and UL App. 2. The rates for DL App. 2 and UL App. 1 achieved by the proposed algorithm have a slight decrease during frames 5000–8000, but increase again thereafter. These behaviors can be explained by analyzing the actual AP association of moving user 3, shown in Fig. 5.10. For *Ref. DQN*, after 3000 frames, user 3 tends to request AP 4 on both interfaces for DL App. 1 and UL App. 2, and to request AP1-mW for DL App. 2, AP2-mW for UL App. 1. This would be a good behavior if user 3 were fixed as in the previous scenario where user 3 was always close to these APs. However when moving, the connections with these APs are no longer suitable for user 3, especially as it moves out of the coverage areas of APs 1, 2 and 4. Moreover, as the exploration probability ε is not adapted under new environments, user 3 is unable to learn to request other APs, resulting in lower data rates over time. On the contrary, in Fig. 5.10(b), by resetting ε whenever new candidate best APs are detected, the proposed algorithm allows moving user 3 to request these new APs along with its mobility pattern, thereby enhancing the system performance. In addition, we can observe that at the beginning, moving user 3 changes its association requests very often, according to the changes of best APs. However, after about 3000 frames, although the set of best APs is still updated frequently, user 3 can quickly reach a stable association request among these best APs, as indicated by the presence of horizontal lines.



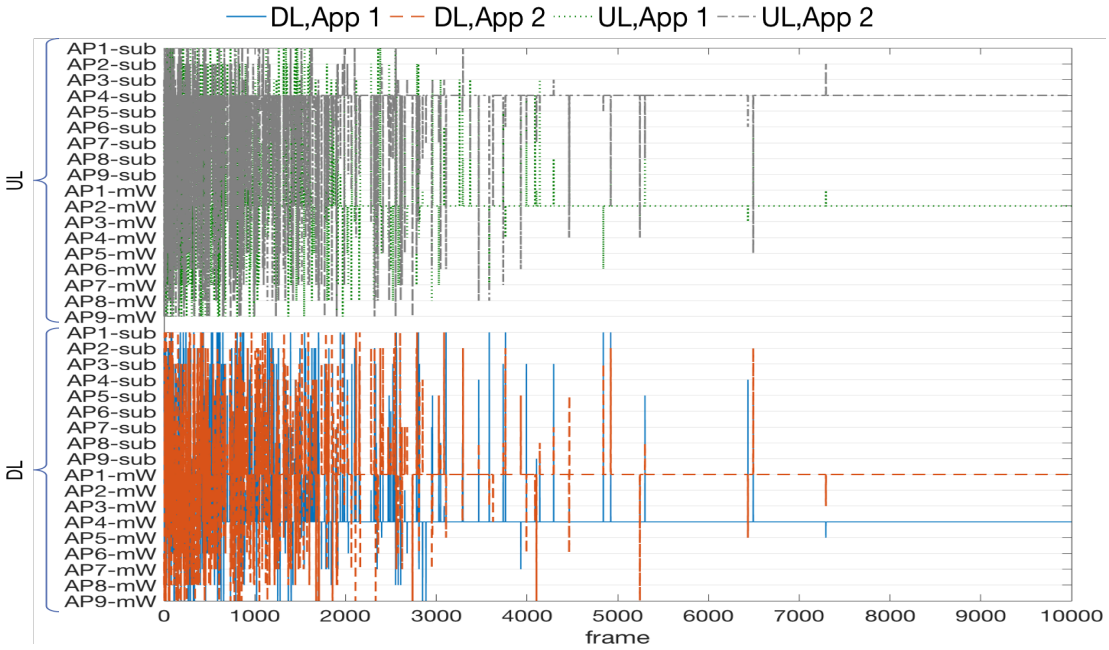
(a) DL, App.1, R_{DL1} = 100Mbps

(b) DL, App. 2, R_{DL2} = 1Mbps

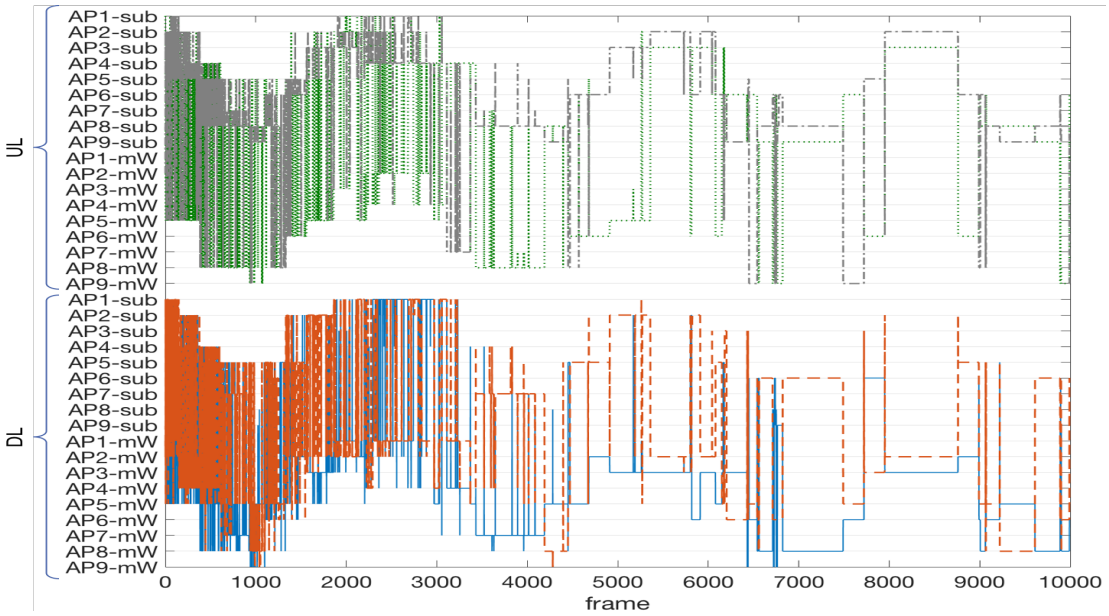
(c) UL, App. 1, R_{UL1} = 10Mbps

(d) UL, App. 2, R_{UL2} = 0.1Mbps

Figure 5.9: Average data rate over time frames, per application, dynamic scenario



(a) Ref. DQN



(b) Prop. 3-best APs

Figure 5.10: Moving user 3's AP and interface association requests over time frames, dynamic environment

Table 5.3: Average sum-rate [Gbps], outage [%], total power consumption [W] and EE [Mbits/J], Dynamic Scenario

Algorithms	Sum-rate	Outage	Total consumed power	EE
Ref. DQN	6.82	35.1	1005.1	0.007
Prop 3-best APs	**9.52**	**6.23**	**0.3**	**31.7**

Finally, the total sum-rate of all applications and average outage in this dynamic scenario are presented in Table 5.3. As expected, the proposed algorithm with 3-best APs outperforms the reference one in all aspects. Namely, *Prop. 3-best AP* gains 40% higher sum-rate, while guaranteeing 80% lower outage probability than *Ref. DQN*. Notably, the proposed algorithm considerably improves the energy efficiency.

5.6 Summary

In this chapter, the overall energy consumption at the user device for running its DQN has been analyzed in details. Based on that, we have introduced two enhancements, namely, the adaptive ε-greedy policy and Branch-and-Bound based user association and beamforming, for the proposed DQN-based association and beamforming algorithm presented in Chapter 4. These enhancements improve not only the network performance, but also the energy efficiency, especially in dynamic environments. Numerical results show that, thanks to the adaptive ε-greedy policy and by adequately setting the number of explored APs $B_{\max}$, the proposed method enables to strike a balanced trade-off between network sum-rate, QoS satisfaction of diverse applications, as well as user energy consumption.

6

Risk-Averse Reinforcement Learning for Reliability Enhancement in Sub-6GHz/mmWave Integrated Networks

6.1 Introduction

As discussed in Chapter 1, reliability is another key requirement of 5G, which will become increasingly important in B5G systems. So far, many approaches have been taken in the literature to enhance the reliability of wireless communications as mentioned in Chapter 2, in particular in the context of massive IoT connectivity.

Recently, key findings within the machine learning community in the field of Risk-Sensitive Reinforcement Learning (RSRL) have triggered the design of learning-based methods for improving reliability without prior knowledge of such network statistics. Based on the seminal paper of [60], this RSRL approach was shown to be highly efficient in [61], which studied the problem of beamforming and transmit power

optimization for URLLC applications. In [62], a different approach of RSRL was taken, by re-engineering the QL algorithm such that the probability of visiting a risk-state (i.e., target PLR violation state) is minimized. However, these works do not exploit multi-interface diversity and are not applicable to Sub-6GHz/mmWave integrated networks envisioned in B5G.

In this chapter, we aim at improving the reliability of a Sub-6GHz/mmWave integrated network based on a novel RSRL approach, without prior knowledge of the statistics of each interface. Specifically, our proposed method enables to learn the adequate interfaces (Sub-6GHz, mmWave, or both) to be selected for each user and every scheduling time frame, solely based on the ACK/NACK feedback information from each user. Indeed, to be compatible with the stringent latency requirements and limited device capabilities of mMTC and URLLC types of applications such as in automated factories, instantaneous CSI is assumed unavailable in the considered system. In addition, mmWave signals may undergo severe dropouts, due to their blockage sensitivity, particularly in such factory scenarios. To cope with the uncertainties and dynamics of the wireless environment in the absence of CSI knowledge or statistical interface models, we propose to leverage upon a new and highly efficient approach initially developed in the context of trading markets, coined as Risk-Averse Averaged Q-Learning (RAQL) [11]. Based on this approach, each AP learns to optimize its interface selection for each device in order to maximize the overall successful packet delivery rate, while avoiding the risk of unsatisfied PLR requirement for each device.

