Intelligent Multi-Dimensional Resource Management in MEC-Assisted Vehicular Networks

by

Haixia Peng

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Abraham Fapojuwo Professor, University of Calgary
Supervisor(s):	Xuemin (Sherman) Shen Professor, University of Waterloo
Internal Member:	Sagar Naik Professor, University of Waterloo
Internal Member:	Xiaodong Lin Associate Professor, University of Guelph
Internal-External Member:	Yaoliang Yu Assistant Professor, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Benefiting from advances in the automobile industry and wireless communication technologies, the vehicular network has been emerged as a key enabler of intelligent transportation services. Allowing real-time information exchanging between vehicle and everything, traffic safety and efficiency are significantly enhanced, and ubiquitous Internet access is enabled to support new data services and applications. However, with more and more services and applications, mobile data traffic generated by vehicles has been increasing and the issue on the overloaded computing task has been getting worse. Because of the limitation of spectrum and vehicles' on-board computing and caching resources, it is challenging to promote vehicular networking technologies to support the emerging services and applications, especially those requiring sensitive delay and diverse resources. To overcome these challenges, in this thesis, we propose a new vehicular network architecture and design efficient resource management schemes to support the emerging applications and services with different levels of quality-of-service (QoS) guarantee.

Firstly, we propose a multi-access edge computing (MEC)-assisted vehicular network (MVNET) architecture that integrates the concepts of software-defined networking (SDN) and network function virtualization (NFV). With MEC, the interworking of multiple wireless access technologies can be realized to exploit the diversity gain over a wide range of radio spectrum, and at the same time, vehicle's computing/caching tasks can be offloaded to and processed by the MEC servers. By enabling NFV in MEC, different functions can be programmed on the server to support diversified vehicular applications, thus enhancing the server's flexibility. Moreover, by using SDN concepts in MEC, a unified control plane interface and global information can be provided, and by subsequently using this information, intelligent traffic steering and efficient resource management can be achieved.

Secondly, under the proposed MVNET architecture, we propose a dynamic spectrum management framework to improve spectrum resource utilization while guaranteeing QoS requirements for different applications, in which, spectrum slicing, spectrum allocating, and transmit power controlling are jointly considered. Accordingly, three non-convex network utility maximization problems are formulated to slice spectrum among base stations (BSs), allocate spectrum among vehicles associated with the same BS, and control transmit powers of BSs, respectively. Via linear programming relaxation and first-order Taylor series approximation, these problems are transformed into tractable forms and then are jointly solved by a proposed alternate concave search algorithm. As a result, optimal spectrum slicing ratios among BSs, optimal BS-vehicle association patterns, optimal fractions of spectrum resources allocated to vehicles, and optimal transmit powers of BSs are obtained. Based on our simulation, a high

aggregate network utility is achieved by the proposed spectrum management scheme compared with two existing schemes.

Thirdly, we study the joint allocation of the spectrum, computing, and caching resources in MVNETs. To support different vehicular applications, we consider two typical MVNET architectures and formulate multi-dimensional resource optimization problems accordingly, which are usually with high computation complexity and overlong problem-solving time. Thus, we exploit reinforcement learning to transform the two formulated problems and solve them by leveraging the deep deterministic policy gradient (DDPG) and hierarchical learning architectures. Via off-line training, the network dynamics can be automatically learned and appropriate resource allocation decisions can be rapidly obtained to satisfy the QoS requirements of vehicular applications. From simulation results, the proposed resource management schemes can achieve high delay/QoS satisfaction ratios.

Fourthly, we extend the proposed MVNET architecture to an unmanned aerial vehicle (UAV)-assisted MVNET and investigate multi-dimensional resource management for it. To efficiently provide on-demand resource access, the macro eNodeB and UAV, both mounted with MEC servers, cooperatively make association decisions and allocate proper amounts of resources to vehicles. Since there is no central controller, we formulate the resource allocation at the MEC servers as a distributive optimization problem to maximize the number of offloaded tasks while satisfying their heterogeneous QoS requirements, and then solve it with a multi-agent DDPG (MADDPG)-based method. Through centrally training the MADDPG model offline, the MEC servers, acting as learning agents, then can rapidly make vehicle association and resource allocation decisions during the online execution stage. From our simulation results, the MADDPG-based method can achieve a comparable convergence rate and higher delay/QoS satisfaction ratios than the benchmarks.

In summary, we have proposed an MEC-assisted vehicular network architecture and investigated the spectrum slicing and allocation, and multi-dimensional resource allocation in the MEC- and/or UAV-assisted vehicular networks in this thesis. The proposed architecture and schemes should provide useful guidelines for future research in multi-dimensional resource management scheme designing and resource utilization enhancement in highly dynamic wireless networks with diversified data services and applications.

Acknowledgements

Pursuing a second Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Waterloo is definitely the bravest but also the least regretful decision I have made. This decision took me extra four years yet brings me closer to my dream career. During the past four years, my persistence in the Ph.D. study and thesis research can never leave great encouragement and support from lots of people, including my parents, my old brother, my supervisor, my examining committee, my lovely friends, and so on. I would like to take this opportunity to express my sincere thanks to all these beautiful and kind people.

Foremost, I would like to express my deep sense of thanks and gratitude to my supervisor, Prof. Xuemin (Sherman) Shen, for his continuous support of my Ph.D. study and research, his motivation, encouragement, patience, and immense knowledge. He is the first one who supports me to do the second Ph.D. degree and also one of the most important people who help me to finish it. His timely guidance, meticulous scrutiny, and scholarly advice helped me in all the time of thesis research and writing. His great understanding of all his students and valuable advice on both academia and life all aspire me to become a better researcher and person, also encourage me to be a good teacher like he does. I feel very fortunate and honored to be one of his students.

I would also like to thank Prof. Sagar Naik, Prof. Xiaodong Lin, Prof. Yaoliang Yu, and Prof. Abraham Fapojuwo for serving my examining committee, and Prof. Fakhri Karray for attending my proposal exam. Their valuable comments and insightful questions have significantly improved the quality of my thesis and guided me to clearly and logically present research works. Their professional guidance and unintentional goodwill, such as Prof. Naik's deliberate slowing down of speech, Prof. Lin's encouragement, Prof. Yu told me to "calm down" before my background exam, Prof. Karray's recommendation, and so on, all encourage me to finish this thesis.

I would also like to thank Prof. Weihua Zhuang and her two students, Prof. Qiang Ye and Dr. Khadige Abboud, for helping me to build a rigorous academic attitude and for their unlimited patience in improving my paper organization and English writing. I have learned from them how to be patient in a research topic and how to improve myself to become a professional researcher, which greatly benefits my research in this thesis.

I would also like to thank Prof. Ning Zhang, Prof. Haibo Zhou, Prof. Nan Cheng, Prof. Kuan Zhang, Prof. Geoffrey Li, Dr. Le Liang, Weiting Zhang, Dairu Han, Huaqing Wu, Conghao Zhou, and Hongli He for their great help in both my research and life. I am also grateful for all the time spent with Meilin Gao, Yuhao Zhang, Dongxiao Liu, Jianbing Ni, Cheng Huang, Prof. Nan Chen, Prof. Yujie Tang, Amr Salah Matar, Yanglong Sun, Prof. Wenjuan Tang, Prof. Xiaoming Yuan, Jiayin Chen, Dr. Junling Li, Mushu Li, Liang Xue, Xiaoyue Xu, Dr. Scott

Chen, Jingjing Wang, Kaige Qu, and all current and former BBCR members. Thanks all of you for your great support and precious friendship. I feel very lucky to have all the supportive and lovely you in my life.

Finally, I would like to show my deep and sincere thank to my great parents and my old brother for their endless love, understanding, and encouragement. Thank you so much for trusting and supporting me all the time even though you may hard to understand my decisions and choices, and for providing me a warm and forever haven.

Haixia Peng
March 1, 2021
Waterloo, Ontario, Canada

Dedication

This Ph.D. dissertation is dedicated to my beloved parents,

my old brother, and my grandparents.

Table of Contents

List of Figures **xiii**

List of Tables **xv**

List of Abbreviations **xvi**

1 Introduction **1**

1.1 Overview of Vehicular Networks 1

1.1.1 Vehicular Network Applications 2

1.1.2 Vehicular Network Characteristics 3

1.1.3 Vehicular Network Classifications 5

1.1.4 Communication Technologies 7

1.2 Challenges in Vehicular Networks 11

1.2.1 Challenges in Communication 11

1.2.2 Challenges in Computing 13

1.3 Motivation and Contribution 13

1.4 Thesis Outline 16

2 Background and Literature Survey **17**

2.1 Resource Management in Vehicular Networks 17

2.1.1 Spectrum Resource Management 17

2.1.2 Computing Resource Management 20
2.1.3 Multi-Resource Management 22
2.2 Multi-Resource Management Approaches 24
2.2.1 Optimization Approaches 24
2.2.2 Learning-based Approaches 25
2.3 Summary 26

3 An MEC-Assisted Vehicular Network Architecture 27
3.1 An MEC-Assisted ADVNET Architecture 27
3.1.1 Problem Statement 28
3.1.2 Proposed MEC-Assisted ADVNET Architecture 29
3.2 SDN-Enabled Resource Management 32
3.2.1 Computing and Storage Resource Management 32
3.2.2 Bandwidth Management 33
3.3 Open Research Issues 35
3.4 Summary 37

4 Spectrum Resource Management in MEC-Assisted ADVNETs 38
4.1 Background Information 38
4.2 System Model 40
4.2.1 Dynamic Spectrum Management Framework 40
4.2.2 Communication Model 41
4.3 Resource Management Scheme 43
4.3.1 Spectrum Resource Allocation 43
4.3.2 Transmit Power Control 46
4.4 Problem Analysis and Suboptimal Solution 47
4.4.1 Problem Analysis 47
4.4.2 Algorithms Design 52
4.5 Simulation Results 53
4.6 Summary 60

5 Multi-Dimensional Resource Management in MVNETs **61**
5.1 Background Information 61
5.2 System Model 63
5.2.1 Spectrum Management Framework 63
5.2.2 Communication Model 65
5.2.3 Computing and Caching Resource Allocation Models 67
5.3 Problem Formulation and Transformation 69
5.3.1 Problem Formulation 69
5.3.2 Problem Transformation with DRL 70
5.4 DDPG Algorithm based Solution 72
5.4.1 DDPG-based Algorithm 74
5.4.2 HDDPG-based Algorithm 78
5.5 Simulation Results and Analysis 79
5.6 Summary 85

6 Multi-Dimensional Resource Management in UAV-Assisted MVNETs **87**
6.1 Background Information 87
6.2 System Model and Problem Formulation 89
6.2.1 UAV-Assisted MVNET 89
6.2.2 Resource Management Model 89
6.2.3 Problem Formulation 93
6.3 MADDPG-based Resource Management Scheme 95
6.3.1 Problem Transformation 95
6.3.2 MADDPG-based Solution 98
6.4 Simulation Results 102
6.5 Summary 109

7 Conclusions and Future Works **110**

7.1 Main Research Contributions 110

7.2 Future Works 111

References **114**

List of Publications **129**

List of Figures

1.1 Examples for sharing event-driven safety information. 2

1.2 Illustration of a heterogeneous vehicular network in an urban area. 11

3.1 An MEC-assisted ADVNET architecture. 29

3.2 Logically-layered structure for an MEC server. 31

3.3 Diagramming for formulating optimization problem of computing/storage resources. 34

4.1 A dynamic spectrum management framework. 40

4.2 Comparison of network throughput vs. aggregate spectrum resources under the same AV distribution with AV density 0.05AV/m. 56

4.3 Average network throughput vs. p (AV density is 0.05AV/m). 57

4.4 Average network throughput vs. AV density. 58

4.5 Spectrum slicing ratios under different AV density for the proposed scheme with $p = 0.8$. 59

5.1 Dynamic spectrum management frameworks. 64

5.2 The fundamental DRL architecture in the MVNET. 70

5.3 The architecture of the DDPG learning. 73

5.4 The total rewards of each episode in scenario with an MeNB-mounted MEC server. 82

5.5 The total rewards of each episode in scenario with an EN-mounted MEC server. . 83

5.6 Average delay/QoS satisfaction ratio over the vehicles under the service area of the MeNB-mounted MEC server. 85

5.7 Average delay/QoS satisfaction ratio over vehicles under the service area of the EN-mounted MEC server. 86

6.1 An illustration of the UAV-assisted MVNET. 90

6.2 A simplified UAV-assisted MVNET scenario for multi-dimensional resource management. 91

6.3 The MADDPG framework in the UAV-assisted MVNET. 100

6.4 Rewards achieved per episode in the training stage. 105

6.5 The averaged Q-value over each episode under the MADDPG algorithm. 106

6.6 The average delay/QoS satisfaction ratios achieved by different schemes versus the amounts of spectrum, computing, and caching resources at the MeNB (subfigures (a), (b), and (c)) and UAVs (subfigures (d), (e), and (f)). 108