6.2 System model

We consider the downlink transmissions in a wireless network composed of multiple APs serving a number of IoT devices, all equipped by both Sub-6GHz and mmWave interfaces. As shown in Fig. 6.1, AP b transmits desired packets to a set $\mathcal{K}$ of IoT devices, while they receive DL interferences from all other APs $b' \neq b$.

At the beginning of each scheduling frame t, AP b has $L_k(t)$ packets, each of size d in number of bits, to be transmitted to device $k \in \mathcal{K}$. $L_k(t)$ can be modeled as an independent and identically distributed (i.i.d) random variable following, e.g., the truncated Poisson distribution as in [62].

AP b transmits these packets through N subchannels on the Sub-6GHz interface, and

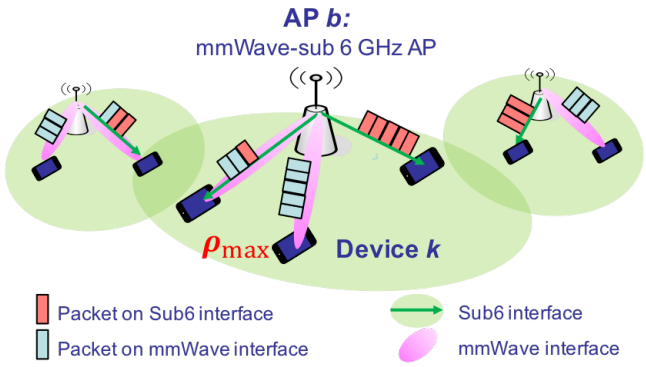


Figure 6.1: System model: DL transmissions to IoT devices, Sub-6GHz and mmWave interfaces

through M beams on the mmWave interface. Each Sub-6GHz subchannel or mmWave beam can be allocated to a unique device, in each scheduling time frame. However, multiple devices can be supported in each frame, over the different subchannels in Sub-6GHz and beams in mmWave.

In Sub-6GHz band, the Signal-to-Interference plus Noise Ratio (SINR) from AP b to device k on subchannel n is,

$$\gamma_{bkn}^{\text{sub}}(t) = \frac{p_{bkn}^{\text{sub}}(t) h_{bkn}^{\text{sub}}(t)}{I_{bkn}^{\text{sub}} + W_{\text{sub}} \sigma_n^2}, \tag{6.1}$$

where transmit power p_{bkn}^{sub} from AP b to device k on subchannel n is assumed equal among subchannels, and W_{sub} is the bandwidth per subchannel. The term h_{bkn}^{sub} is the channel power between AP b and device k on subchannel n, given by $h_{bkn}^{\text{sub}}(t) = |\tilde{h}_{bkn}^{\text{sub}}(t)|^2$ with $\tilde{h}_{bkn}^{\text{sub}}(t)$ the complex channel coefficient including small-scale and large-scale fading effects. The term σ_n^2 denotes the Additive White Gaussian Noise (AWGN) power and I_{bkn}^{sub} is the interference power from APs $b' \neq b$ towards device k on subchannel n, where it is assumed that all APs transmit with full power.

For the mmWave interface, assuming analog beamforming, the transmit beamwidth and beam direction from AP b to device k on beam m are denoted as θ_{bkm} and β_{bkm}, respectively, and adjusted depending on the served device k at each beam m and time frame t. For simplicity and without loss of generality, the receive beam gain G_k^{Rx} at the device k will be assumed fixed as in [51].

To maximize the achievable rate, θ_{bkm} is set to the narrowest beamwidth, and β_{bkm}

is given by the Line-of-Sight (LoS) direction from AP b to device k. Hence, the SINR at device k served by AP b on beam m is given as

$$\gamma_{bkm}^{\mathrm{mW}} = \frac{p_{bkm}^{\mathrm{mW}} h_{bkm}^{\mathrm{mW}}(\theta_{bkm}, \beta_{bkm})}{I_{bkm}^{\mathrm{mW}} + W_{\mathrm{mW}} \sigma_n^2}, \tag{6.2}$$

where p_{bkm}^{mW}, h_{bkm}^{mW} are the transmit power and channel power between AP b and device k on beam m, respectively, and W_{mW} is the bandwidth. The channel power h_{bkm}^{mW} is a function of transmit beamwidth and direction on beam m,

$$h_{bkm}^{\mathrm{mW}}(\theta_{bkm}, \beta_{bkm}) = G_b(\theta_{bkm}, \beta_{bkm}) \tilde{h}_{bkm}^{\mathrm{mW}} \mathrm{PL}_{bkm} G_k^{\mathrm{Rx}}, \tag{6.3}$$

where $\tilde{h}_{bkm}^{\mathrm{mW}}$ and PL_{bkm} denote small-scale fading and path loss between AP b and device k on beam m, and $G_b(\theta_{bkm}, \beta_{bkm})$ is the main transmit beam gain between AP b and device k, modeled as [51]

$$G_b(\theta_{bkm}, \beta_{bkm}) = \frac{2\pi - (2\pi - \theta_{bkm})\epsilon}{\theta_{bkm}}, \tag{6.4}$$

where ϵ is the side lobe beam gain. In (6.2), I_{bkm}^{mW} denotes the interference power from all APs $b' \neq b$ towards device k served by AP b, computed based on their side lobe beam gains.

The achievable rate of device k served by AP b is hence

$$r_{bk}^{v}(t) = \begin{cases} W_{\mathrm{sub}} \log_2(1 + \gamma_{bkn}^{\mathrm{sub}}(t)), & \text{if device } k \text{ is allocated Sub-6GHz subchannel } n, \\ W_{\mathrm{mW}} \log_2(1 + \gamma_{bkm}^{\mathrm{mW}}(t)), & \text{if device } k \text{ is allocated mmWave beam } m, \end{cases} \tag{6.5}$$

where $v = \{\mathrm{Sub}, \mathrm{mW}\}$ (Sub 6GHz or mmWave). Given the low delay requirements of IoT applications, no instantaneous CSI feedback is assumed from devices to APs, as in [62]. Hence, APs should make allocation decisions without the knowledge of achievable rates (6.5). In addition, it should be noted that mmWave transmissions result in a small but non-zero rate with non-LoS (NLoS) path loss in case of blockage.

We denote as $l_k^v(t) \in \{0, \ldots, L_k(t)\}$ the number of transmit packets to device k on interface v at frame t, where $l_k^{\mathrm{sub}}(t) + l_k^{\mathrm{mW}}(t) \leq L_k(t)$, as $L_k(t)$ is the total number of queued packets at frame t [1]. Then, the number of successfully received packets $\Omega_k^v(t)$

[1] Given the low latency requirements of targeted applications, packets that were not served in

at device k on each interface, can be calculated by AP b based on device k's ACK feedback, by

$$\Omega_k^v(t) = \sum_{l=1}^{l_k^v(t)} \omega_{kl}^v(t), \tag{6.6}$$

where $\omega_{kl}^v(t)$ denotes the feedback from device k for packet l on interface v for frame t, where

$$\omega_{kl}^v(t) = \begin{cases} 1 & \text{for ACK packets} \\ 0 & \text{for NACK packets.} \end{cases} \tag{6.7}$$

Moreover, within a frame of duration T_s, the maximum number of packets of size d bits that can be received successfully at device k on interface v is given by

$$l_{k,\max}^v(t) = \lfloor \frac{r_{bk}^v(t) \times T_s}{d} \rfloor. \tag{6.8}$$

It should be noted that $l_{k,\max}^v$ is unknown at the AP, since $r_{bk}^v(t)$ is unknown. It is thus assumed that if $l_k^v(t) \leq l_{k,\max}^v(t)$, i.e., the number of transmit packets is smaller than that can be received by device k in its allocated subchannel or beam, all these packets will be successfully received and their ACKs will be fed back to the AP. But if $l_k^v(t) \geq l_{k,\max}^v(t)$, then $l_k^v(t) - l_{k,\max}^v(t)$ packets will be in NACK state.

Based on the above, we define the PLR of device k for frame t by averaging over packet loss occurrences up to frame t over both interfaces, as

$$\rho_k(t) = \frac{1}{t} \sum_{\tau=1}^{t} \left[1 - \frac{\Omega_k^{\text{sub}}(\tau) + \Omega_k^{\text{mW}}(\tau)}{l_k^{\text{sub}}(\tau) + l_k^{\text{mW}}(\tau)} \right], \tag{6.9}$$

where $\frac{\Omega_k^{\text{sub}}(\tau)+\Omega_k^{\text{mW}}(\tau)}{l_k^{\text{sub}}(\tau)+l_k^{\text{mW}}(\tau)}$ expresses the Packet Successful Delivery Rate (PSR) over both interfaces for frame τ. Finally, the PLR of device k at frame t on each interface is updated as

$$\rho_k^v(t) = \begin{cases} \frac{t-1}{t}\rho_k^v(t-1) & \text{if } l_k^v(t) = 0 \\ \frac{1}{t}\left[(t-1)\rho_k^v(t-1) + (1 - \frac{\Omega_k^v(t)}{l_k^v(t)})\right] & \text{if } l_k^v(t) > 0, \end{cases} \tag{6.10}$$

previous frames are dropped, as in [62].

with $\rho_k^v(0) = 0, \forall v = \{\text{Sub, mW}\}$.

6.3 Problem formulation

We formulate the problem of maximizing the long-term average PSR over all devices, while satisfying individual PLR constraints set to $\rho_{\max}$ here, as

$$\max_{l_k^{\text{sub}}(t), l_k^{\text{mW}}(t)} \mathbf{E}_t \left[\frac{1}{K} \sum_{k \in \mathcal{K}} \frac{\Omega_k^{\text{sub}}(t) + \Omega_k^{\text{mW}}(t)}{l_k^{\text{sub}}(t) + l_k^{\text{mW}}(t)} \right] \tag{6.11}$$

$$\text{s.t.} \quad l_k^{\text{sub}}(t), l_k^{\text{mW}}(t) \in \{0, \dots, L_k(t)\}, \forall k \in \mathcal{K}, \tag{6.11a}$$

$$l_k^{\text{sub}}(t) + l_k^{\text{mW}}(t) \leq L_k(t), \forall k \in \mathcal{K}, \tag{6.11b}$$

$$\sum_{k \in \mathcal{K}} ||l_k^{\text{sub}}(t)||_0 \leq N, \quad \sum_{k \in \mathcal{K}} ||l_k^{\text{mW}}(t)||_0 \leq M, \tag{6.11c}$$

$$\rho_k(t) = \frac{1}{t} \sum_{\tau=1}^{t} 1 - \frac{\Omega_k^{\text{sub}}(\tau) + \Omega_k^{\text{mW}}(\tau)}{l_k^{\text{sub}}(\tau) + l_k^{\text{mW}}(\tau)} \leq \rho_{\max}, \forall k \in \mathcal{K}. \tag{6.11d}$$

Eq. (6.11a) sets the domain of definition for each variable. Eq. (6.11b) expresses that the number of allocated packets on both interfaces should not exceed that of generated packets. Eqs. (6.11c) fixes the maximum amount of resources on each interface. Finally, Eq. (6.11d) constrains the PLR of each device to be lower than target $\rho_{\max}$. Given the intractability of this problem, we design a RL framework based on the MDP model explained next.

6.4 Proposed method

We first formulate the considered distributed problem as a MDP. Then, the proposed method based on Risk-Averse Averaged Q-Learning (RAQL) is devised, whereby each AP learns to optimize interface selection for packet transmission to devices in each time frame, in order to best satisfy the PLR requirement for each device.

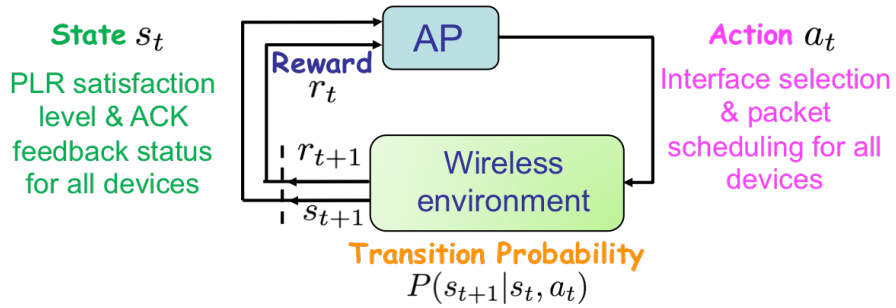


Figure 6.2: MDP model of the interface selection and packet scheduling in Sub-6GHz/mmWave integrated networks

6.4.1 Markov Decision Process Formulation

The goal is to maximize the long-term PSR averaged over all devices, while satisfying the individual PLR constraints for each device, set to $\rho_{\max}$ here. This problem can be modeled as an MDP as depicted in Fig. 6.2, characterized by its state space, action space, transition probabilities and reward function whose detailed definitions will be given in subsection 6.4.3. Namely, each AP is an agent that takes its decisions of interface selection and packet scheduling. At each frame t, the AP knows the current state environment s_t composed of the current PLR satisfaction levels of its associated devices and their feedback status for previous frame $t-1$. Based on s_t, the AP takes action a_t, i.e., it decides the number of packets to be transmitted over each interface for each device for current frame t, then obtains an immediate reward r_t from the environment, which then moves to a new state s_{t+1}. Here, the AP has no knowledge of the transition probabilities $P(s_{t+1}|s_t, a_t)$ since information such as instantaneous CSIs or interface statistics are unknown. We thus exploit the RL framework to address this problem [50].

6.4.2 Risk-Averse Reinforcement Learning

To best meet the severe reliability requirements, we propose to exploit a newly developed approach of RSRL in [11], coined as the Risk-Averse Averaged Q-Learning (RAQL). Compared to traditional RL methods such as QL whose goal is to maximize the expected return, conventional RSRL as in [60] introduced the notion of risk, linked to the variance of the reward. The RAQL proposed in [11] achieves further variance

reductions, thereby reducing risk, while also providing convergence guarantees.

Formally, instead of taking the expected reward as the objective function as in traditional RL, the following expected utility of the reward is used [60] [11]

$$J_\pi = \frac{1}{\beta}\mathbb{E}_{\pi,h}\left[\exp\left(\beta\sum_{t=0}^{\infty} r_t\right)\right], \tag{6.12}$$

where the expectation is over the stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, which is a probability distribution function for choosing actions, and channel realizations h over both interfaces. By taking the Taylor expansion of (6.12),

$$\mathbb{E}_{\pi,h}\left[\sum_{t=0}^{\infty} r_t\right] + \frac{\beta}{2}\mathbb{V}ar\left[\sum_{t=0}^{\infty} r_t\right] + O(\beta^2), \tag{6.13}$$

it is clear that $\beta < 0$ makes the objective function to be risk-averse, as the expected reward can be maximized while its variance is minimized.

While the algorithm of [60] was proven to converge to the optimum of (6.12), the RAQL of [11] enabled to further reduce the training variance by choosing more risk-averse actions, thereby potentially reducing the convergence time to the optimum of (6.12). This is achieved by training multiple Q-tables in parallel. Then, to select more stable actions, the sample variance of those Q-tables is used as an approximation to the true variance, from which a risk-averse $\hat{Q}$-table is computed and used for risk-averse action selection. Next, we explain the details of the proposed algorithm based on RAQL.

6.4.3 Proposed RAQL based interface selection and packet scheduling method

The proposed RAQL based algorithm takes place at each AP, where the state and action spaces are defined as follows.

State: $s(t)$ is the current QoS satisfaction level in terms of PLR, for all devices $k \in \mathcal{K}$ at frame t, and their latest ACK feedback for packets sent at frame $(t-1)$, i.e.,

$$s(t) = \{u_k^{\text{sub}}(t), u_k^{\text{mW}}(t), \Omega_k^{\text{sub}}(t-1), \Omega_k^{\text{mW}}(t-1), \forall k \in \mathcal{K}\}, \tag{6.14}$$

where

$$u_k^v(t) = \begin{cases} 1 & \text{if } \rho_k^v(t) \leq \rho_{\max} \\ 0 & \text{otherwise.} \end{cases} \tag{6.15}$$

As such, the maximum number of possible actions for each user k is $[2(L_k + 1)]^2$, resulting a total the cardinality of state spaces of $[2(L_k + 1)]^{2|\mathcal{K}|}$.

Action: $a(t)$ is the interface choice over which each device's packets will be transmitted. To avoid the explosion of the action space size and to make the proposed method scalable for future work, we propose to condense the interface selection task and packet scheduling tasks into the three actions $a_k(t)$ for device k defined next, based on the following rationale. While APs don't have instantaneous CSI knowledge, it is reasonable to assume that the long-term CSI such as average path loss or average SINRs are known, based on sporadic feedback. Hence, each AP can perform subchannel and beam allocation based on the average CSI of each device. In that case, all subchannels are equivalent for each device, so APs can randomly choose each subchannel to be allocated to each device. Then, each AP's scheduling task amounts to deciding the number of packets to be transmitted per subchannel for each device. The maximum number of packets that can be successfully received by device k from AP b during framelength T_s can be estimated as

$$\tilde{l}_{k,\max}^v = \lfloor \frac{\tilde{r}_{bk}^v \times T_s}{d} \rfloor, \tag{6.16}$$

where $\tilde{r}_{bk}^v$ is the known average rate of device k on interface v. Actions $a_k(t)$ are hence given as

- $a_k(t) = 0$: only Sub-6GHz interface is used and the number of transmit packets are

$$l_k^{\text{sub}}(t) = \min\{\tilde{l}_{k,\max}^{\text{sub}}, L_k(t)\}, \quad l_k^{\text{mW}}(t) = 0, \tag{6.17}$$

- $a_k(t) = 1$: only mmWave interface is used and the number of transmit packets are

$$l_k^{\text{sub}}(t) = 0, \quad l_k^{\text{mW}}(t) = \min\{\tilde{l}_{k,\max}^{\text{mW}}, L_k(t)\}, \tag{6.18}$$

- $a_k(t) = 2$: both Sub-6GHz and mmWave interfaces are used, but with higher

priority on mmWave by maximizing its transmit packets to take advantage of its high data rates,

$$l_k^{\mathrm{mW}} = \min(\tilde{l}_{k,\max}^{\mathrm{mW}}, L_k(t)) \tag{6.19}$$

$$l_k^{\mathrm{sub}}(t) = \begin{cases} \min(\tilde{l}_{k,\max}^{\mathrm{sub}}(t), L_k(t) - l_k^{\mathrm{mW}}(t)) & \text{if } l_k^{\mathrm{mW}}(t) < L_k(t), \\ 1 & \text{otherwise.} \end{cases} \tag{6.20}$$

Finally, overall action $a(t)$ is given for all devices, under the constraints of the number of subchannels and beams, as

$$a(t) = \{a_k(t) \in \{0,1,2\}, \forall k \in \mathcal{K} \Big| \sum_{k\in\mathcal{K}} \mathbb{I}(a_k(t)=0) + \mathbb{I}(a_k(t)=2) \le N \\ \& \sum_{k\in\mathcal{K}} \mathbb{I}(a_k(t)=1) + \mathbb{I}(a_k(t)=2) \le M\}. \tag{6.21}$$

In case of a sufficient number of subchannels and beams, i.e., the resource constraint is satisfied, the maximum number of possible actions is $3^{|\mathcal{K}|}$.