List of Tables

1.1 Six distinct autonomy levels of on-road vehicles. 6

4.1 Parameters values. 55
4.2 Optimal transmit powers and number of iterations for the three schemes ($p = 0.8$). 59

5.1 Parameters for the learning stage. 81

6.1 Parameters for the learning stage. 104

List of Abbreviations

MEC	Multi-access Edge Computing
UAV	Unmanned Aerial Vehicle
MVNETs	MEC-assisted Vehicular Networks
MANETs	Mobile Ad Hoc Networks
MDVNETs	Manual Driving Vehicular Networks
ADVNETs	Automated Driving Vehicular Networks
HDVNETs	Heterogeneous Driving Vehicular Networks
AI	Artificial Intelligence
DDPG	Deep Deterministic Policy Gradient
MADDPG	Multi-Agent DDPG
SADDPG	Single-Agent DDPG
HDDPG	Hierarchical DDPG
MDPs	Markov Decision Processes
DPG	Deterministic Policy Gradient
DQN	Deep Q-Network
RL	Reinforcement Learning
DRL	Deep RL
DNN	Deep Neural Network
V2X	Vehicle-to-Everything
V2V	Vehicle-to-Vehicle
V2P	Vehicle-to-Person
V2I	Vehicle-to-Infrastructure
I2I	Infrastructure-to-Infrastructure
V2C	Vehicle-to-Cloud

QoS	Quality of Service
AVs	Autonomous Vehicles
HD	High Definition
SAE	Society of Automotive Engineers
OBU	On-Board Unit
AUs	Application Units
DDT	Dynamic Driving Task
ODD	Operational Design Domain
ADS	Automated Driving System
DSRC	Dedicated Short Range Communications
VLC	Visible Light Communication
ITS	Intelligent Transportation System
FCC	Federal Communications Commission
CCH	Control Channel
SCHs	Service Channels
BSs	Base Stations
eNBs	evolved NodeBs
LTE	Long Term Evolution
3GPP	3rd Generation Partnership Project
D2D	Device-to-Device
mmWave	millimeter-wave
5G	5th Generation
WLANs	Wireless Local Area Networks
APs	Access Points
FCW	Forward Collision Warning
SDN	Software-Defined Networking
NFV	Network Function Virtualization
VNFs	Virtual Network Functions
VMs	Virtual Machines
RRM	Radio Resource Management
RAN	Radio Access Network
CR	Cognitive Radio
MINLP	Mixed-Integer Nonlinear Programming

DMIPS	Dhrystone Million Instructions Executed Per Second
RSUs	Road Side Units
VR	Virtual Reality
RRHs	Remote Radio Heads
BBUs	BaseBand Units
AWGN	Additive White Gaussian Noise
SINR	Signal to Interference plus Noise Ratio
ACS	Alternate Concave Search
max-utility	maximization-utility
max-SINR	maximization-SINR
MBS	Macro-cell BS
EN	Edge Node
OFDM	Orthogonal Frequency-Division Multiplexing
NOMA	Non-Orthogonal Multiple Access
MIMO	Multiple-Input-Multiple-Output

Chapter 1

Introduction

In the last decade, vehicular communication networks have attracted tremendous interest not only from academia and industry but also from governments. Via enabling vehicles to exchange information with everything, i.e., vehicle-to-everything (V2X) communications, including vehicle-to-vehicle (V2V), vehicle-to-person (V2P), vehicle-to-infrastructure (V2I), infrastructure-to-infrastructure (I2I), and vehicle-to-cloud (V2C), a variety of applications and data services can be supported, such as intelligent transportation and safety management [1, 2]. However, due to high vehicle mobility and the complicated communication environment, it is very challenging to provide efficient and reliable vehicular communications to satisfy the heterogeneous quality of service (QoS) requirements of different applications, especially, the sensitive and lower latency requirements. In this chapter, we first give out an overview of vehicular networks, including applications, characteristics, classifications, and available communication technologies of vehicular networks. Then, challenges faced by vehicular networks in communication and computing perspectives are discussed. Finally, we summarize the motivation and main contribution of this thesis.

1.1 Overview of Vehicular Networks

In this section, we will first introduce the applications of vehicular networks and describe the unique characteristics of vehicular networks. Then, different types of vehicular networks are outlined according to the driving patterns and communication technologies available for vehicular communications are summarized. Note that in-vehicle communications, which refer to wired or wireless communications between an on-board unit (OBU) and one or multiple

application units (AUs) in a vehicle [3], are not considered here. Moreover, the mentioned autonomous vehicles (AVs) should be distinguished from autonomous robots, unmanned aerial vehicles (UAVs), and unmanned underwater vehicles.

1.1.1 Vehicular Network Applications

For moving vehicles, communication networks are usually designed for sharing information and supporting a large number of cooperative applications, which can be categorized into safety and non-safety applications.

Safety applications: Through sharing safety-related information, safety services can be provided, the number of traffic accidents can be significantly reduced, and commuters' life, health, and property can be effectively protected. Once obtaining safety-related information from other vehicles, drivers can take actions in advance to enhance driving safety or be informed about unexpected dangerous situations to avoid traffic accidents [4]. One type of safety-related information is the vehicle's driving state information, such as real-time position, speed, acceleration, and direction. This type of information is not only important to assist drivers or automated driving systems in passing and changing lanes and avoiding collision, but also a necessary condition for cooperative driving among vehicles to maintain the string stability of platoons/convoys [5]. Another type of safety information is event-driven safety information, e.g., emergency vehicle warning, traffic condition warning, work zone warning, lane change warning, rear-end collision warning, and so on. Event-driven safety information, generated by certain vehicles involved in or discovering a dangerous situation (such as an emergency brake or sudden lane change), should be shared to help other vehicles obtain real-time situational awareness and detect possible dangers. As shown in Figure 1.1, sharing cooperation collision and rear-end collision warning information among vehicles can help to avoid accidents in several scenarios.

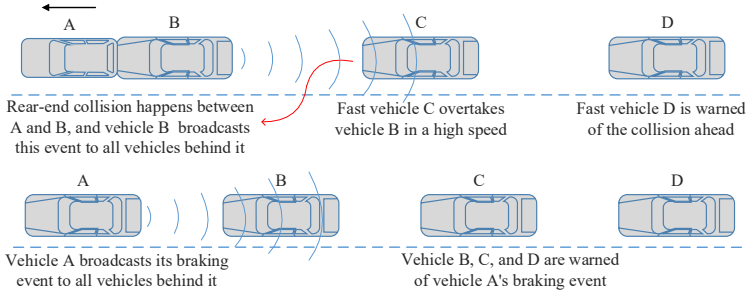


Figure 1.1: Examples for sharing event-driven safety information.

Non-safety applications: By sharing information among moving vehicles, value-added services, such as traffic management and infotainment support, can be provided to enhance the comfort of commuters. Similar to some safety applications, most of the traffic management applications are designed for reducing traffic jams to improve traffic flow and save travel time for commuters. For example, via sharing information about traffic monitoring and road conditions among moving vehicles, traffic management applications can be applied to help drivers to reroute to their destinations and to improve the efficiency of traffic light schedules, and consequently reduce traffic jams [6]. Also, enabling "platooning" or "convoying" with automated driving [7], can not only reduce energy consumption and exhaust emission by minimizing air drag due to the streamlining of the vehicle but also increase the driving safety by cooperative driving among vehicles [8]. Different from traffic management, infotainment-support applications are mainly focusing on providing traveler location-based services and entertainments. For instance, infotainment-support applications can provide location information, such as location information about fuel station, parking, restaurant, and hotel, to moving vehicles when the related services are required by drivers or passengers. Furthermore, infotainment-support applications can also provide Internet access for moving vehicles to download multimedia entertainment information or to experience virtual reality.

Despite the benefits of the above applications, the implementation of safety/non-safety applications faces challenges. In a vehicular network with certain communication technology, all available spectrum resources can be used for either safety or non-safety applications. For safety applications, their related information is with high priority in terms of transmission delay and reliability so that drivers can receive them and take the corresponding actions in time [9]. Moreover, due to high mobility, time-varying vehicle density, and unstable network topology in vehicular networks, meeting these requirements is sometimes very challenging [10, 11]. Thus, safety applications are usually given higher priority over non-safety applications. Different from safety applications, even though most of the basic non-safety applications do not have stringent real-time requirements. However, some new emerging applications, such as automated driving and virtual reality, are also with sensitive delay requirements [7]. How to reduce the delay and packet loss for non-safety information without impacting safety applications is important to improve the service quality of the non-safety applications, especially for some infotainment and delay-sensitive applications.

1.1.2 Vehicular Network Characteristics

In vehicular networks, V2V communications are basically executed by applying the principles of mobile ad hoc networks (MANETs), namely, wireless communications are spontaneously created for data exchange. In addition to some characteristics similar to MANETs [12], such

as self-organization and management, short to medium transmission range, omnidirectional broadcast, and low bandwidth, a vehicular network has its own unique characteristics due to high vehicle mobility. According to whether they are beneficial to information exchange or not, these characteristics are classified into detrimental and beneficial ones.

Detrimental characteristics: These characteristics, including high mobility, stringent delay constraints, complicated communication environments, and connected and/or autonomous vehicles, pose obstacles or challenges to vehicular networks.

1) High mobility: High moving speed of vehicles often results in frequently disconnected wireless communication links and then reduces the amount of effective communication time among vehicles. Furthermore, it also causes the network topology to change dynamically and further adds challenges to information exchange among vehicles.

2) Heterogeneous QoS requirements [13]: In some vehicular network applications, such as safety applications and some infotainment applications, information exchange is required to be successfully finished within a particular time to avoid traffic accidents and ensure the quality of infotainment services. Stringent delay is required for information exchanging to support such kinds of applications. Note that the delay mentioned here is the maximum delay from the source to the destination, not the average delay in vehicular networks. In addition to delay, high transmission rate is another important QoS requirement to some vehicular applications, such as video streaming related services and online games.

3) Complicated communication environment: Vehicular networks are usually applied in three kinds of communication environments. The first one is a one-dimensional communication environment, such as highway traffic scenarios. Even though vehicles on highways always move faster than in other types of environments, the one-dimensional environment is relatively simple due to the straightforward moving direction and relatively fixed driving speed. The second one is a two-dimensional communication environment. A typical example is the urban traffic scenario, which is more complex compared with the highway scenario [14]. Streets in most urban areas are often divided into many segments due to intersections. Therefore, for two moving vehicles in different road segments, a direct communication link may not exist due to obstacles around the intersections, such as buildings and trees. Moreover, the vehicle density is relatively high in urban areas, which implies more communication links within the communication range and significantly impacts the spectrum resource occupation probability. The last one is a three-dimensional communication environment, such as viaducts [15]. For vehicles in a viaduct, communication links in different physical space layers make this type of environment the most complex.

4) Connected and/or autonomous vehicles: With the development of sensing capability and automobile industry, AVs are emerging in our real life and would show up on the road with

manually driving vehicles. In addition to some traditional safety and non-safety applications, there are some new applications are emerging to support automated driving, such as the processing of sensing data and video streaming.

Beneficial characteristics: These characteristics are beneficial to the wireless communications in vehicular networks, such as weak energy constraints and driving route prediction.

1) Weak energy constraints: Since vehicles always have sufficient energy, vehicular networks do not suffer from power constraints as in regular mobile communication networks. Moreover, vehicles can afford significant sensing and computing demanding (including data storage and processing) since the sensing and communication devices can power themselves while providing continuous communications with other vehicles.

2) Driving route prediction: A vehicle is limited to moving on the road in usual circumstances, which makes it possible to predict the driving route for itself or even for other vehicles when the road map and speed information are available. Driving route prediction plays an important role in routing protocol design for vehicular networks, especially when addressing the challenges presented by high vehicle mobility.

1.1.3 Vehicular Network Classifications

Owing to the advances in sensor technologies, wireless communications, computational power, and intelligent control, a new driving pattern, named automated driving, has been gradually applied in vehicles. According to the society of automotive engineers (SAE) International's standard J3016 [16], vehicles can be classified into six distinct levels of autonomy, as shown in Table 1.1. In this thesis, vehicles with level 0 and 1 autonomy are referred to as manually driving vehicles, and those with level 4 and 5 autonomy are referred to as AVs. Accordingly, vehicular networks are classified into three categories, manually driving vehicular networks (MDVNETs), automated driving vehicular networks (ADVNETs), and heterogeneous driving vehicular networks (HDVNETs).

Manually driving vehicular networks: An MDVNET is defined as a wireless network to enable communications among manually driving vehicles. Almost all current vehicles are manually driving and always move individually on the roads. For example, some drivers may accelerate suddenly to pass other vehicles and some may get used to low speeds. Thus, the impacts of high and heterogeneous mobility on the MDVNETs are significant. How to address the challenges caused by high and heterogeneous mobility and enable communications in different types of communication environments has attracted more and more attention and has also been widely considered in the existing works. MDVNETs can help manually driving

Table 1.1: Six distinct autonomy levels of on-road vehicles.