Reward: $r(s(t), a(t))$ denotes the immediate reward achieved by performing action $a(t)$ at frame t, given by the average PSR over devices. In particular, this reward function also takes into account the risk state defined in (6.15). Based on the ACK/NACK feedbacks (6.7) from which the AP gets $\Omega_k^{\nu}(t)$ in (6), the reward is computed as

$$r(s(t), a(t)) = \frac{1}{t}\sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}} \frac{\Omega_k^{\mathrm{sub}}(\tau) + \Omega_k^{\mathrm{mW}}(\tau)}{l_k^{\mathrm{sub}}(\tau) + l_k^{\mathrm{mW}}(\tau)} - (1 - u_k^{\mathrm{sub}}(t)) - (1 - u_k^{\mathrm{mW}}(t)). \tag{6.22}$$

Obviously, when $u_k^{\nu}(t) = 0$, i.e., device k is in a risk state for not satisfying its PLR as in (6.15), the reward is penalized. In other words, a risk-averse action has a higher probability of achieving a higher reward, thereby of being selected.

The details of the proposed RAQL based interface selection and packet scheduling method are given in Algorithm 6.1. Namely, at the beginning, the AP initializes I Q-tables, along with table V to count the number of selections of each action a given state s. The corresponding learning rate α is also initially set to 0 and the algorithm starts with a random state (Lines 1–2). At each frame t, a Q-table is randomly chosen and used to calculate a risk-averse $\hat{Q}$-table by (6.23) (Lines 3–5).

$$\hat{Q}(s,a) = Q(s,a) - \lambda_p \frac{\sum_{i=1}^{I} (Q^i(s,a) - \bar{Q}(s,a))^2}{I-1}, \tag{6.23}$$

where λ_p is the risk control parameter and $\bar{Q}(s,a) = \frac{1}{I}\sum_{i=1}^{I} Q^i(s,a)$ is the average Q-table.

Algorithm 6.1: Proposed interface selection and scheduling algorithm based on Risk-Averse Average Q-Learning

Input: Exploration rate ε, decay factor λ, number of Q-tables I, risk control parameter λ_p, utility function parameter β

1 **for** $i = 1, \ldots, I$ **do**
2 Initialize $Q^i = \mathbf{0}, V^i = \mathbf{0}, \alpha^i = \mathbf{0}$, state s;
3 **for** $t = 1, 2, \ldots, T$ **do**
4 Take a Q-table randomly: $Q = Q^H$, where $H \leftarrow$ Random number in $[1, \ldots, I]$;
5 Compute $\hat{Q}$ by (6.23); $\varepsilon \leftarrow \varepsilon \times \lambda$;
6 **if** *random number* $p < \varepsilon$ **then**
7 Select random action a
8 **else** Select action a with max $\hat{Q}(s,a)$;
9 Perform action a and receive a reward by (6.22);
10 Generate a mask $J \in \mathbb{R}^I \sim Poisson(1)$;
11 **for** $i = 1, \ldots, I$ **do**
12 **if** $J_i = 1$ **then**
13 Update Q^i by (6.24);
14 Update $V^i(s,a) = V^i(s,a) + 1$;
15 Update $\alpha^i(s,a) = \frac{1}{V^i(s,a)}$;
16 Move to the new state;

Next, given the current state and exploration rate ε, action $a(t)$ is selected by the ε-greedy strategy. The AP transmits packets based on the selected action, and receives the immediate reward (6.22) (Lines 6–9). Finally, the Poisson masks guarantee parallel updates of the Q tables, V and α, for the current state-action pair (s,a). Unlike

conventional QL as in [50], RAQL updates its Q-function by

$$Q(s(t), a(t)) = Q(s(t), a(t)) + \alpha(s(t), a(t)) \times \left[u\Big(r(s(t), a(t)) + \gamma \max_{a} Q(s(t+1), a) - Q(s(t), a(t))\Big) - x_0\right], \quad (6.24)$$

where $\alpha(s(t), a(t))$ is the learning rate for the state-action pair $(s(t), a(t))$, γ is a discount factor and $u(x)$ is a monotonically increasing concave utility function, given as in [11,60] by

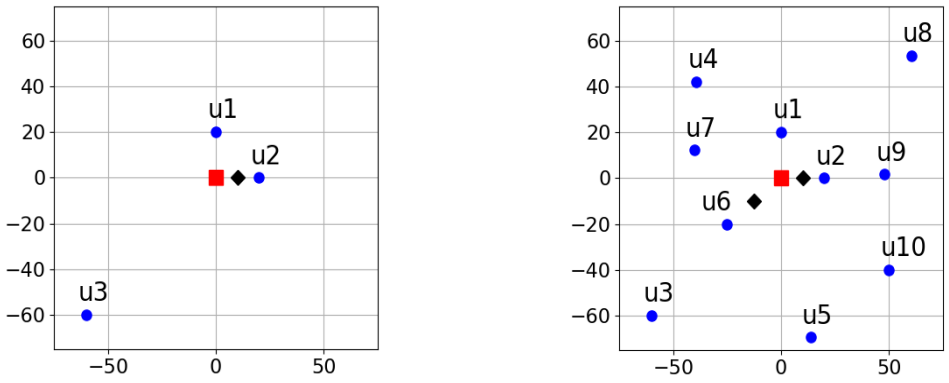
$$u(x) = -e^{\beta x}, \quad (6.25)$$

with $\beta < 0$. The environment then moves to a new state (Lines 10–16). This process is repeated until the maximum number of frames T is reached.

6.5 Numerical evaluations

6.5.1 Simulation Settings

The proposed algorithms are evaluated in various scenarios by varying the number of devices K and the number of Sub-6GHz subchannels N and mmWave beams M, where each subchannel has different channel power due to small-scale and large-scale fading effects, while each beam has a smallest beamwidth and its direction is assumed to be adjustable for LoS direction from AP to the considered user. We consider two scenarios with different network statistics: the small network (Scenario 1) in Fig. 6.3(a) with $K = 3$, $N = M = 4$, and the larger network (Scenario 2) in Fig. 6.3(b) with $K = 10$, $N = M = 16$ which produces up to 3^{10} possible actions. In scenario 1, devices 1 and 2 are at the same distance to the AP, but device 2's link is blocked by an obstacle (the black diamond), and device 3 is the farthest one, allowing us to access the behavior of the proposed algorithm in term of interface usage distribution. Besides these three devices, other seven devices with different distances and directions to the AP are added in scenario 2, where devices 2 and 6's links are blocked, for generating more uncertain environment. In both scenarios, the number of transmit packets to each device per time frame is set to 6, i.e,. $L_k(t) = 6$, for all $T = 10000$ frames. The PLR requirement $\rho_{\max}$ is set to 0.1. Moreover, all devices are assumed to be fixed but undergo block Rayleigh



(a) Scenario 1 (b) Scenario 2

Figure 6.3: Simulation scenarios (obstacles in black diamonds)

Table 6.1: Simulation Parameters

Parameter	**Description**
Transmit power (p^{sub}_{bkn}, p^{mW}_{bkm})	5 dBm
Noise power at device sides	-169 dBm/Hz
Bandwidth (Sub-6GHz, mmWave)	(100 MHz, 1 GHz)
Carrier frequency (Sub-6GHz, mmWave)	(2 GHz, 28 GHz)
Small-scale fading model	Rayleigh fading
Path loss - Sub-6GHz LoS Path loss - mW NLoS Path loss - mmWave [56]	$38.5 + 30\log_{10}(d)$ $61.4 + 20\log_{10}(d) + X_\sigma$, $X_\sigma \sim \mathcal{N}(0,5.8\text{ dB})$ $72 + 29.2\log_{10}(d) + X_{\sigma'}$, $X_{\sigma'} \sim \mathcal{N}(0; 8.7\text{ dB})$ (d: AP-user distance [m])

channel fading, i.e., the channel coefficient for each subchannel (beam) remains fixed during each frame of 1 ms, but changes randomly across frames.
We built our own simulator using Keras API and Numpy library in Python 2.7. Detailed settings of the wireless system are given in Table 6.1. For the proposed method, the number of Q-tables I is set to 2 or 4; the risk control λ_p and utility function parameters β are set to 0.5 and -0.5, respectively. Learning parameters are set as in [54] to $\gamma = 0.9$, $\varepsilon = 0.5$, and decay factor $\lambda = 0.995$.

The algorithms are also evaluated in terms of the metric $\bar{\Delta}_\rho = \frac{1}{K}\sum_{k=1}^{K}(\rho_{\max} - \rho_k)$, which quantifies the margin to the target PLR, averaged over devices. As lower PLR levels are preferred, the higher $\bar{\Delta}_\rho$, the better the performance.

6.5.2 Benchmark schemes

We compare the proposed algorithm to RSQL [60] and the standard Q-Learning [50] approaches:

- **Reference RSQL with three actions** (*Ref. RSQL-3a*): This method has the same state/action/reward definitions as the proposed one, but uses the conventional RSRL of [60] with one Q-table, without any risk-averse $\hat{Q}$-table.
- **Reference QL with three actions** (*Ref. QL-3a*): This is similar to the above, but using basic QL of [50].

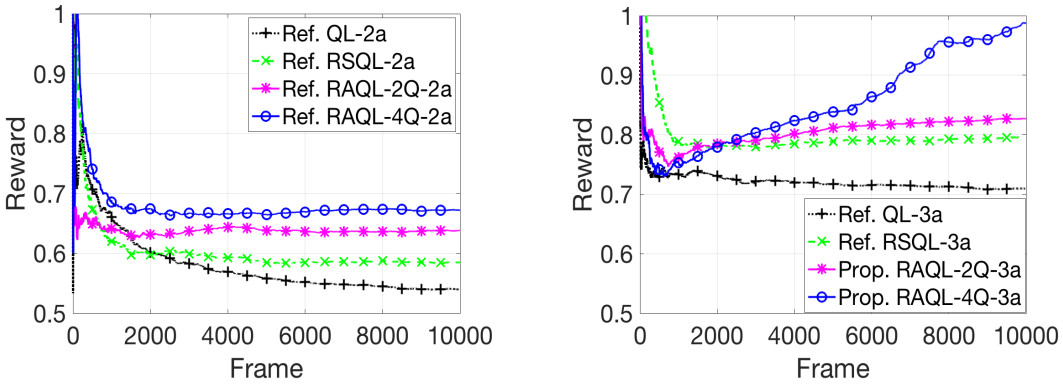
In addition, all algorithms are evaluated with only two types of actions instead of three, where either Sub-6GHz or mmWave interface can be used but not both simultaneously (i.e., $a_k \in \{0, 1\}$), as in most existing methods in the literature. These methods will be denoted by suffix "*2a*" instead of "*3a*".

6.5.3 Simulation Results

6.5.3.1 Scenario 1

Firstly, the reward evolution against time frames in Fig. 6.4 shows that all algorithms converge well. When using four Q-tables, the reward for the proposed method even keeps increasing after 10000 frames. The algorithms with only two possible actions are all outperformed by their three-action counterparts. We also note that *Prop. RAQL-2Q-3a* requires a training time of around 5000 frames and *Prop. RAQL-4Q-3a*, around 8000 frames, namely 5 and 8 seconds, respectively, when learning from scratch. In practice, this training period can be made offline, so this does not reflect the delay during actual packet transmissions which are initiated once the network has been trained. Furthermore, less time would be required in the case of online learning for coping with subsequent variations of the network environment. These aspects will be investigated in depth in the future work.

Next, the detailed results in Table 6.2 show that the proposed methods outperform all baselines in terms of reward, average PSR over devices, individual PLRs as well as $\bar{\Delta}_\rho$ metric. In addition, we can see that by allowing to transmit over Sub-6Ghz and mmWave simultaneously, the overall performance is greatly improved. Namely,



(a) Algorithms with two actions

(b) Algorithms with three actions

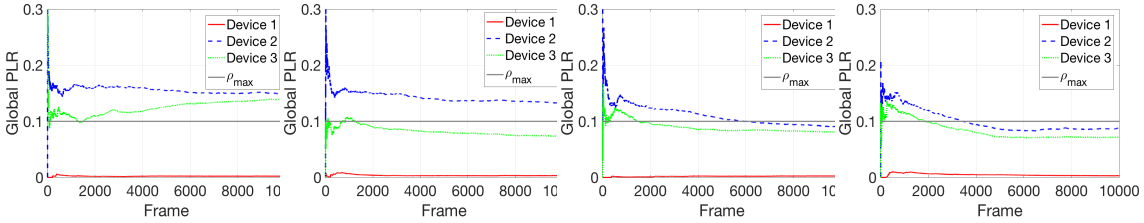
Figure 6.4: Reward evolution of all algorithms, scenario 1

all reference methods with two possible actions and *Ref. QL-3a* only satisfy the LoS device 1 which is also the closest to the AP and hence, whose target PLR is easiest to meet. By contrast, the other devices suffer very high PLRs. By allowing three possible actions and being risk-sensitive, *Ref. RSQL-3a* also satisfies the PLR constraint of the farthest device 3 besides device 1, but not that of device 2 whose link is blocked. By our proposed methods based on RAQL using 2 and 4 Q-tables, all devices can finally satisfy their PLR requirements.

Table 6.2: Detailed results for scenario 1

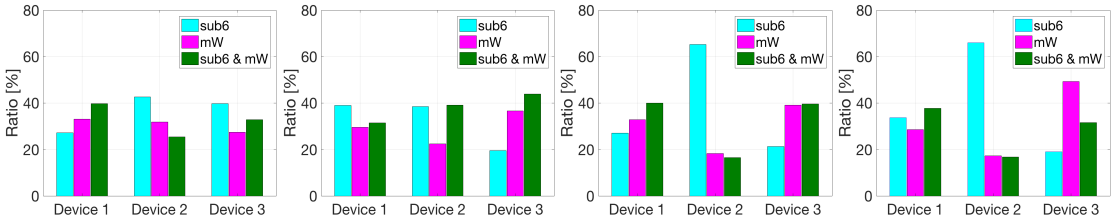
Algorithm	**Reward**	**Avg. Success**	**PLR**			$\bar{\Delta}_\rho$
			u1	**u2**	**u3**	
Ref. QL-2a	0.540	0.846	0.004	0.271	0.188	-0.054
Ref. RSQL-2a	0.585	0.861	0.005	0.260	0.153	-0.039
Ref. RAQL-2Q-2a	0.638	0.879	0.003	0.220	0.140	-0.021
Ref. RAQL-4Q-2a	0.672	0.889	0.003	0.185	0.145	-0.011
Ref. QL-3a	0.709	0.903	0.002	0.149	0.140	0.003
Ref. RSQL-3a	0.796	0.930	0.003	0.133	0.077	0.029
Prop. RAQL-2Q-3a	0.827	0.942	0.002	0.091	0.081	0.042
Prop. RAQL-4Q-3a	0.987	0.946	0.003	0.088	0.072	0.046

The evolution of the PLR perceived by each device and the interface usage distribution for each device are shown in Figs. 6.5 and 6.6, in the case of three actions. Again, we observe that device 1 with the best channel conditions can be easily satisfied



(a) *Ref. QL-3a* (b) *Ref. RSQL-3a* (c) *Prop. RAQL-2Q-3a* (d) *Prop. RAQL-4Q-3a*

Figure 6.5: Evolution of the packet loss rate per device, scenario 1



(a) *Ref. QL-3a* (b) *Ref. RSQL-3a* (c) *Prop. RAQL-2Q-3a* (d) *Prop. RAQL-4Q-3a*

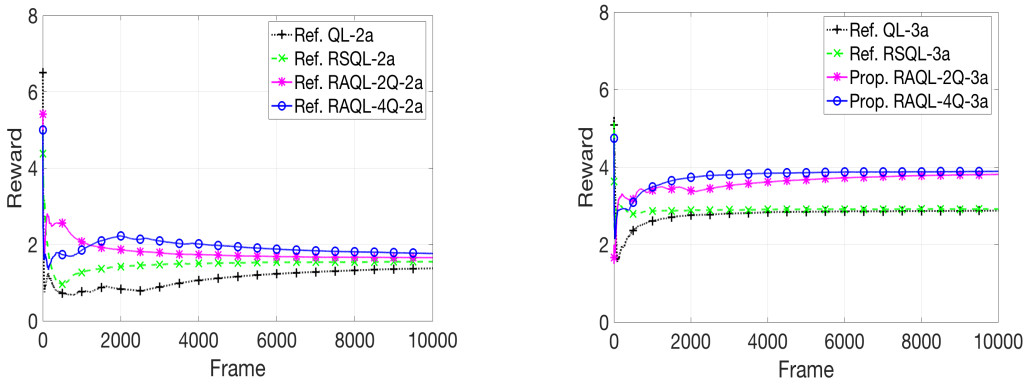
Figure 6.6: Interface usage distribution per device, scenario 1

by all algorithms. For edge device 3, *Ref. RSQL-3a* enables to satisfy its PLR requirement by using more the mmWave or both interfaces, as in proposed algorithms. For blocked device 2, all algorithms show a decreasing tendency of its PLR over time, by using more the Sub-6Ghz interface as shown in Fig. 6.6. However, the proposed RAQL-based algorithms are the only ones enable to satisfy its PLR target, by learning to mostly use Sub-6Ghz.

6.5.3.2 Scenario 2

Fig. 6.7 shows the reward evolution over frames for all algorithms. It can be observed that the three-action methods improve over their two-action counterparts, and the proposed methods largely outperform the baselines ones as shown in Figs. 6.7(a) and (b).