Autonomy levels	**Narrative definition**
0 - No Driving Automation	The entire dynamic driving task (DDT) is performed by the driver, even when the DDT is enhanced by active safety systems.
1 - Driver Assistance	The driving automation system executes the sustained and specific operational design domain (ODD) of either the lateral or longitudinal vehicle motion control sub-task, while the driver is expected to perform the remainder of the DDT.
2 - Partial Driving Automation	The automated driving system (ADS) executes the sustained and specific ODD of both the lateral or longitudinal vehicle motion control sub-task, and the driver completes the object and event detection/response sub-task and supervises the driving automation system.
3 - Conditional Driving Automation	The sustained and specific ODD is performed by the ADS, while the DDT fallback-ready user is receptive to intervene requests from ADS and responds appropriately to DDT performance-relevant system failures in other vehicle systems.
4 - High Driving Automation	The sustained and specific ODD is performed by ADS of the entire DDT and DDT fallback, and a user has no need to respond to intervene requests.
5 - Full Driving Automation	The sustained and unconditional ODD is performed by an ADS of the entire DDT and DDT fallback, and a user has no need to respond to intervene requests.

vehicles to improve traffic safety and provide infotainment services to drivers and passengers [10, 17].

Automated driving vehicular networks: It is expected that no human actions or interventions are required for an AV that can automatically navigate a variety of environments, other than setting the destination and starting the system. In other words, people can be relieved from driving stress [7, 18]. Moreover, AVs are considered as a good solution for increased safety, velocity, convenience, and comfort with reduced energy consumption [19, 20]. Despite the attractive advantages of AVs, how to ensure the automated driving system to be safe enough to completely leave humans' actions and interventions presents significant challenges. Being "safe" means at least the AV can correctly execute vehicle-level behaviors, such as obeying traffic regulations and dealing with road and roadside hazards [18, 20]. On the other hand, several works indicate that inter-vehicle information plays an important role for AVs' "safe", in which, helpful information, either via on-board sensors or vehicular communications, is needed for collision avoidance and cooperative driving among AVs [21]. Therefore, it enables advanced features, such as "platooning", and alerts AVs of real-time mapping information and surrounding environments, such as other AVs and potential hazards [22]. Thus, ADVNETs, used for wireless communications among AVs, have been regarded as another important application of vehicular networks.

Heterogeneous driving vehicular networks: In addition to MDVNETs and ADVNETs, another important category of vehicular networks is HDVNETs. Recently, vehicles with level 2 and 3 autonomy have been generated and begun to be test-driven on the roads. Table 1.1 shows that vehicles with level 2 and 3 autonomy may be completely controlled either by the ADS or the driver in different road environments. Moreover, with AVs gradually prevailing on roads, scenarios, where AVs and manually driving vehicles move on roads simultaneously, will be more common. To achieve information sharing between manually driving vehicles and AVs or among vehicles with level 2 and 3 autonomy, HDVNETs are essential. In this thesis, unless specifically stated, a "vehicular network" can be either an MDVNET, an ADVNET, or an HDVNET. Namely, vehicles considered in the vehicular network can be any vehicles with level 0 to 5 autonomy.

1.1.4 Communication Technologies

To support the wireless communications in vehicular networks, several communication technologies have been considered, including dedicated short range communications (DSRC), cellular [23], Wi-Fi [24], and White-Fi [25], and other short-range communication technologies, such as infrared and visible light communication (VLC). Among them, most of the vehicular communications are usually supported by the DSRC and cellular technologies, such as in

[23]. In what follows, we summarize the advantages, disadvantages, and challenges of these technologies.

DSRC: It is a dedicated wireless communication technology used for exchanging information between vehicle and everything over short to the medium range [11]. DSRC is based on the IEEE 802.11p, which is amended from the IEEE 802.11 Wi-Fi standard. As the only communication technology specifically designed for vehicular users, DSRC can provide

1) Designated licensed bandwidth: 75 MHz radio spectrum in the 5.9 GHz band was allocated to support the DSRC-based communications in intelligent transportation system (ITS) applications by the Federal Communications Commission (FCC) of the United States;

2) High reliability: DSRC-based wireless links can work in high mobility and harsh weather conditions, such as rain, fog, and snow;

3) Priority for safety applications: The total 75 MHz bandwidth is divided into one control channel (CCH) and six service channels (SCHs). Among these seven channels, safety applications are given priority over non-safety applications [26];

4) Security and privacy: Message authentication and privacy are provided [27].

Thanks to the above benefits, DSRC is regarded as a promising technology applied to support ITS applications, especially the safety-related ones [28]. However, DSRC also exposes some drawbacks. First, due to the limited spectrum resource, a broadcast storm may occur when disseminating safety information over a large area, especially in situations with high vehicle density. With the increase in the number of vehicles attempting to transmit in the same channel simultaneously, the packet delay and transmission collision probability will increase and the performance of DSRC will degrade. Another obvious drawback in DSRC communications is poor and short-lived V2V and V2I connectivity. Short-lived V2V connectivity always occurs in an environment with low vehicle density, where the number of vehicles is too sparse to disseminate the information to all destination vehicles. Furthermore, due to the short radio transmission distance, i.e., around 300 m, DSRC can only provide short-lived V2I connectivity [29] if there is no pervasive roadside communication infrastructure. In order to improve the performance of vehicular networks, some medium and long-range communication technologies [23] can be commonly used.

Cellular: Nowadays, cellular networks are distributed over land areas, where each cell is served by a base station (BS), e.g., the evolved nodeBs (eNBs) in the long term evolution (LTE) system. The key enabler of cellular-based vehicular networks is the LTE standard developed by the 3rd generation partnership project (3GPP), which provides efficient information

dissemination to user equipment [23]. Lots of academic research and field tests have indicated that cellular technologies, such as LTE technologies, possess great advantages in vehicular networks [29, 30]. Specifically, benefited from the large coverage area of the eNB and high penetration rate, cellular technologies can provide relatively long-lived V2I connectivity [29]. Compared with other communication technologies, cellular technologies can potentially support several vehicles within a small region simultaneously due to their relatively high capacity. Furthermore, the channel and transport modes in cellular technologies, i.e., the dedicated/common modes and the unicast/broadcast/multicast downlink transport modes, can help to reduce the transmission delay and improve the capacity for a communication environment with high vehicle density. Device-to-device (D2D) communications can provide short range direct links between two vehicles to reuse the spectrum, and therefore mitigate the problems caused by the limited radio spectrum resources [31].

Recently, cellular-based vehicular networks have been widely investigated [29, 31, 32]. Due to the above mentioned advantages, cellular technology is regarded as a promising alternative to DSRC for vehicular networks. However, due to the current cellular data pricing model, the corresponding cost for data transmission in cellular-based vehicular networks is much higher than other communication technologies [33]. On the other hand, in dense traffic areas, the heavy data traffic-load generated by vehicles may significantly challenge the cellular capacity and potentially affect the delivery of traditional cellular data. To address this challenge, millimeter-wave (mmWave) communications with advantages of multi-gigabit transmit ability and beamforming technique have been considered for the 5th generation (5G) or beyond networks. For example, millimeter-wave communications are applied for sharing vehicles' massive sensing data in [34], where the beam alignment overhead has been reduced by configuring the mmWave communication links based on sensed or DSRC-based information.

Wi-Fi: It is a technology for wireless local area networks (WLANs) based on the IEEE 802.11 standards. It has been shown in [33, 35] that Wi-Fi technology is an attractive and complimentary Internet access method for moving vehicles. Equipped with a Wi-Fi radio or Wi-Fi-enabled mobile devices, such as mobile phones, vehicles can access the Internet when they drive through the coverage of Wi-Fi access points (APs). The obvious advantages of Wi-Fi technology include low per-bit cost, extremely widespread global deployments, and higher peak throughput, which are beneficial to some vehicular applications requiring high data transmission rates, such as infotainment applications. However, due to the limited coverage of each Wi-Fi AP and high vehicle mobility, the Wi-Fi technology suffers from intermittent connectivity in vehicular networks. Thus, handoff schemes become particularly important to Wi-Fi technology in such scenarios. Furthermore, instead of establishing Wi-Fi-based inter-vehicle communications, Wi-Fi technology is considered as a complementary access method to offload delay tolerant data traffic [33].

White-Fi: It is a term coined by the FCC of the United States to describe communications that allow unlicensed users to access the TV white space spectrum in the VHF/UHF bands between 54 and 790 MHz. Note that even though White-Fi is also referred to as super Wi-Fi, it is not endorsed by the Wi-Fi Alliance or based on Wi-Fi technology. The progress of White-Fi technology has yielded many new insights into vehicular networks, which has motivated researchers to explore unlicensed spectrum to solve the spectrum-scarcity issue for vehicular networks. It has been shown in [25] and [36] that the White-Fi enabled vehicular networks can improve the dissemination capacity by offloading a portion of data traffic from the DSRC band or cellular band to the TV band. Furthermore, different from the 2.4 GHz radio frequency used by Wi-Fi, TV white space spectrum is at a lower frequency range and allows the signal to penetrate walls better and travel farther than the higher frequency range. Thus, White-Fi technology can provide relatively long range communications to improve transmission efficiency. For example, applying White-Fi for long-distance dissemination to avoid multi-hop transmission can reduce the transmission delay of some safety-related information [37]. However, White-Fi enabled vehicular communications generate potential interference to incumbent TV band users, which may bring challenges to protect the incumbent services. Moreover, due to the unlicensed characteristic of TV bands, vehicular networks and other existing wireless networks are all allowed to co-exist. Vehicle users may experience interference caused by other networks, and therefore impacting the service quality of some vehicular applications [38].

Multiple technologies interworking: It has been shown that single technology applied in vehicular networks always has its own limitations as discussed before. The aforementioned advantages and disadvantages of DSRC, cellular, Wi-Fi, and White-Fi have motivated the works on establishing heterogeneous vehicular networks [39, 40]. In which, vehicular communications are supported by at least two types of communication technologies, an example of a heterogeneous vehicular network in an urban area is illustrated in Figure 1.2. A typical heterogeneous vehicular network is the interworking of the DSRC and cellular technologies, where cellular-based communications can act as 1) a backup for traffic data when DSRC-based multihop links are disconnected in sparse vehicles situations, 2) a long-range Internet access method, and 3) a powerful backbone network for control message dissemination [39]. Even though multiple technologies interworking can make the best use of the advantages and bypass the disadvantages of every single technology, how to select the applicable technology for each communication link and achieve seamless handoff among different technologies is still challenging.

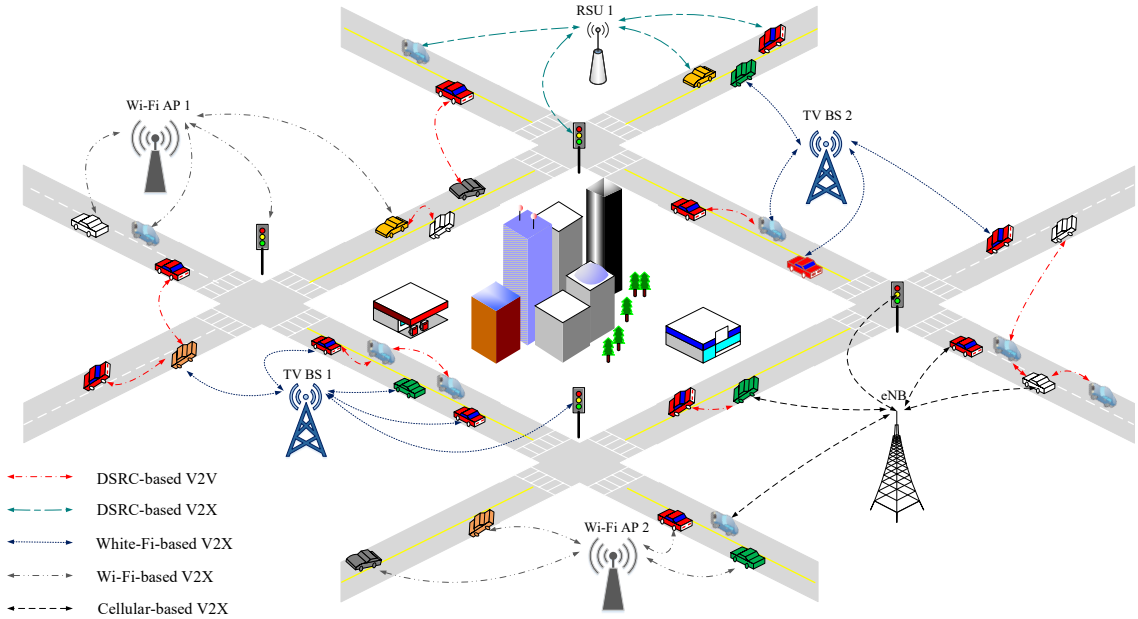


Figure 1.2: Illustration of a heterogeneous vehicular network in an urban area.