The detail results are given in Table 6.3. We can see that although the three-action methods achieve higher average PSR over devices than the two-action algorithms, they still suffer from negative $\bar{\Delta}_\rho$. This means that the PLRs of many devices are much higher than their target PLR. In particular, *Prop. RAQL-4Q-3a* provides the best



(a) Algorithms with two actions

(b) Algorithms with three actions

Figure 6.7: Reward evolution of all algorithms, scenario 2

Table 6.3: Detailed numerical results for scenario 2

Algorithm	Reward	Avg. Success	$\bar{\Delta}_\rho$
Ref. QL-2a	1.377	0.858	-0.0422
Ref. RSQL-2a	1.551	0.859	-0.0414
Ref. RAQL-2Q-2a	1.657	0.861	-0.0393
Ref. RAQL-4Q-2a	1.765	0.861	-0.0392
Ref. QL-3a	2.878	0.890	-0.0095
Ref. RSQL-3a	2.918	0.891	-0.0086
Prop. RAQL-2Q-3a	3.813	0.893	-0.0070
Prop. RAQL-4Q-3a	3.891	0.894	-0.0060

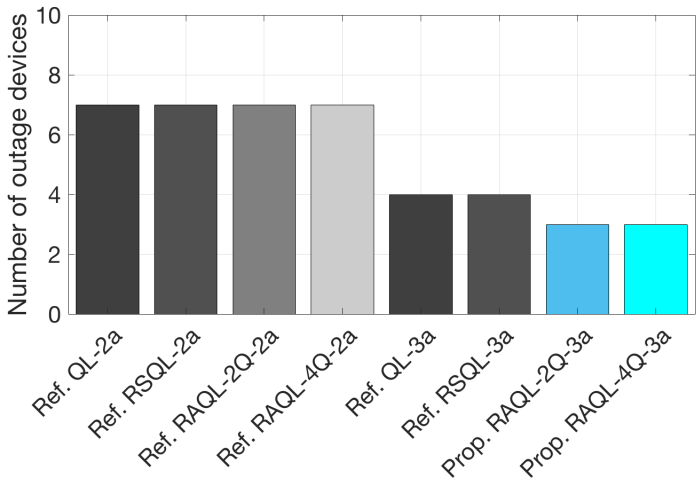


Figure 6.8: Number of outage devices, scenario 2

performance in terms of reward, average PSR over devices, achieving a 4.2% increase of average PSR and up to 7 times higher $\bar{\Delta}_\rho$ as compared to *Ref. QL-2a*.

Next, we show the number of outage devices for all algorithms in Fig. 6.8. It can be observed that, three-action algorithms can significantly reduce the number of outage devices compared to the two-action methods. Among them, the proposed algorithms provide the lowest number of outage devices.

6.6 Summary

We have investigated the issue of improving the reliability of IoT packet transmissions without perfect CSI knowledge, by exploiting multi-connectivity over the Sub-6GHz and mmWave interfaces of B5G integrated networks. We have proposed a method based on Risk-Averse Average Q-Learning, enabling the AP to optimize its own interface selection decision with minimal feedback from devices. Numerical results showed the effectiveness of the proposed method compared to the baselines, by enhancing the overall successful packet reception rate while reducing individual PLRs, despite unknown channel fluctuations.

7 Conclusions and Future Perspectives

7.1 Conclusions

This thesis investigated wireless access optimization and multi-interface connectivity for future networks by focusing upon three major directions: the joint exploitation of a wide range of spectrum from Sub-6GHz to mmWaves, AI-enabled capabilities at devices or AP and energy efficient Deep Learning. In such a context, the study explored and designed RL-based solutions for three major issues:

- User-to-multiple APs association,
- Energy efficiency enhancement of the DQN-based methods for User-to-multiple APs association,
- Reliability enhancement for mMTC use case.

7.1.1 User-to-multiple APs association

Devising efficient user-to-AP association methods is essential for guaranteeing stringent system performance metrics required by B5G/6G networks. However, all these diverse and strict QoS requirements can hardly be fully satisfied by restricting each user to associate with only one AP as in most existing studies. Allowing each user to connect to different radio interfaces across different APs, much higher user satisfaction levels may be achieved. To verify this, we hence investigated the issue of user-to-multiple APs association in two systems and proposed RL/DRL-based multi-AP association at user side, and radio resource allocation optimization at the AP side, under a harsh environment where all APs mutually interfere and where no global CSI knowledge is available for users nor APs.

In Sub-6GHz networks

Firstly, we considered the scenario where each user device requests multiple applications with various QoS requirements, thereby requiring to be served by different APs simultaneously. For this, we proposed algorithms based on Q-Learning, DQN and DDQN techniques at the user side, each of which had two versions: the *fully distributed* method where users receive the minimal feedback notifying only their desired APs' association decision, and the *partially distributed* method where an additional information of local environment regarding the outage of neighboring users is also fed back to the users. The numerical results showed that multiple APs association enables higher network performance than single AP association scheme. Moreover, the proposed algorithms work well and provide the highest performance compared to baseline methods. The partially distributed schemes slightly enhance the performance compared to the fully distributed methods. In addition, the DQN-based methods obtain a performance close to their corresponding DDQN-based counterparts.

In Sub-6GHz/mmWave integrated networks

In these systems, users need to perform both AP and interface selections to make association requests, which increases the difficulty of the user-to-multiple APs association problem. Moreover, to enhance the whole network performance, we then

proposed a DQN-based user association and beamforming method which took into account the features of high path loss and obstacle sensitivity of mmWave bands. Namely, user devices learn to optimize the pair (AP, interface) for their requested applications through their own DQNs, whereas APs perform greedy strategies on each interface for deciding the users/applications to be served or to be dropped. The simulation results showed that the proposed algorithm is efficient for solving the issue-to-multiple APs association in Sub-6GHz/mmWave integrated networks by enhancing the network performance while better satisfying the QoS requirement compared to the reference methods.

7.1.2 Energy efficiency enhancement of the DQN-based methods for User-to-multiple APs association

Given the proposed DQN-based method for user-to-multiple APs association implemented at the user side, we conducted a comprehensive analysis of energy consumption on both computation and data access for operating the DQN. Then, to cope with the dynamics of environments, and particularly to improve the energy efficiency for DQN-enabled user devices and the network performance, we enhanced the proposed algorithm of user-to-multiple APs association and beamforming by 1) introducing the adaptive ε-greedy policy which allows devices to do exploration whenever the significant changes of their surrounding environments are detected, and 2) adapting the classic Branch-and-Bound algorithm for user selection and beamforming at AP side on mmWave interface. Based on that, we investigated the trade-off between achievable network performances and energy cost at the user side. The numerical results showed that the proposed enhanced frameworks work efficiently even in the dynamic environments given moving user with a pedestrian velocity. In particular, the proposed adaptive ε-greedy user association and beamforming methods achieve much higher energy efficiency than the benchmark schemes by allowing the user devices to explore a subset of APs at each time frame instead of all available APs in their sensing area.

7.1.3 Reliability enhancement for mMTC use case

In the Sub-6GHz/mmWave integrated systems, we also investigated the issue of reliability enhancement for mMTC use case through interface selection (Sub-6GHz, mmWave or both) and packet scheduling. It should be noted that the delay requirement is inherent to reliability. In our work, we assumed that a packet is dropped and considered in outage if it is not received within a frame duration. The proposed Risk-Averse Average Q-Learning based algorithm allows the DQN-enabled APs to optimize their decisions across interfaces and the number of packets to be transmitted to their associated users while avoiding the risk states. The simulation results showed that the proposed method can enhance the reliability of the system while satisfying more users than the baseline methods.

7.2 Future perspectives

The aforementioned achieved results of this thesis show a promising approach to apply RL/DRL, namely QL/DQN techniques, in wireless systems due to their high performances and their ability for handling the uncertainty of wireless environments. At user side, QL/DQN-based methods enable users to independently learn the dynamics of their surrounding mobile environment including the impact of APs' scheduler and traffic distribution, for optimizing their requests, thereby achieving high data rate and low outage probability. At AP side, the method based on RL techniques, namely RAQL, allows the AP to leverage the average CSI, instead of perfect instantaneous CSI which may not be available, for achieving higher reliability of the whole network with a lower number of outage users. However, energy-efficient DQN-based methods are necessary, especially as they are performed at user devices. In addition, the performance of RAQL-based method at AP side in large networks was still not optimized, which should be further enhanced. The findings of thesis also open some interesting directions, which can be listed as follows.

- **Highly dynamic environments:** Mobile users move in and out of the coverage area of APs, even with high velocities, creating challenging dynamics of interference and AP traffic loads. In addition, users also have their own changes of application requirements, i.e., they do not usually keep the same demands for the

same applications during their operating time. In the context of B5G/6G, this becomes more uncertain with the expected Terahertz communication, resulting in more dynamic and unstable wireless environment.

For the user-to-AP association issue, the proposed adaptive ε-greedy strategy seems to be no longer suitable to handle such context as user devices will keep doing exploration with a small probability of exploitation, making the networks difficult to converge. This requires novel techniques to speed up their learning processes. A potential solution for that is to apply transfer learning frameworks [63] allowing the agents to learn from the trained data of different but related problems, instead of learning from scratch.

Regarding reliability enhancement being based on QL techniques, the proposed RAQL-based method suffers from the necessity of visiting all states and actions for guaranteeing its convergence. Consequently, this method becomes inadequate to handle such highly dynamic environments. For that, a direct extension of the RAQL-based method is to leverage DRL techniques, where the AP will be equipped with several DQNs to perform risk-averse deep learning, which is expected to achieve high reliability while enhancing the convergence time. In addition, to further cope with the highly dynamic environments, methods based on transfer learning techniques can be also beneficial, similarly to above.

- **Integration of DNN energy constraints:** In this thesis, we showed that a huge amount of power is consumed by DNN operation for optimizing wireless access. Given the limited-energy devices, the conventional DQN-based schemes seem to be inapplicable. Therefore, the proposed simple yet efficient energy-aware DQN-based method opens a direction for the joint optimization of wireless communication protocols and DNN operation at the device side. For that, integration of DNN energy constraints into the wireless access optimization problem can be a promising way. In such a context, new DNN-based methods need to be investigated to take into account the DNN energy constraints while optimizing the network performances. One possible approach is to leverage the framework of [64] where the sparsity level of DNN can be adjusted depending on the current available device energy through binary input masks of all DNN layers. Hence, only APs/interfaces corresponding to active nodes and weights

(which are not masked) are explored. As such, devices with a high energy budget can spend full capacity of their DNNs, whereas under a low energy situation, some nodes and weights should be put inactive, for optimizing their wireless access requests.