1.2 Challenges in Vehicular Networks

As we discussed in Section 1.1, the vehicular network has been emerged as a key enabler of ITS and can significantly enhance traffic safety and efficiency and enable ubiquitous Internet access to support new data services and applications. However, how to guarantee the performance of the vehicular network to achieve its important role as expected by industry and academic still faces challenges. In this section, we will investigate the major challenges faced by vehicular networks from both communication and computing perspectives.

1.2.1 Challenges in Communication

In vehicular networks, demanding for wireless communications is mainly from two respects, i.e., sharing safety-related and non-safety related information between vehicle and everything. One type of safety-related information is vehicle's driving state information, including real-time position, speed, acceleration, and moving direction information, which is not only required to assist drivers or automated driving systems in passing and changing lane and avoiding collision but also a necessary condition for cooperative driving among AVs [40]. For example, a beacon,

including the vehicle's kinematics information, is broadcasted periodically from each vehicle to avoid an upcoming collision or potential danger [17]. To enable the cooperative adaptive cruise control for string stability, the speed and acceleration information of the leader vehicle and its preceding vehicle is required for each member vehicle in a platoon [41]. In some communication-assisted platoon control schemes [42], a member vehicle is required to share its braking or leaving information with the leader vehicle, and then the leader vehicle makes accelerating or braking decision for and sends the decision to that member vehicle. Another type of safety information is event-driven safety information, e.g., emergency vehicle warning, traffic condition warning, cooperation collision warning, and rear-end collision warning. For instance, to avoid rear-end collision, a forward collision warning (FCW) system has been proposed in [43]. By sharing the front vehicle's driving intention and other driving parameters with the following vehicles, the FCW module then can determine the potential collision risk for each vehicle.

To provide value-added services, such as traffic management and infotainment support, for enhancing the comfort of commuters, sharing some non-safety related information among vehicles is required. For example, via sharing information about traffic monitoring and road conditions among moving vehicles, traffic management applications can be applied to help drivers to reroute to their destinations and to improve the efficiency of traffic light schedules, and consequently reduce traffic jams [44]. On the other hand, sharing static high definition (HD) maps (which illustrate the static road environment, such as the lane lines and the surrounded buildings and trees) and dynamical HD maps (i.e., dynamic environment information, such as driving state information about the adjacent AVs) to AVs to show the road environment to AVs can timely compensate the inaccurate sensing information to improve AV safe navigation [40]. Moreover, information sharing is also required by some new emerging artificial intelligence (AI)-powered applications and services. For instance, to support an AI-powered application designed to reduce the number of accidents that involve some road users, like children, the vehicle needs to keep sending the video stream from its front-facing camera to a multi-access edge computing (MEC) server carried by a base station [45]. And the AI model deployed onto the MEC server then will analyze these video streams to judge whether a dangerous situation is happening or likely to happen, and notify the vehicle of its judgment if needed.

However, with the increase of the number of vehicles and the emerging applications, sharing both safety and non-safety related information among moving vehicles substantially increases the data traffic volume of the vehicular networks. Also, building/updating the HD maps of a road environment, which are then shared with vehicles, leads to a large amount of data traffic, and therefore resulting in increasing demand on spectrum resources. Even though the wireless communication technologies summarized in Subsection 1.1.4 can be applied to support vehicular communications, a single wireless access technology cannot fulfill the communication requirements. Thus, it is critical to enable more communication technologies and improve

the available spectrum resource utilization to guarantee the heterogeneous application QoS requirements in vehicular networks.

1.2.2 Challenges in Computing

From the computing perspective, a huge body of heavy computations needs to be executed on board for various driving tasks in real time, such as video recognitions for detecting the surrounding traffic environment and AI model training to make the immediate driving decisions [46]. For example, high computing and storing capabilities are required by leader vehicles of platoons/convoys to process the aggregated information to make accelerating or braking decisions for all the member vehicles [47]. Also, efficiently handling dynamic driving state information and traffic flow is necessary for timely updating dynamic HD maps at vehicles. Moreover, to judge dangerous situations which may result in accidents, a vehicle can analyze the video streaming obtained from its front-facing camera with an AI model [45]. And a large amount of computing and storage resources is required to support the operation of such AI models and notify the vehicle of the AI model's judgment if needed.

To support the computing tasks generated by vehicles, we can either choose to process such tasks with local computing or cloud computing. Namely, a vehicle can process its tasks with the on-board resources or offload the tasks to the remote cloud computing server or other AVs with spare computing and storage resources. However, there is no on-board resource in most of the manually driving vehicles and the amount of computing and storage resources carried by each AV is limited [46], processing all the computing tasks by vehicles' local computing power requires an external powerful and expensive on-board computing device or physically increasing the amount of computing/storage resources carried by each AV. On the other hand, offloading vehicles' tasks to the cloud computing server can cause a significant data traffic burden on the core network and a high response delay to the offloaded task [48], whereas performing collaborative computing only among vehicles with guaranteed QoS requirements is difficult and with high task migration cost due to high vehicle mobility. How to address these challenges in computing to perform the real-time computing tasks for vehicles in a low-cost, QoS-guaranteed, and efficient approach is the key for vehicular networks.

1.3 Motivation and Contribution

As we mentioned above, benefiting from advances in the automobile industry and wireless communication technologies, the vehicular network has been emerged as a key enabler of

intelligent transportation services [49–51]. Allowing real-time information exchanging between vehicle and everything, traffic safety and efficiency are significantly enhanced, and ubiquitous Internet access is enabled to support new vehicular data services and applications [40, 52]. However, with more and more services and applications, mobile data traffic generated by vehicles has been increasing and the issue on the overloaded computing task has been getting worse [53]. Because of the limitation of spectrum resources and vehicles' on-board computing and storage resources [54], it is challenging to promote vehicular networking technologies to support the emerged services and applications, especially those requiring sensitive delay and diverse resources. To effectively address these challenges, two potential technologies, MEC and UAV [55, 56], have been widely exploited in vehicular networks [57–59].

Considering the stringent delay requirements of some applications and the huge latency on wireless and wired communications between a vehicle and the cloud computing server, offloading vehicles' tasks to the cloud computing server is sometimes inapplicable [59, 60]. Meanwhile, physically increasing the amount of on-board resources would result in a serious increase in the manufacturing cost of vehicles. The MEC server, as an extension of the cloud computing server, shifts computing and caching capabilities close to user devices [61–64], and allows vehicles to offload tasks to it via multiple wireless communication technologies. By saving the time consumption on communications between the MEC and the cloud computing server, the sensitive delay requirement of the task offloaded to an MEC server can be satisfied.

Even though MEC technology can help to provide real-time computing support to vehicular applications, underused or wasted resource issues would happen in scenarios with only terrestrial MEC servers (i.e., MEC servers placed at ground BSs), especially when there exists bursty traffic caused by some social activities or events. The reason is, the amount of resources carried by each MEC-mounted BS is fixed and pre-allocated according to the average resource demanding from vehicles under its service area, yet the resource demanding from vehicles is time-varying due to the high vehicle mobility and dynamic application requirements. Taking the flexibility advantage of UAVs, mounting MEC servers in UAVs can help to address the above issues by dispatching the MEC-mounted UAVs to assist the designated BSs [57, 65]. Related applications have been also considered in different projects launched by many leading companies [66].

To implement MEC- and/or UAV-assisted vehicular networks, many research works have been performed recently, including design of architecture, task offloading scheme, resource management scheme, and so on. Among that, efficient resource management for computing, storage, and spectrum is of paramount importance for such vehicular network scenarios. However, most of the existing schemes proposed for MEC-assisted vehicular networks (MVNETs) only consider one or two dimensions of resources, which cannot be directly adopted to support some vehicular applications where high dimensional resources are involved, such as the computing tasks generated by the leading vehicle for platoon/convoy control [1]. And existing works

focusing on multi-dimensional resources management are usually targeted for the scenarios with low mobility users [67, 68]. For MVNETs, the computational complexity of the multi-dimensional resource management problem increases due to the high vehicle mobility and time-varying demand on resources, and therefore increasing the time consumption on the resource management scheme itself. Therefore, it is infeasible to adopt the pure optimization approach-based schemes to achieve multi-dimensional resource management in MVNETs, especially for the scenarios with delay-sensitive applications. Furthermore, how to design practical and QoS guaranteed multi-dimensional resource management schemes for the MVNETs and UAV-assisted MVNETs still needs effort.

This thesis focuses on addressing the challenges in communication and computing faced by vehicular networks to provide QoS guaranteed services to vehicle users. Specifically, the main contributions of this thesis are summarized as follows,

1. A novel vehicular network architecture by integrating the software-defined networking (SDN) and network function virtualization (NFV) concepts into MEC is proposed [13]. Via MEC, multiple wireless access networks can interwork to support the increased data traffic volume, and quick computing responses can be provided by uploading the computing/storing tasks (e.g., cooperative driving) to the network edge to avoid extra delay for data transmission between MEC servers and the cloud-computing server, and hot content can be cached in and processed by the MEC servers; And, by integrating SDN and NFV concepts with the cloud-computing/MEC servers, unified control plane interfaces are provided by decoupling the control plane from the data plane without placing new infrastructures, and different network functions supporting vehicular applications can be programmed on servers flexibly with reduced provisioning cost;

2. To improve the spectrum resource utilization under the MVNET architecture, we propose a dynamic two-tier spectrum management framework by considering the trade-off between spectrum resource utilization and inter-cell interference [59]. By leveraging logarithmic and linear utility functions, three aggregate network-wide utility maximization problems are formulated to fairly slice spectrum resources among BSs connected to the same MEC server, optimize BS-vehicle association patterns and spectrum allocation, and control the transmit power of BS. And then the optimization technique is used to transform and solve the formulated problems;

3. For service requests demanding multiple types of resources, allocation results for computing, caching, and spectrum resources can impact the service QoS satisfaction. Efficient management for the three types of resources is of paramount importance. Thus, we leverage the deep deterministic policy gradient (DDPG) and hierarchical learning

architecture to achieve the joint optimal allocation of spectrum, computing, and caching resources for the MVNET [54]. Via off-line training, the network dynamics can be automatically learned and appropriate resource allocation decisions can be rapidly obtained to satisfy the QoS requirements of vehicular applications;

4. To address the inefficient resource management issue caused by the pre-placed MEC servers and high vehicle mobility (spatial dynamics of resource demand), in addition to MEC-mounted BSs, aerial computing is also taken into account to enable a UAV-assisted MVNET. In order to avoid time and spectrum cost on wireless communication between the MEC server and the central controller, instead of centralized resource scheduling, we let each MEC server act as an agent and propose a multi-agent DDPG (MADDPG) based solution to enable distributed multi-dimensional resource scheduling for the considered UAV-assisted MVNET [69].

1.4 Thesis Outline

The remainder of this thesis is organized as follows: In Chapter 2, we investigate the background and literature survey related to resource management schemes in vehicular networks. Then, an MEC-assisted architecture is proposed to support vehicular applications in Chapter 3. In Chapter 4, a dynamic spectrum resource management scheme based on optimization method is presented for the MEC-assisted architecture. By adopting deep reinforcement learning (DRL) algorithms, Chapter 5 and Chapter 6 investigate the multi-dimensional resource management issues in the MVNETs with and without aerial computing, respectively. Finally, Chapter 7 concludes the thesis and presents future works.

Chapter 2

Background and Literature Survey

In this chapter, existing works related to resource management in vehicular networks are presented. We first survey the spectrum resource management schemes, computing resource management schemes, and multi-resource management schemes proposed for the vehicular networks, and then, approaches applied for integratedly managing the spectrum, computing, and storage resources are summarized.

2.1 Resource Management in Vehicular Networks

Vehicular networks are expected to accommodate a large number of data-heavy mobile devices and services, whereas they are facing challenges in both communications capacity and computing capability because the mobile data traffic has explosively increased. Investigating spectrum allocation, computation offloading, and content caching to improve the overall resource utilization to address these challenges is essential to the vehicular networks. In this section, we survey existing spectrum resource management schemes, computing resource management schemes, and multi-resource management schemes in vehicular networks.

2.1.1 Spectrum Resource Management

To address the issues caused by the increasing demand for vehicular data transmission and the limited spectrum resources, plenty of existing works have investigated the spectrum resource management problem in vehicular networks. In MVNETs, the spectrum resources available to different types of communication technologies are aggregated, and then sliced among BSs and

allocated among vehicles. The process of spectrum management can be summarized as follows from different time granularities:

1. Long-term spectrum slicing: That is slicing the aggregated spectrum resources among different BSs. Time is partitioned into different slicing windows, where each slicing window is composed of a fixed number of time slots. The spectrum slicing is controlled by the radio resource management (RRM) unit at a central controller and is executed at the beginning of each slicing window. Define the communication workload of a BS as the number of data transmission requests generated by vehicles within its coverage. Based on the communication workloads of different BSs in each slicing window, the controller makes spectrum slicing and communication workload division decisions for the BSs. The amount of available aggregated spectrum resources are sliced with different ratios and allocated to the BSs within the service area of the controller. Furthermore, the communication workload under the overlapping area of two or more BSs is divided into several parts with the partitioning ratios and then assigned to each BS.

2. Short-term spectrum allocation: This is operated at the beginning of each time slot within a slicing window. Once the spectrum slicing is done in a target slicing window, the amount of spectrum resources available to each BS then is fixed and depends on the corresponding slicing ratio. At a time slot within the target slicing window, vehicles send their data transmission requests to the accessible BSs. According to the received data transmission requests, the BSs then cooperatively determine the vehicle-BS association patterns for vehicles under the overlapping area and allocate proper amounts of spectrum resources to the associated vehicles.

In what follows, we summarize the existing spectrum management schemes from spectrum slicing, spectrum allocation, and joint spectrum management perspectives.

Spectrum slicing schemes: The use of spectrum slicing involves the assignment of spectrum resources to each slice in accordance with the expected requirements. Aiming at constructing multiple logically isolated slices on different BSs to support the diversified services with different QoS requirements, the spectrum slicing approaches have emerged for vehicular networks. For example, to support the V2X services that involve vehicles exchanging data with other vehicles, the infrastructure, and any communicating entity for improved transport safety, fluidity, and comfort on the road, 5G network slicing has been designed in [70, 71]. [72] and [73] have investigated the radio access network (RAN) slicing problem for V2X services. In which, the RAN slicing scheme proposed by [72] allocates radio resources to different slices to maximize the resource utilization while guaranteeing the traffic requirements of each RAN slice, and the one proposed by [73] is to dynamically allocate radio spectrum and computing resource and

distribute computation workloads for the slices to minimize the long-term system cost. In [74], the MEC and network slicing technologies are used to optimize the allocation of the network resources and guarantee the diverse requirements of vehicular applications, and a model-free approach based on DRL has been proposed to solve the resource allocation problem in MEC-enabled vehicular network based on network slicing.

Spectrum allocation schemes: Differ from slicing spectrum among BSs, spectrum allocation is mainly among vehicles, namely, each BS allocates its available spectrum resources among the vehicles associated with it. According to the types of spectrum resources allocated to the vehicular users, we summarize the existing spectrum allocation schemes from the following two respects.

Designated licensed spectrum resources: As mentioned in Subsection 1.1.4, 75 MHz radio spectrum resources in the 5.9 GHz band were allocated to support the DSRC-based vehicular communications. For a DSRC-based vehicular network, since the total number of available channels is fixed and carrier-sense multiple access with collision avoidance is applied, existing works related to spectrum management usually focused on analyzing the performance and reliability of the network [75]. Existing theoretical analysis and simulation results demonstrate that the 75 MHz DSRC band is insufficient to provide reliable safety message transmissions. Moreover, due to the relatively low priority, non-safety related users of the DSRC band have to be severely restricted during peak hours of traffic to guarantee the reliable transmission of safety messages [76]. More importantly, the spectrum scarcity problem is becoming severer due to the growing number of vehicles as well as the emerging vehicular applications. A potential method to alleviate the spectrum scarcity problem is to enable other access technologies on vehicles to allow them to communicate over other bands.

Other licensed/unlicensed spectrum resources: As mentioned in Subsection 1.1.4, in addition to DSRC, other wireless communication technologies have also been considered to support vehicular networks. That is, other licensed/unlicensed spectrum resources can be also occupied by vehicle users in conditions permitting (such as enabling the corresponding on-board access equipment) [77]. For scenarios where V2X communications are over licensed or unlicensed band beyond DSRC, vehicle users usually share the spectrum with other types of users or the primary users, how to design efficient spectrum allocation mechanisms to simultaneously guarantee the service quality of different users is critical.

Cognitive radio (CR), an effective technique to improve the efficiency of spectrum utilization by allowing cognitive users to share the wireless channel with primary users in an opportunistic manner, has been applied to allocate licensed/unlicensed spectrum resources beyond DSRC band among vehicular users [78, 79]. Such as, a novel CR assisted vehicular network framework has been proposed in [78], where vehicle users (i.e., secondary users) are empowered to make

opportunistic usage of the licensed spectrum on the highways. To improve fairness in spectrum allocation among vehicle users, the authors have proposed a light-weight cooperative three-state spectrum sensing model under this framework. And to support vehicular communication over unlicensed spectrum, a mixed-integer nonlinear programming (MINLP) problem has been formulated in [79] to address the coexistence problem between a cognitive vehicular network and an IEEE 802.22 network via resource allocation.

Another technique that has been considered to address the spectrum scarcity faced by vehicular networks is the cellular network and D2D communication. In a D2D underlaying cellular infrastructure, two physically close user devices can directly communicate with each other by sharing the same resources used by the regular cellular user, with the benefits of proximity gain, reuse gain, and hop gain. Moreover, direct local packet dissemination via cellular network can also substantially reduce latency and power consumption, thus it is suitable for delay-sensitive V2V communications. Some existing spectrum resource management schemes have been proposed for cellular-based vehicular networks [30, 80, 81]. For example, the authors of [80] have performed the spectrum sharing and power allocation for the D2D-assisted vehicular networks based only on slowly varying large-scale fading information of wireless channels.

Joint spectrum management schemes: In addition to separately investigating spectrum slicing or spectrum allocation issues, jointly considering these two issues would significantly enhance the spectrum utilization for vehicular networks and has been studied by some existing works [54, 59, 82]. For example, to support the increasing communication data traffic with guaranteed QoS, we have jointly studied the spectrum slicing and spectrum allocating based on optimization method in [59] and based on DRL method in [54]. In [82], the authors have developed an online network slicing scheduling strategy for joint resource block allocation and power control. The developed strategy can maximize the long-term time-averaged total system capacity while guaranteeing strict ultra-reliable and low-latency requirements of vehicular communication links.

2.1.2 Computing Resource Management

With the rapid increase of the vehicle density on the road, the gap between the computation service requirements and the computing capacity of vehicles becomes a serious problem. To address that, how to optimally manage the available computing resources have been studied by some existing works. Different from spectrum resources, managing the overall computing resources among vehicles is always achieved via adjusting the computation offloading association patterns between the vehicles and the resource providers, as the amount of computing resources is fixed and always mounted at different edge servers and autonomous vehicles. With

any known spectrum management strategies, the computing resource management problem is identified with a computation offloading decision making problem. In the following, we will summarize the existing computation offloading schemes based on their application scenarios.

Computation offloading among vehicles: To support automated driving and a vast variety of on-board entertainment services, AVs are equipped with substantial computing and storage resources. It is forecast that each AV will have the computing power of 10^6 dhrystone million instructions executed per second (DMIPS) in the near future [83], which is tens of times that of the current laptops. However, task overload or underload happens frequently on each individual AV due to the fluctuated demand for computing/storage resources from different AV applications, and there is always no on-board resources in some traditional manually driving vehicles, therefore resulting in low overall resource utilization and affecting the QoS satisfaction of vehicular applications.

To explore more computing power and support the emerging applications and services, some computation offloading schemes have been proposed recently to share idle computing power among neighboring vehicles, where the driving systems or passengers of some vehicles generate computing tasks while some other surrounding vehicles provide computing services. For example, an SDN based collaborative task computing scheme has been proposed in [46] to support automated driving with distributed computing task allocation, in which, a centered AV is the service requester that offloads the computing tasks, and its neighboring AVs are the service providers and dynamically form the automated driving group to execute task collaboratively based on an SDN-based market mechanism. In [84], the authors have designed a distributed task offloading algorithm based on the multi-armed bandit theory to minimize the average offloading delay, such that the service requester vehicle is able to learn the delay performance when offloading its tasks. To motivate vehicles to share their idle computing resources while simultaneously evaluating the service availability of vehicles in terms of vehicle mobility and on-board computing resources in heterogeneous vehicular networks, the authors have proposed a task offloading scheme in [85]. In which, vehicles are incentivized to share their idle computing resources by dynamic pricing with comprehensively considering the mobility of vehicles, the task priority, and the service availability of vehicles.

Computation offloading in edge server based vehicular networks: To address the challenges faced by vehicles to satisfy the increase of the demanded computation capabilities, updating hardware to physically increase the amount of on-board computing resources as expected for vehicles is the easiest way, but it introduces high monetary cost. Accessing the remote cloud computing server is another option, but it suffers from long latency and unstable connections in vehicular environments. Therefore, as discussed before, edge computing (or fog

computing[1]), enabling computing and caching capabilities at the edge of the core network to close to end devices, has been considered as a promising technology to provide low latency and high-reliability computing support for vehicle users.

To efficiently perform edge computing in highly dynamic vehicular environments, some computation offloading schemes have been proposed to perform optimal task offloading in edge-based vehicular networks [86–88]. For example, to select edge server (placed at the road side unit (RSU)) for offloading vehicle's task to and determine optimal offloading fraction, a system utility maximization problem has been formulated under the permissible latency constraint to joint consider load balancing and task offloading in [86]. In [87], the authors have investigated efficient task offloading schemes for two scenarios, independent mobile edge computing server (placed at the AP) scenario and cooperative mobile edge servers scenario, to minimize the task processing cost while satisfying the task completed maximum latency. For the independent server scenario, the offloading decisions, offloading time, and computation resource allocation in MEC servers have been jointly optimized, while for the cooperative scenario, the computation offloading among MEC servers has been also considered.

2.1.3 Multi-Resource Management

Although outstanding works mentioned in the above two subsections have been dedicated to studying spectrum allocation and computation offloading in vehicular networks, these important aspects are generally considered separately. However, similar to most wireless networks, it is necessary to jointly manage communication, computing, and storage resources to improve the performance of vehicular networks due to the following reasons,

1. Spectrum allocation, computation offloading, and content caching are all parts of the server-assisted vehicular network system, and they may all contribute to guaranteeing the heterogeneous QoS requirements of different vehicular applications, which is hardly achieved by the optimization of one single type of resource in the whole system;

2. In edge server based vehicular networks, if multiple vehicles choose to offload their computing tasks to the edge servers via RSUs or small BSs of the cellular network

[1]Generally, the key difference between fog computing and edge computing lies in where the location of intelligence and computing power is placed, where the computing power is placed at the local area network (e.g., processors connected to the local area network or installed into the local area network hardware itself) in fog environment while placed at a device that is physically "close" to the end users (e.g., the devices to which the end users are attached or a gateway device) for edge computing. However, fog computing and edge computing considered in most of the existing works that surveyed in this thesis are not specified the location of the fog/edge server, edge computing and edge computing server will be used throughout the thesis.

simultaneously, severe interference can be caused from the communication perspective, which decreases the data transmission rate significantly. From the computing perspective, the edge server could be overloaded. In such a case, it is not beneficial for all the vehicles to offload their tasks to the edge server. Instead, parts of the vehicle users should be selected to offload their tasks, while others should process their tasks either by themselves or cooperatively by the neighboring vehicles;

3. Different amounts of spectrum and computing resources should be allocated to different vehicles to fulfill the heterogeneous QoS requirements of their time-varying applications and services;

4. Due to the limited storage resources of each edge server, different caching strategies should be applied upon different contents, in order to maximize the revenue obtained by content caching.

Therefore, it is important to design multi-resource management schemes for jointly allocating communication, computing, and storage resources, which has the potential to significantly improve the resource utilization of each edge server, and therefore maximizing the commercial return of the placement of edge servers in vehicular networks.

Some works, focusing on designing resource management schemes for joint addressing computation offloading, resource allocation, and content caching in edge-based wireless networks [62, 67, 89–95], have emerged in the past years. For instances, in order to jointly tackle computation offloading and content caching issues in wireless cellular networks with mobile edge computing, in [89], the authors have formulated the computation offloading decision, resource allocation, and content caching strategy as an optimization problem with maximizing the total revenue of the network, where the net revenue is defined with considering the net revenue of system for assigning radio resources to, allocating computation resources to, and caching the internet content first requested by each user's equipment. And for vehicular networks, the joint caching and computing design problem has been investigated in [90]. To tackle the challenges in managing multiple resources caused by vehicle mobility and the hard service deadline constraint, a DRL framework has been developed. And in [95], the authors have applied AI to inspire computing, caching, and communication resources to the proximity of smart vehicles in the vehicular network. In which, an MINLP problem is formulated to minimize total network delay and solved by an AI algorithm.

2.2 Multi-Resource Management Approaches

As we mentioned in Subsection 2.1.3, it is important to design multi-resource management schemes in vehicular networks. However, there are only a few existing works dedicated to investigating this issue for vehicular networks due to the challenges caused by high vehicle mobility and vehicles' time-varying resource demand. Therefore, instead of focusing on existing works dedicated to vehicular networks, we summarize the approaches that have been considered in multi-resource management schemes proposed either for vehicular networks or other general wireless networks, and review some existing works as examples for each type of approaches in this section.