- **Scalability issue:** A fundamental difficulty to apply RL, namely QL, in optimization problems is its scalability since the agent needs to store Q-values for all possible pairs (states, actions) and requires a long training time before convergence. It is more challenging for centralized QL-based methods with an exponentially increasing number of states and actions as the problem size becomes larger. This situation can be observed in the reliability enhancement issue. Namely, in the proposed RAQL-based method, the sizes of the state and action spaces are exponential functions of the number of users. Moreover, the proposed scheme was not optimized, for very large networks. One solution is to integrate an action elimination technique [65] where upper and lower estimated Q-values are used to eliminate actions, i.e., whenever the upper estimated Q-value of an action is below the lower estimated Q-values of other actions, that action is eliminated. However, this approach may cause a higher computational complexity compared to primary QL methods, which should be considered in designing algorithms. One more potential solution is to extend the RAQL-based methods by making use of DRL as mention above. By this, besides the aforementioned convergence issue, the scalability issue can be also controlled in two other aspects: 1) no need to store the Q-values of all possible pairs of states and actions, 2) adjustment of number of hidden layers and hidden nodes to provide a good performance given the size of wireless networks.

  Similarly, the scalability of proposed DQN/DDQN-based methods for the issue of user to AP association can be handled to guarantee the good performance for large-scale networks by adjusting the size of DNN. However, since these methods are performed at limited-battery user devices, setting the number of hidden layers and hidden neural nodes should be cautious, which again requires to investigate the trade-off between the network performance and energy efficiency. In such a context, the proposed adaptive ε-greedy strategy seems to be useful. This is because by limiting the number $B_{\max}$ of explored APs, the number of

actions and states can be reduced, thereby enabling a small number of hidden layers and hidden nodes for obtaining an acceptable performance. Moreover, the small number of actions also reduces the energy consumption by DNN. In addition, in the case of large number of applications and interfaces, the proposed method is still applicable. Namely, if the user needs to request all APs in its sensing area to obtain its minimum requirement, it has no choice. Otherwise, by considering $B_{\max}$ that is lower than the total number of APs, we can achieve a good performance whereas saving energy consumption as discussed.

In addition, so far, the number of hidden layers and hidden nodes were set through manual tuning in general, by running many sample experiments, with the goal of maximizing training and validation accuracies. Our work opens up an interesting direction for tuning these parameters, where the aim would be to strike a desired trade-off between energy costs and accuracy levels, which will be considered in future work.

- **Application of the proposed user-to-multiple APs association methods across different systems:** Considering the issue where each user device requests several applications simultaneously, each of which can be served in different systems, such as 5G, WLAN or LPWA, the proposed user-to-multiple APs association methods need to be adapted to handle the impact of 1) heterogeneous interferences, and wireless signal features, 2) heterogeneous signaling and protocol procedures. Obviously, beside suffering interferences from APs in one system, the user would also encounter interferences from other systems, which may significantly reduce its achievable performances. Moreover, each system operates upon different physical layer technologies providing different connection qualities. For example, LPWA provides long-range communication at the cost of low power and low data rate, whereas WLAN supports high data rates with higher transmit power. These factors should hence be carefully modelled in the proposed user-to-multiple APs association, especially in the reward function design. Furthermore, the current method assumes each device is basically connected to its set of requested interfaces, and that any application may be served freely through both interfaces. If these interfaces belong to different wireless technologies, the effects of heterogeneous signaling, protocol procedures

and incurred delays should also be incorporated in the DRL framework. Although the proposed method incorporates the differences between mmWave and Sub-6GHz interfaces, it still needs to be carefully re-designed to cope with these additional levels of heterogeneities.

- **Hand-off issues:** Considering the hand-off event when a user device switches to other APs instead of staying with its current associated APs, under the proposed user-to-multiple APs association scheme, the static user may suffer from too frequent hand-off events due to its exploration at the beginning. However, this situation will be gradually reduced when the user shifts towards more exploitation. In the case of mobile users performing the adaptive ε-greedy strategy, the users experience even more hand-off events. Although this proposed scheme enables the mobile users to converge fast given a fixed set of APs, such handoff issue still needs to be enhanced. One straight solution is to set a constraint where a user has to stay with the current association APs at least a defined period of frames before they can change. However, this may reduce the network performance when users associate with "bad" APs, but cannot request other APs immediately. This trade-off design can be further investigated in the extended work.

7.3 Concluding remarks

In this thesis, we studied the design of wireless access protocols and radio resource allocation for fulfilling the stringent requirements of future wireless networks by exploiting multi-interface connectivity and AI functionalities. Namely, three issues: user-to-multiple APs association, energy efficiency of Deep Neural Network-based methods and reliability enhancement for mMTC use case, are investigated. For the first issue, we aimed at designing a framework where reinforcement learning-based user-to-multiple APs method is performed at user devices for enabling them to optimize their own APs/interfaces association requests, while APs optimize their user selection and beamforming. Our goal is to maximize the global network performance while satisfying all individual user QoS requirements. For that, user-to-multiple APs association methods are based on Q-Learning (QL)/Deep Q-Networks (DQN) or Double

DQN (DDQN) to handle the dynamics and uncertainty of wireless environments, and APs optimized their beamforming and user selection through greedy-like and Branch-and-Bound-based methods. We then analyzed comprehensively the energy consumption of Deep Neural Network, for both computation and data movement, in the proposed DQN-based user-to-multiple APs association method. Based on that, we enhanced the energy efficiency of the proposed algorithm by introducing an adaptive ε-greedy and $B_{\max}$ strategies. The key idea is to let each user explore only an adaptive subset of best APs according to the dynamics of the mobile environment, instead of exploring all available APs, and to adapt this APs' subset according to the desired trade-off level of communication quality and energy consumption. Finally, concerning the reliability enhancement of mMTC use case which is crucial in B5G/6G, we aimed at designing a method which allows the APs to exploit multi-interface connectivity for maximizing the packet successful rate while satisfying the packet loss rate (PLR) of each user. For that, Risk-averse aversage QL was leveraged, enabling the APs to avoid the risk of violating the PLR requirement and thereby improve the system's reliability.

We believe that our findings and ideas can be adapted to further improve the performance of future wireless networks under more challenging scenarios such as high mobility cases and large-scale and high density networks, as discussed through the open directions and perspectives.

Bibliography

[1] W. Saad, M. Bennis, and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems," *IEEE Network*, vol. 38, no. 3, pp. 134–142, May/Jun. 2020.

[2] T.-J. Yang, Y.-H. Chen, J. Emer, and V. Sze, "A Method to Estimate the Energy Consumption of Deep Neural Networks," *Asilomar Conference on Signals, Systems, and Computers*, pp. 1916–1920, 2017.

[3] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, 2017.

[4] I. F. Akyildiz, A. Kak, and S. Nie, "6G and Beyond: The Future of Wireless Communications Systems," *IEEE Access*, vol. 8, pp. 133 995–134 030, Jul. 2020.

[5] A. S. Andrae and T. Edler, "On Global Electricity Usage of Communication Technology: Trends to 2030," *Challenges*, vol. 6, no. 1, pp. 117–157, 2015.

[6] M. H. Alsharif and R. Nordin, "Evolution Towards Fifth Generation (5G) Wireless Networks: Current Trends and Challenges in The Deployment of Millimetre Wave, Massive MIMO, and Small Cells," *Telecommunication Systems*, vol. 64, no. 4, pp. 617–637, 2017.

[7] H. Bogucka, F. Idzikowski, and B. Bossy, "Balancing Explosive Growth with Dramatic Energy Efficiency Improvements: The Necessity for A Green Revolution in 5G and Beyond," *IEEE Comsoc Technology News*, Sep. 2020.

[8] I. F. Akyildiz, C. Han, and S. Nie, "Combating the Distance Problem in the Millimeter Wave and Terahertz Frequency Bands," *IEEE Communications Magazine*, vol. 56, no. 6, pp. 102–108, 2018.

[9] O. Semiari, W. Saad, M. Bennis, and M. Debbah, "Integrated Millimeter Wave and Sub-6 GHz Wireless Networks: A Roadmap for Joint Mobile Broadband and Ultra-reliable Low-latency Communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 109–115, Feb. 2019.

[10] J.J. Nielsen, R. Liu and P. Popovski, "Ultra-Reliable Low Latency Communication Using Interface Diversity," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1322–1334, Mar. 2018.

[11] Y. Gao, K. Y. C. Lui, and P. Hernandez-Leal, "Robust Risk-Sensitive Reinforcement Learning Agents for Trading Markets," *RL4RealLife Workshop in International Conference on Machine Learning (ICML)*, 2021.

[12] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K.-K. Wong, R. Schober, and L. Hanzo, "User Association in 5G Networks: A Survey and An Outlook," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1018–1044, Jan. 2016.

[13] O. Ercetin, "Association Games in IEEE 802.11 Wireless Local Area Networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 5136–5143, 2008.

[14] T. Zhou, Y. Huang, W. Huang, S. Li, Y. Sun, and L. Yang, "QoS-aware User Association for Load Balancing in Heterogeneous Cellular Networks," in *IEEE Vehicular Technology Conference (VTC - Fall)*, pp. 1–5, Sep. 2014.

[15] D. Liu, Y. Chen, K. K. Chai, and T. Zhang, "Nash Bargaining Solution based User Association Optimization in HetNets," *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, pp. 587–592, 2014.