2.2.1 Optimization Approaches

Similar to managing a single type of resources, such as communication or computing resources, designing a multi-resource management scheme for a wireless network can be regarded as a decision-making process, and the key is how to simultaneously allocate the overall communication, computing, and storage resources among end users to improve the network efficiency while satisfying the resource demanding from end users. To achieve that, optimization theory and techniques have been applied in some existing multi-resource management schemes [89, 92, 96, 97].

In [89], an integrated problem of computation offloading, resource allocation, and content caching was formulated to find out the maximum value of the total revenue of mobile edge computing system operations. In which, impacts of the limited available spectrum, computing, and storage resources on the net revenue achieved by each user are considered in the constraints. In [97], the authors have established the joint problem of caching, computing, and bandwidth resource allocation, and the objective is to minimize the combined cost of network usage and energy consumption. And in [92], an integrated model jointly considering computation offloading, content caching, and communication resource allocation has been investigated. Different from the above two works, the objective of this work is to minimize the total latency consumption of the computing tasks by considering the user's QoS requirements rather than the net revenue.

Via formulating the multi-resource management problem as an optimization problem with some necessary constraints, we can always get the optimal or sub-optimal decisions for computation offloading, spectrum allocation, and content caching to maximize or minimize the corresponding objective functions. However, the formulated optimization problems would

become extremely complex, especially for scenarios with heterogeneous application requirements, diversified resource demanding, and highly dynamic network topology (such as vehicular networks), and therefore resulting in a long time cost for solving such optimization problems via the purely traditional optimization methods.

2.2.2 Learning-based Approaches

With the ability to learn to approximate various functional relations in a complex and dynamic network environment, AI and machine learning can be exploited to accomplish the objective of network resource management automatically and efficiently. The Q-learning is the most widely used model-free reinforcement learning (RL) algorithm for caching problems, computation offloading strategies [91], and resource allocation policies. Since the computational complexity and storage cost increase exponentially with the number of states and actions, the biggest challenge for Q-learning is to handle applications with extremely large size of states and actions. As deep learning becomes a hot research topic recently, exploiting the deep neural network (DNN) to estimate the value functions in RL can provide more accurate regression or approximation. Enhancing traditional RL with DNN creates a promising approach, named DRL, which is good at handling complicated control applications, such as game play or even automated driving. Due to the advantages of machine learning methods, some works make efforts to apply learning-based approaches for jointly managing the spectrum, computing, and storage resources in general wireless networks and vehicular networks [67, 84, 90, 93–95, 98].

In [94], a DRL based joint mode selection and resource management approach has been proposed for fog RANs. The core idea of this approach is that the network controller (installed at edge server) makes intelligent decisions on user equipment communication modes and processors' on-off states. In [67], joint optimization solution has proposed to simultaneously tackle the issues of content caching strategy, computation offloading policy, and radio resource allocation in the fog-enabled wireless network with the objective of minimizing the average end-to-end delay for all service requests. Considering the unknown environment's dynamic in wireless networks, the model-free RL is used to learn the best policy through a large number of training steps for the current state, where the RL agent is implemented by the edge server. Also, an actor-critic DRL algorithm is proposed to learn the optimal stochastic policy for content caching, computation offloading, and radio resources allocation. Learning-based approaches have been also applied to manage multiple resources for vehicular networks. For example, a joint optimal caching and computing allocation problem has been formulated in [90] to minimize the system cost under the constraints of limited and dynamic storage capacities and computation resources at the vehicles and the RSUs, as well as the constraints of the vehicle's mobility and the hard deadline delay. In which, to configure the parameters of caching placement and

computing resource allocation as well as to determine the sets of possible connecting RSUs and neighboring vehicles, a deep Q-learning based algorithm was developed with a multi-timescale framework. Also, to reduce the complexity caused by the large action space, a mobility-aware reward estimation was proposed for the large timescale model. Moreover, in [95], AI has been applied to study the computing, caching, and communication resource management in the vehicular network. In which, an MINLP problem is formulated to minimize total network delay, and an efficient imitation learning based branch-and-bound algorithm has been proposed to rapidly solve the formulated problem.

As the works reviewed above indicated, compared with optimization approaches, learning-based approaches can be used in wireless networks with more complicated scenarios. However, some hardware requirements such as devices with strong computing and caching power have to be enabled in such scenarios to support the learning-based approaches. Therefore, most learning-based approaches have been used in edge server or cloud server based network scenarios.

2.3 Summary

In this chapter, related works are surveyed, including spectrum resource allocation, computing resource management, and multi-resource management in vehicular networks. Also, approaches applied for multi-resource management schemes that are either for vehicular networks or other general wireless networks are summarized. Even though plenty of schemes have been proposed for managing resources in wireless networks, none of them studied how to address the challenges in communication and computing faced by vehicular networks and manage the three types of resources to support different applications with guaranteed QoS requirements. Targeted at addressing these challenges faced by vehicular networks, in the next chapter, we will introduce the proposed MEC-assisted vehicular network architecture.

Chapter 3

An MEC-Assisted Vehicular Network Architecture

In this chapter, an MVNET architecture is proposed to support vehicular applications. As the applications related to AVs are usually with stringent delay and intensive computing requirements, which can be regarded as typical emerging vehicular applications. Thus, in this chapter, designing the MVNET architecture is mainly motivated from the AVs perspectives yet can be easily extended to general vehicular networks. Improving the navigation safety by enabling HD-map-assisted cooperative driving among AVs is of great importance to the market perspective of AVs as we mentioned in Chapter 1, however, faces technical challenges due to increased communication traffic volume for data dissemination and increased number of computing/caching tasks on AVs. To address these challenges, an MEC-assisted ADVNET architecture is proposed in this chapter. Specifically, an MEC-assisted ADVNET architecture that incorporates both SDN and NFV technologies is proposed in Section 3.1. Then, a joint multi-resource management scheme is presented in Section 3.2, and some future research issues are discussed in Section 3.3. Finally, we draw concluding remarks in Section 3.4. The complete research results of this chapter can be seen in [13].

3.1 An MEC-Assisted ADVNET Architecture

In the following, we first present the problem statement to consider MEC, SDN, and NFV in ADVNETs, and then propose an MEC-assisted ADVNET architecture.

3.1.1 Problem Statement

Existing studies have indicated that enabling cooperative driving among AVs and providing real-time HD maps to AVs are important to improve AV safe navigation [99]. In cooperative driving, such as platooning, neighboring AVs on one lane move at a steady speed and keep a small steady inter-vehicle space [100]. A leader vehicle is chosen to lead all the other AVs (referred to as member vehicles) within a platoon to maintain the string stability [40]. To do so, some delay-sensitive information (e.g., speed and acceleration, braking, joining/leaving [42]) needs to be shared among cooperative driving AVs. Although providing static HD maps to show the road environment to AVs can timely compensate the inaccurate sensing information to improve AV safe navigation, the large amount of traffic volume is generated due to data exchanges among AVs, which results in increasing demand on bandwidth resources.

Moreover, new computing-intensive vehicular applications have sprung up with increasing demands for multiple types of resources. For example, large amounts of computing, storage, and bandwidth resources are demanded by virtual reality (VR). However, the increasing amount of computing/storage resources at each AV can be cost-ineffective, and uploading tasks with high computing/caching requirements to the cloud-computing server increases task response delay due to extra data transmission, and also results in high traffic load onto the core network. Therefore, we leverage the MEC technology to form an MEC-assisted ADVNET architecture. By enabling computing and caching capabilities at the edge of the core network, AVs' tasks with high computing/caching requirements can be offloaded to MEC servers when a short response delay is demanded by delay-sensitive applications. At the same time, AVs are allowed to access the computing/storage resources at MEC servers via different wireless access technologies to accommodate an increasing communication demand. To further improve the resource management efficiency, we integrate SDN control into the MEC-assisted ADVNET architecture, to enable programmable network configuration for flexible resource orchestration.

In addition to installing computing/storage resources and enabling multiple access technologies via MEC, another promising way to address the increasing demand for multiple types of resources is to improve resource management efficiency. Thus, we integrate SDN, a key technology to enable programmable network configuration, into the MEC-assisted ADVNETs for flexible resource orchestration. By activating SDN control functions on the edge servers, a hierarchical SDN control is deployed to realize radio resource pooling and resource slicing [101] among the wireless access infrastructures to improve the overall resource utilization and achieve differentiated QoS guarantees. Note that MEC runs on top of the virtualisation infrastructure [102] to support application programmability with reduced provisioning cost. With the decoupling of applications from its underlying hardware in NFV, each MEC server can host functions based on flexible application/service demands.

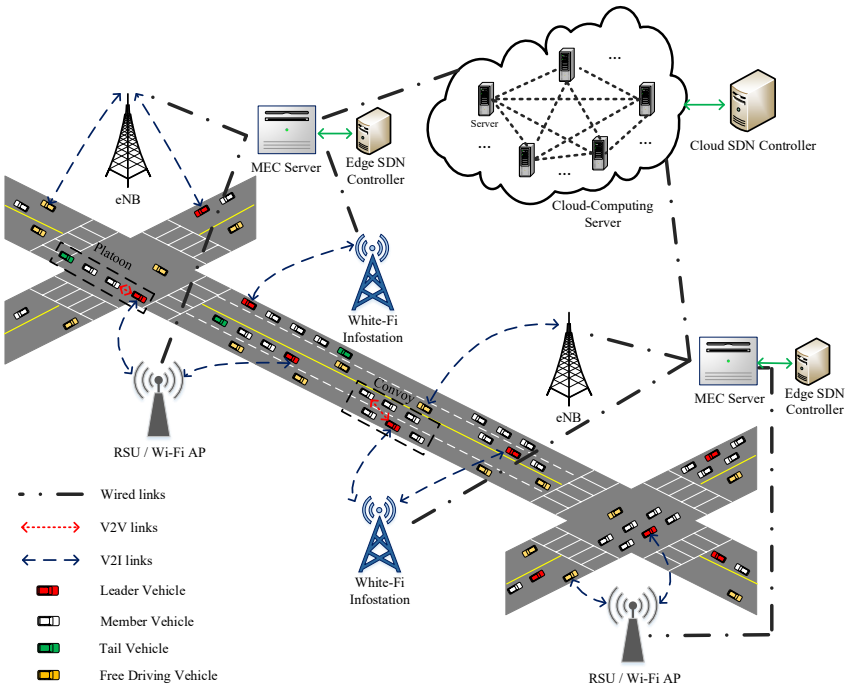


Figure 3.1: An MEC-assisted ADVNET architecture.

3.1.2 Proposed MEC-Assisted ADVNET Architecture

An MEC-assisted architecture: The proposed MEC-assisted ADVNET architecture is illustrated in Figure 3.1, which is a two-tier server structure [103, 104], i.e., a cloud-computing server in the first tier and some MEC servers in the second tier. AVs are enabled to access the internet via different BSs, such as Wi-Fi APs, RSUs, White-Fi infostations, and eNBs. Based on the resource availability (including computing, storage, and bandwidth resources) and application QoS, computing tasks can be processed on AVs, or be uploaded to and processed by MEC servers or the cloud-computing server. For example, delay-sensitive applications, such as platooning management, dynamic HD map management, and other safety-related applications, are prioritized to be processed on MEC servers for less processing latency.

To improve both resource utilization and network scalability, the MEC servers are placed at the edge of the core network to close to and maintain very few communication hops with vehicles. Thus, an MEC server can control the computing task offloading for a large number of vehicles under the coverages of several BSs, and the enhanced service area of each MEC server can better support the fluctuate service demand caused by high vehicle mobility, where

the service area of an MEC server is defined as the coverage areas of all BSs connected to the server. Via integrating the NFV concept into each MEC server, computing and storage resources on each MEC server are virtualized to host functions for different applications and services. Thus, different MEC applications, represented by virtual network functions (VNFs), can share the same set of virtualized resources that are under the NFV infrastructure and can be jointly orchestrated for different multi-access edge hosts for heterogeneous QoS guarantee. Moreover, an SDN control module is also running on the NFV-enabled MEC server. With the assistance of the edge SDN controller's partial-global information, the local computing/storage resources, including computing/storage resources placed in AVs and in the MEC server, can be efficiently allocated among different AV tasks with heterogeneous QoS guarantee. Bandwidth resources on different access networks are orchestrated through resource pooling and resource slicing to improve the overall radio resource utilization.

On the cloud server placed at the core network, computing and storage resources are also virtualized, and the processing of integrating the NFV concept into the cloud computing server is similar to that of MEC servers. SDN control functions are programmed in cloud virtual machines (VMs) to decouple the control plane from the data plane, and to manage global traffic steering among the cloud servers, upon which the resource availability and usage can be significantly enhanced. On the other hand, each MEC server forwards its state information, including the amount of idle computing/storage resources and QoS demands from different AV applications, to the cloud SDN controller after pre-analysis and pre-processing (e.g, quantization), and the controller then make decisions for task migration among MEC servers based on the received information.