[16] H. Zhang, W. Wang, X. Li, and H. Ji, "User Association Scheme in Cloud-RAN based Small Cell Network with Wireless virtualization," *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 384–389, 2015.

[17] Y. Xu, R. Q. Hu, L. Wei, and G. Wu, "QoE-aware Mobile Association and Resource Allocation over Wireless Heterogeneous Networks," in *IEEE Global Communications Conference (GLOBECOM)*, pp. 4695–4701, Dec. 2014.

[18] N. Liakopoulos, G. Paschos, and T. Spyropoulos, "Robust User Association for Ultra Dense Networks," in *IEEE INFOCOM Conference on Computer Communications*, pp. 2690–2698, 2018.

[19] N. Liakopoulos, G. S. Paschos, and T. Spyropoulos, "Robust Optimization Framework for Proactive User Association in UDNs: A Data-driven Approach," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1683–1695, 2019.

[20] Q. V. Do and I. Koo, "Actor-Critic Deep Learning for Efficient User Association and Bandwidth Allocation in Dense Mobile Networks with Green Base Stations," *Wireless Networks*, vol. 25, no. 8, pp. 5057–5068, 2019.

[21] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep Learning for Hybrid 5G Services in Mobile Edge Computing Systems: Learn from a Digital Twin," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, 2019.

[22] S. Bayat, R. H. Louie, Z. Han, B. Vucetic, and Y. Li, "Distributed User Association and Femtocell Allocation in Heterogeneous Wireless Networks," *IEEE Transactions on Communications*, vol. 62, no. 8, pp. 3027–3043, Aug. 2014.

[23] D. Liu, L. Wang, Y. Chen, T. Zhang, K. K. Chai, and M. Elkashlan, "Distributed Energy Efficient Fair User Association in Massive MIMO Enabled HetNets," *IEEE Communications Letters*, vol. 19, no. 10, pp. 1770–1773, 2015.

[24] S. Maghsudi and E. Hossain, "Distributed User Association in Energy Harvesting Small Cell Networks: A Probabilistic Bandit Model," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1549–1563, Mar. 2017.

[25] X. Ge, X. Li, H. Jin, J. Cheng, and V. C. Leung, "Joint User Association and User Scheduling for Load Balancing in Heterogeneous Networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3211–3225, 2018.

[26] P.-Y. Chou, W.-Y. Chen, C.-Y. Wang, R.-H. Hwang, and W.-T. Chen, "Deep Reinforcement Learning for MEC Streaming with Joint User Association and Resource Management," in *IEEE International Conference on Communications (ICC)*, pp. 1–7, 2020.

[27] Z. Li, M. Chen, K. Wang, C. Pan, N. Huang, and Y. Hu, "Parallel Deep Reinforcement Learning Based Online User Association Optimization in Heterogeneous Networks," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2020.

[28] A. Hajijamali Arani, M. J. Omidi, A. Mehbodniya, and F. Adachi, "A Distributed Learning-based User Association for Heterogeneous Networks," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 5, pp. 1–13, Nov. 2017.

[29] N. Zhao, X. He, M. Wu, P. Fan, M. Fan, and C. Tian, "Deep Q-Network for User Association in Heterogeneous Cellular Networks," in *Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 398–407, Jul. 2018.

[30] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, and Y. Jiang, "Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Networks," *IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec. 2018.

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

[32] M. Mezzavilla, S. Goyal, S. Panwar, S. Rangan, and M. Zorzi, "An MDP Model for Optimal Handover Decisions in mmWave Cellular Networks," in *European conference on networks and communications (EuCNC)*, pp. 100–105, 2016.

[33] S. Goyal, M. Mezzavilla, S. Rangan, S. Panwar, and M. Zorzi, "User Association in 5G mmWave Networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2017.

[34] C. Chaieb, Z. Mlika, F. Abdelkefi, and W. Ajib, "On The User Association and Resource Allocation in Hetnets with Mmwave Base Stations," in *2017 IEEE 28th an-*

nual international symposium on personal, indoor, and mobile radio communications (PIMRC), pp. 1–5, 2017.

[35] O. Semiari, W. Saad, and M. Bennis, "Joint Millimeter Wave and Microwave Resources Allocation in Cellular Networks with Dual-mode Base Stations," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4802–4816, 2017.

[36] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "EIE: Efficient Inference Engine on Compressed Deep Neural Network," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 243–254, 2016.

[37] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. Emer, S. W. Keckler, and W. J. Dally, "SCNN: An Accelerator for Compressed-Sparse Convolutional Neural Wetworks," *ACM SIGARCH computer architecture news*, vol. 45, no. 2, pp. 27–40, 2017.

[38] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, "In-Datacenter Performance Analysis of a Tensor Processing Unit," pp. 1–12, 2017.

[39] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both Weights and Connections for Efficient Neural Network," *Advances in neural information processing systems*, vol. 28, 2015.

[40] R. Yazdani, M. Riera, J.-M. Arnau, and A. González, "The Dark Side of DNN Pruning," in *ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pp. 790–801, 2018.

[41] Y. Chen, T.-J. Yang, J. Emer, and V. Sze, "Understanding the limitations of existing energy-efficient design approaches for deep neural networks," *Energy*, vol. 2, no. L1, p. L3, 2018.

[42] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[43] D. Sesto-Castilla, E. Garcia-Villegas, G. Lyberopoulos, and E. Theodoropoulou, "Use of Machine Learning for Energy Efficiency in Present and Future Mobile Networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2019.

[44] A. Zappone and M. Debbah, "Complexity-Aware ANN-Based Energy Efficiency Maximization," *IEEE International Conference on Communications (ICC)*, pp. 1–6, 2020.

[45] M.-T. Suer, C. Thein, H. Tchouankem, and L. Wolf, "Multi-Connectivity as An Enabler for Reliable Low Latency Communications—An 0verview," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 156–169, 2019.

[46] J. Sachs, G. Wikstrom, T. Dudda, R. Baldemair, and K. Kittichokechai, "5G Radio Network Design for Ultra-Reliable Low-Latency Communication," *IEEE Network*, vol. 32, no. 2, pp. 24–31, 2018.

[47] J. J. Nielsen, R. Liu, and P. Popovski, "Ultra-Reliable Low Latency Communication Using Interface Diversity," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1322–1334, 2017.

[48] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529—-533, Feb. 2015.

[49] H. Van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-Learning," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[50] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[51] R. Ismayilov, B. Holfeld, R. L. G. Cavalcante, and M. Kaneko, "Power and Beam Optimization for Uplink Millimeter-Wave Hotspot Communication Systems," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2019.

[52] M. Kountouris, D. Gesbert, and T. Sälzer, "Enhanced Multiuser Random Beamforming: Dealing with The Not So Large Number of Users Case," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 8, pp. 1536–1545, 2008.

[53] G. Yang, J. Du, and M. Xiao, "Maximum Throughput Path Selection with Random Blockage for Indoor 60 GHz Relay Networks," *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3511–3524, 2015.

[54] T. H. L. Dinh, M. Kaneko, K. Wakao, H. Abeysekera, and Y. Takatori, "Reinforcement Learning-aided Distributed User-to-Access Points Association in Interfering Networks," *IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec. 2019.

[55] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An Introduction to Deep Reinforcement Learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[56] M. Shi, K. Yang, Z. Han, and D. Niyato, "Coverage analysis of integrated sub-6ghz-mmwave cellular networks with hotspots," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 8151–8164, 2019.

[57] T. Zahavy, M. Haroush, N. Merlis, D. J. Mankowitz, and S. Mannor, "Learn What not to Learn: Action Elimination with Deep Reinforcement Learning," in *Advances in Neural Information Processing Systems*, pp. 3562–3573, 2018.

[58] K. Rahman, N. A. Ghani, A. A. Kamil, and A. Mustafa, "Analysis of Pedestrian Free Flow Walking Speed in a Least Developing Country: a Factorial Design Study," *Research journal of applied sciences, engineering and technology*, vol. 4, no. 21, pp. 4299–4304, 2012.

[59] P. J. Kolesar, "A Branch and Bound Algorithm for the Knapsack Problem," *Management Science*, vol. 13, no. 9, pp. 723–735, 1967.

[60] Y. Shen, M. J. Tobia, T. Sommer, and K. Obermayer, "Risk-Sensitive Reinforcement Learning," *Neural computation*, vol. 26, no. 7, pp. 1298–1328, 2014.

[61] T.K. Vu, M. Bennis, M. Debbah, M. Latva-Aho and C.S. Hong, "Ultra-Relibable Communication in 5G mmWave Networks: A Risk-Sensitive Approach," *IEEE Communications Letters*, vol. 22, no. 4, pp. 708–711, 2018.

[62] N. B. Khalifa, M. Assaad, and M. Debbah, "Risk-Sensitive Reinforcement Learning for URLLC Traffic in Wireless Networks," *IEEE WCNC*, pp. 1–7, 2019.

[63] M. E. Taylor and P. Stone, "Transfer Learning for Reinforcement Learning Domains: A Survey," *Journal of Machine Learning Research*, vol. 10, no. 7, 2009.

[64] H. Yang, Y. Zhu, and J. Liu, "Energy-Constrained Compression for Deep Neural Networks via Weighted Sparse Projection and Layer Input Masking," *International Conference on Learning Representations (ICLR)*, May 2019.

[65] E. Even-Dar, S. Mannor, Y. Mansour, and S. Mahadevan, "Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems," *Journal of Machine Learning Research*, vol. 7, no. 6, 2006.