A Logically-Layered Structure: To better illustrate the internal information exchange among different network components (functionalities), the proposed network architecture can be explained by using a logically-layered presentation from both the edge and cloud perspectives. Since a logically-layered structure for an MEC server shares some common components with a cloud-computing server, we describe both structures separately and emphasize their differences. Each logically-layered structure is composed of an infrastructure layer, a virtualization layer, an application layer, and a separate control functionality, as shown in Figure 3.2 for an MEC server.

The infrastructure layer represents the underlying hardware, including computing hardware and memory hardware placed at AVs and at cloud-computing/MEC servers, and network hardware such as physical remote radio heads (RRHs) in BSs and baseband units (BBUs) deployed at MEC servers. Even though the computing hardware, memory hardware, and network hardware can be managed by the cloud or edge SDN controllers, how to improve their utilization to support the ever-increasing AV applications is challenging due to the regional distribution of hardware resources and the changing density of AVs. To address that, the resource virtualization technique is utilized in the proposed architecture to achieve resource programmability and

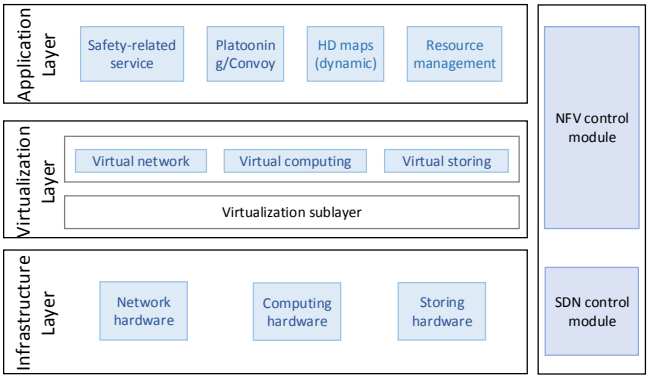


Figure 3.2: Logically-layered structure for an MEC server.

enhance global resource utilization.

Underlying hardware resources are virtualized at the virtualization layer to make computing or caching environments independent of physical components. By doing that, virtual resources can be sliced and reallocated to different applications based on their QoS requirements by the cloud/edge SDN controllers, and each application or service function is decoupled from the physical components and run as software instances on the virtual resources. Therefore, one or more application services can be supported by one MEC server, and an application can be flexibly added in or removed from the cloud-computing/MEC server without impacts on other applications. The decoupled applications or service functions are represented at the application layer. Considering heterogeneous QoS requirements and available resources in MEC servers, delay-sensitive AV applications are prioritized for registration in the MEC servers, while delay-tolerant computing tasks (e.g., hot information caching, entertainment services, and static HD map management) are registered in the cloud-computing server.

The control functionality implemented in cloud/edge SDN controllers is composed of two modules, i.e., the NFV control module and the SDN control module. For both the cloud and edge controllers, the NFV control module is responsible for resource virtualization, service function virtualization, and function placement. For example, to better utilize the computing and storage resources of the ADVNET, the NFV control module in the cloud SDN controller globally orchestrates the virtual computing/storage resources at the cloud-computing servers and the idle virtual computing/storage resources at each MEC server. In an edge SDN controller, the NFV control module locally orchestrates the virtual computing/storage resources at the MEC server and at each AV. Different from the NFV control module, the SDN control module in each controller is responsible for centralized routing and forwarding management through abstracting the control functions from the data plane. The interaction between the control plane and the data

plane is realized through the data-controller interface enabled by the OpenFlow protocol. Data flows going through the core network are under the control of the SDN control module in the cloud SDN controller, and data flows among AVs and BSs are controlled by the SDN control module in the edge SDN controller. Combining the control functions provided by the SDN and NFV control modules, the cloud/edge SDN controller allocates virtual resources to each virtual service function, and abstracts bandwidth resources and reallocates them to each connected BS, known as bandwidth slicing [105].

3.2 SDN-Enabled Resource Management

Based on the proposed MEC-assisted ADVNET architecture, the increased amount of data traffic and computing/caching tasks can be supported with guaranteed QoS. In this section, we investigate how to optimize the management of computing/storage resources among MEC servers and the slicing of bandwidth resources among BSs.

For MEC servers with pre-allocated computing and storage resources, inter-MEC resource sharing is of paramount importance. Through migrating computing/caching tasks, the distribution of task requests can be balanced among MEC servers according to their resource usages, thus enhancing the computing/storage resource utilization. Processing results of the migrated tasks will be returned to the original MEC server to respond to the request. In addition, the task migration decision can also be made based on its requester's moving direction, and the task processing results can be directly delivered to the requester once the requester moves into the service area of the new MEC server, to reduce the response delay. Consider a scenario with one cloud-computing server, M MEC servers denoted by M_i, $i = 1, \ldots, M$, and N AVs distributed over the entire ADVNET. Each BS is connected to one of the M MEC servers, and Wi-Fi/DSRC, White-Fi, and cellular technologies are applied to support AV applications. In the following, resource management schemes for computing, storage, and bandwidth are investigated for the considered scenario.

3.2.1 Computing and Storage Resource Management

Let $C_i^k(t)$ and $S_i^k(t)$ denote the amounts of computing and storage resources that MEC M_i allocates to AV k at time slot t. Due to the fixed amount of computing/storage resources at each MEC server and the varying amount of computing/caching tasks generated by the regionally distributed moving AVs, each MEC only processing the computing/caching tasks of AVs within its service area can lead to overloaded or underloaded task processing. To mitigate

the imbalanced task requests, computing/caching tasks can be migrated among MEC servers to increase the computing/storage resource utilization, which, on the other hand, incurs migration cost in terms of extra bandwidth consumption and extra response delay.

In order to obtain optimal computing and storage resource allocation while balancing the tradeoff between increasing the computing/storage resource utilization and reducing the task migration cost, an optimization problem is described as shown in Figure 3.3. The objective is to maximize the network utility which is defined as the summation of utilities of each individual MEC server. The utility of an MEC server allocating computing/storage resources to AVs is defined with the consideration of computing/storage resource utilization and task migration cost, where the computing (or storage) resource utilization of MEC server M_i is defined as the ratio of the amount of occupied resources over its total amount of computing (or storage) resources. The input of the formulated problem includes: 1) the computing, storage, and bandwidth resources placed at MEC server M_i, denoted as $C_i^{\max}$, $S_i^{\max}$, and $B_i^{\max}$, respectively, where $B_i^{\max}$ is the total available bandwidth resources of the multiple radio access technologies for ADVNETs; 2) the total amount of resources required by AV k at time slot t, denoted as $D^k(t) = \{C^k(t), S^k(t), B^k(t)\}$, including the required amounts of computing resources $C^k(t)$ and storage resources $S^k(t)$ for processing its application requests, and bandwidth resources $B^k(t)$ for downlink transmissions, where we have $C^k(t) \geq 0$, $S^k(t) \geq 0$, and $B^k(t) > 0$; 3) a downlink response delay threshold, T_{th}^k, used to guarantee that AV k (either generating delay-sensitive requests or delay-tolerate requests) receives the response before it moves out of the service area of the MEC server, and a latency threshold, L_{th}^k, used to guarantee the delay requirement of AV k generating a delay-sensitive request. In the problem formulation, the following constraints are considered: 1) for AVs that either generate delay-sensitive requests or delay-tolerate requests, the total time cost, i.e., the time interval from the time that AV k's computing/caching task is received by MEC server M_i until the time instant the corresponding response packet generated by MEC server M_i is received by AV k through a BS, should be less than the downlink response delay threshold, T_{th}^k. Moreover, for AVs that generate delay-sensitive requests, the total time cost should be also less than the latency threshold, L_{th}^k; 2) computing/storage resource constraints, i.e., the total computing (or storage) resources allocated by all the M MEC servers to AV k should satisfy the computing (or storage) resources required by AV k, i.e., $C^k(t)$ ($S^k(t)$). By solving the formulated maximization problem, the optimal computing and storage resource allocation $C_i^k(t)$ and $S_i^k(t)$ can be obtained.

3.2.2 Bandwidth Management

Due to the large service area of each MEC server, bandwidth reuse is considered among the BSs connected to the same MEC server. With the consideration of different BS coverages

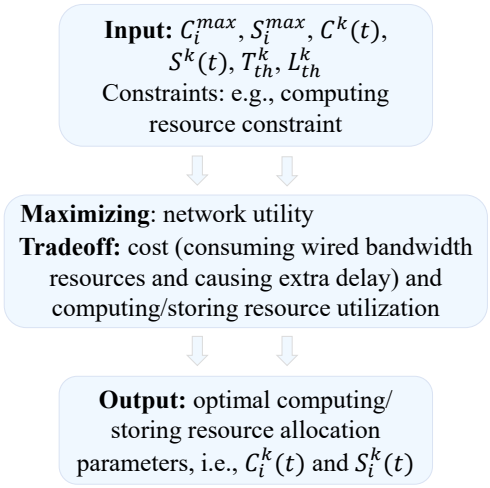


Figure 3.3: Diagramming for formulating optimization problem of computing/storage resources.

(RSUs/Wi-Fi APs, White-Fi infostations, and eNBs) from different wireless access technologies, AVs can choose to associate with BSs providing different levels of communication qualities (e.g., transmission rates). To improve bandwidth resource utilization, BSs can reuse bandwidth resources with acceptable inter-cell interference. Therefore, the goal of bandwidth slicing is to determine the set of optimal slicing ratios for different BSs, such that the aggregate network utility is maximized, and the heterogeneous application QoS requirements are satisfied.

Taking BS j and AV k under the service area of MEC server M_i as an example. Let $\gamma_j^k(t)$ denote the achievable downlink transmission rate when AV k is associated with BS j at time slot t. The utility of AV k associated with BS j is a concave function of $\gamma_j^k(t)$, e.g., a logarithmic function, and the aggregate network utility is the summation of utilities of each individual AV. Then, a network utility maximization problem is formulated, in which a two-level resource allocation is considered: 1) slicing the total bandwidth resources $R_i^{\max}$ into small resource slices, where the set of slicing ratios are denoted by $\{\beta_j | j = 1, 2, \cdots, I_i(t)\}$, where $\sum_{j=1}^{I_i(t)} \beta_j = 1$ and $I_i(t)$ is the number of BSs within the service area of MEC server M_i at time slot t; 2) partitioning the sliced resources to AVs under the coverage of and associating with each BS. Constraints under consideration include: i) $D_i^k + R_i^k \leq T_{th}^k$ and $D_i^k + R_i^k \leq L_{th}^k$, where D_i^k is the task processing delay (i.e., the time interval from the time that AV k's computing/storing task is received by MEC server M_i until the MEC server finishes processing this task) and R_i^k is the downlink transmission delay from MEC server M_i to AV k through a BS (i.e., the time interval from the time that a packet reaches the transmission queue of MEC server M_i, until the time instant the packet is received by AV k through a BS); ii) $\gamma_j^k(t) \geq \widehat{\gamma}^k$, where $\widehat{\gamma}^k$ is defined as a transmission

rate threshold for AVs that generate delay-tolerant requests; iii) $\sum_k R_j^k(t) \leq \beta_j R_i^{\max}$, where $R_j^k(t)$ is the amount of bandwidth resources that BS j allocates to AV k at time slot t.

The latency constraints for AV k reflect the coupling relationship between the two formulated problems for computing/storage resource and bandwidth resource management. Thus, these two problems have to be jointly solved, and the obtained optimal computing/storage resource allocation results and bandwidth resource allocation results maximize the network utility while collaboratively satisfying the delay requirements for delay-sensitive applications.

3.3 Open Research Issues

Combining SDN and NFV with MEC architecture to support the increasing data traffic volume while guaranteeing heterogeneous QoS requirements for different services in ADVNETs, is still in its infancy. In this section, some open research issues are discussed.

Task offloading: Since computing/storage resources on each MEC server are limited and task migration from one MEC server to another incurs an extra cost, the number of tasks allowed to be registered and processed in MEC servers should be constrained. Designing a proper task offloading criterion is necessary to maximize the computing/storage resource utilization under the constraints of task migration costs. For the proposed architecture, we determine where to register AV applications based on the application types, i.e., only safety-related applications are registered in the MEC servers. However, other criteria, such as delay requirements for each type of computing task, should also be taken into consideration to optimize the offloading decisions among MEC servers. Given the amount of resources on each MEC server, how to design appropriate criteria for task offloading among MEC servers to balance the tradeoff between QoS satisfaction and minimizing the offloading cost needs more investigation. From the perspective of AVs, each AV can process the computing task by itself, offload it to neighboring AVs when cooperative computing is enabled, or upload it to the MEC server. Due to the limited available computing/storage resources in each AV and the fluctuations of service demands and available bandwidth resources, task offloading decisions made by any AV can affect the resource utilization of the whole network and other tasks' QoS satisfaction, and therefore, how to make the task offloading decision for each AV with consideration of the distribution of service requests and available bandwidth is important. However, making task offloading decisions for AVs requires a central controller, and the interactions between the controller and AVs can increase the cost for signaling exchange and time complexity. To address this issue, a decentralized RL-based offloading decision making scheme can be designed for each individual AV.

QoS-guaranteed multiple resource management: Bandwidth and at least one type of the computing and storage resources are required by most of the tasks in the ADVNETs, such as

offloaded computing task and content caching task, and even all the three types of resources are demanded by some tasks, such as virtual reality. However, due to the coupled relation among them to guarantee QoS requirements for different tasks or balance the fairness among AVs, it is challenging to simultaneously manage the three types of resources among different tasks from both MEC servers and AVs perspectives. Moreover, task offloading among MEC servers and AVs impacts the amount of resource demanding in each MEC server and AV, which makes the multiple resource management in our proposed architecture more challenging. To address these challenges, designing a DRL-based algorithm for each MEC server and AV to jointly make task offloading decisions and manage the three types of resources is a potential solution.

MEC deployment: In our designed resource management scheme, we consider that MEC servers are placed at the edge of the core network to maintain two-hop wireless transmissions between an AV and an MEC server. Placing MEC servers close to BSs reduces the computing task response delay, but increases the computing server deployment cost. Therefore, how to place MEC servers and how much computing and storage resources should be placed on each MEC server need to be investigated for the MEC deployment problem. A simple method to deploy MEC servers is based on local service requirements to balance the placement cost with AVs' application QoS requirements. Moreover, considering service demand fluctuations due to the high AV mobility, vehicular traffic variations, and increasingly diversified applications, pre-placed MEC deployment results in inefficient resource utilization. UAV-assisted MEC deployment, i.e., mounting MEC servers in UAVs, is a promising method to this problem. Via the decentralized controlling by each edge SDN controller, moving paths for MEC-mounted UAVs and multiple resources can be scheduled to satisfy the QoS requirements even in service demand fluctuation.

Bandwidth management for uplinks: Different from downlink transmissions where BSs are at fixed locations, bandwidth allocation for uplink transmissions is more complex due to the following challenges: First, due to high AV mobility, the inter-cell interference changes dynamically and is difficult to be characterized; Second, it is inefficient for each vehicle to collect information from all neighboring vehicles to achieve local centralized control due to the highly dynamic network topology. To overcome these challenges, vehicle trajectory prediction schemes and distributed control methods can be applied when managing bandwidth for uplink transmissions.

Fairness: With the SDN control module, multiple access networks can be integrated to support AV applications. How to achieve fairness in selecting different wireless access technologies from end devices is an important research issue, where a proper fairness metric is required. Appropriate revenue models among network operators can be considered for designing a fair network selection and resource allocation scheme in terms of maximizing the revenue for each individual operator. From the end device perspective, an appropriate prioritization scheme

is necessary among different AV applications, so that fairness among AVs can be well balanced while guaranteeing QoS satisfaction.

Security and privacy: How to ensure secure communications among AVs is a key research issue. Since the accelerating or braking decisions from communication-assisted AVs are based on the collected information via V2V and V2I communications, security attacks on communication channels and sensor tampering may result in driving safety issues. Due to the MEC controllers, the privacy of applications registered in MEC servers can be improved through local communications. However, the MEC servers or cloud-computing servers can become the major targets of hackers, from which the attacks indirectly cause driving safety issues and result in serious privacy invasion. Moreover, exchanging individual vehicle information is required to support cooperative driving among AVs. How to ensure identity privacy, data privacy, and location privacy is essential to stimulate cooperative driving among AVs. To deal with these security and privacy issues, potential solutions include identity authentication for communications, access control at MEC and cloud-computing servers, and trust management from AVs and servers.

3.4 Summary

In this chapter, we have proposed a new MEC-assisted architecture considering both SDN and NFV technologies to support the increasingly intensified computing and communication requirements in ADVNETs. By applying MEC in ADVNETs, computing and storage resources are moved to the edge of the core network and AVs access the network via different wireless access technologies. Via integrating NFV into MEC, functions supporting different applications can be host on servers flexibly with reduced function provisioning cost. To achieve intelligent traffic steering and efficient multiple resource management, SDN control functionality is applied in the proposed architecture. To further improve the overall resource utilization with heterogeneous QoS guarantee, a joint multi-resource management scheme has been designed, where computing/storage resource utilization is enhanced through task migration and bandwidth resource slicing is conducted among heterogeneous BSs. Some important open research issues related to the proposed architecture are also discussed. In the next chapter, a case study will be conducted to demonstrate the effectiveness of the designed resource management framework under the proposed ADVNET architecture.

Chapter 4

Spectrum Resource Management in MEC-Assisted ADVNETs

In this chapter, the spectrum resource management issue is investigated under the MEC-assisted ADVNET architecture proposed in Chapter 3. Specifically, a dynamic spectrum management framework is proposed to improve spectrum resource utilization in the MEC-assisted ADVNET. To support the increasing communication data traffic and guarantee QoS, spectrum slicing, spectrum allocating, and transmit power controlling are jointly considered. Accordingly, three non-convex network utility maximization problems are formulated to slice spectrum among BSs, allocate spectrum among AVs associated with a BS, and control transmit powers of BSs, respectively. Via linear programming relaxation and first-order Taylor series approximation, these problems are transformed into tractable forms and then are jointly solved through an alternate concave search (ACS) algorithm. As a result, optimal spectrum slicing ratios among BSs, optimal BS-vehicle association patterns, optimal fractions of spectrum resources allocated to AVs, and optimal transmit powers of BSs are obtained. Based on our simulation, a high aggregate network utility is achieved by the proposed spectrum management scheme compared with two existing schemes. The complete research results of this chapter can be seen in [59].

4.1 Background Information

Inspired by existing works, a new architecture that combines MEC with NFV and SDN to address the challenges in computing and communication in ADVNETs is proposed in Chapter 3. Via the MEC technology, 1) AVs with limited on-board computing/storage resources can offload the

tasks requiring high computing/caching requirements to the MEC servers, such that a shorter response delay can be guaranteed by avoiding the data transfer between the core network and MEC servers; 2) multiple types of access technologies are permitted, thus moving AVs can access MEC servers via different BSs, such as Wi-Fi APs, RSUs [106], White-Fi infostations, and eNBs. Moreover, by integrating SDN and NFV concepts in each MEC server [107], global network control is enabled, and therefore, the computing/storage resources placed at MEC servers can be dynamically managed and various radio spectrum resources can be abstracted and sliced to the BSs and then be allocated to AVs by each BS.

Efficient management for computing, storage, and spectrum resources is of paramount importance for the MEC-assisted ADVNET. However, it is challenging to simultaneously manage the three types of resources while guaranteeing the QoS requirements for different AV applications, especially in a scenario with a high AV density. In this chapter, we focus on spectrum resource management which can be extended to multiple resource management as our future work discussed in the next chapter. The main contributions of this chapter are summarized as follows:

1. By considering the tradeoff between spectrum resource utilization and inter-cell interference, we develop a dynamic two-tier spectrum management framework for the MEC-assisted ADVNET, which can be easily extended to other heterogeneous networks.

2. Leveraging logarithmic and linear utility functions, we formulate three aggregate network utility maximization problems to fairly slice spectrum resources among BSs connected to the same MEC server, optimize BS-vehicle association patterns and resource allocation, and control the transmit power of BS.

3. Linear programming relaxation and first-order Taylor series approximation are used and an ACS algorithm is designed to jointly solve the three formulated optimization problems.

The remainder of this chapter is organized as follows. First, the dynamic spectrum management framework is introduced in Section 4.2, followed by the communication model. In Section 4.3, we formulate three optimization problems to slice and allocate spectrum resources among BSs and among AVs and control the transmit power of BS. Then, the three problems are transformed into tractable problems in Section 4.4 and an ACS algorithm is proposed to jointly solve them. In Section 4.5, extensive simulation results are presented to demonstrate the performance of the proposed spectrum management framework. Finally, we draw concluding remarks in Section 4.6.

4.2 System Model

In this section, we first present a dynamic spectrum management framework under the MEC-assisted ADVNET architecture with one MEC server, and then describe the communication model under the considered ADVNET.

4.2.1 Dynamic Spectrum Management Framework

Due to the high vehicle mobility and heterogeneous vehicular applications, ADVNET topology and QoS requirements change frequently, and therefore, resource allocation should be adjusted accordingly. To improve spectrum resource utilization, a dynamic spectrum management framework is developed for downlink transmission. Taking a one-way straight road with two lanes as an example in Figure 4.1, two wireless access technologies, cellular and Wi-Fi/DSRC, are available to the AVs. Wi-Fi APs/RSUs and eNBs are uniformly deployed on one side of the road, where the ith Wi-Fi AP and the jth eNB are denoted by W_i and S_j, respectively. The transmit power of each eNB, P, is fixed and high enough to guarantee a wide-area coverage, such

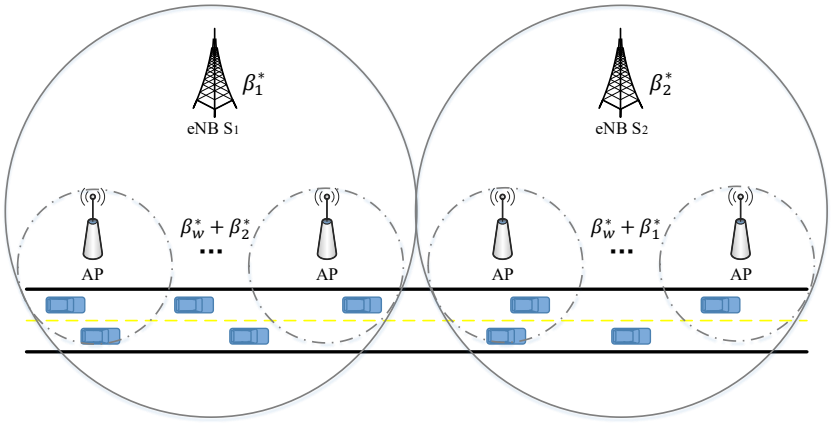


Figure 4.1: A dynamic spectrum management framework.

We divide the eNBs into two groups, denoted by $\mathcal{B}_1$ and $\mathcal{B}_2$, where eNBs in the same group share the same spectrum resources and are not neighbored to each other. ENBs S_1 and S_2 shown

in Figure 4.1 are the two target eNBs from the two different sets, where $S_1 \in \mathcal{B}_1$ is adjacent to $S_2 \in \mathcal{B}_2$. Set of Wi-Fi APs under the coverage of eNB S_j is denoted by $\mathcal{A}_j$. Denote the total available spectrum resources for vehicular applications to be $R^{\max}$. After collecting the application requests from AVs via BSs, the controller performs dynamic spectrum management for downlink transmission. The procedure can be divided into two tiers as the following.

1. *Spectrum slicing among BSs:* The controller slices the spectrum resource, $R^{\max}$, into three slices with ratio set $\{\beta_1, \beta_2, \beta_w\}$ with $\beta_1 + \beta_2 + \beta_w = 1$, and allocates them to eNBs in $\mathcal{B}_1$, eNBs in $\mathcal{B}_2$, and Wi-Fi APs, respectively.

2. *Spectrum allocating among AVs:* Once the spectrum is sliced, each BS allocates its available spectrum resources to AVs associated with it. By allocating an appropriate amount of spectrum resources to each AV, the QoS requirements of various vehicular applications can be satisfied and the sum of transmission rates over the whole ADVNET can be maximized.

Spectrum slicing among BSs, spectrum allocating among AVs, and transmit power controlling for Wi-Fi APs are updated once the traffic load of each eNB fluctuates, which is in a relatively large time scale compared to network dynamic due to vehicle mobility. The traffic load of an eNB is defined as the average arrival traffic for AVs in the coverage of the eNB.

4.2.2 Communication Model

Assume the three slices of spectrum resources are mutually orthogonal, therefore, there is no inter-slice interference. To improve the spectrum resource utilization, two levels of spectrum reusing are considered. The first level is reusing the spectrum resource $\beta_w R^{\max}$ among all the Wi-Fi APs as long as with an acceptable inter-cell interference. Moreover, we assume that the Wi-Fi APs with no overlapping coverage area with an eNB can reuse the spectrum allocated to that eNB. Thus, the interference to eNBs caused by the Wi-Fi APs can be controlled by adjusting the transmit powers of the Wi-Fi APs while the spectrum resource utilization can be further improved by allowing each Wi-Fi AP to reuse either the spectrum resource $(\beta_w + \beta_1)R^{\max}$ or $(\beta_w + \beta_2)R^{\max}$.

According to the dynamic spectrum management framework presented in Section 4.2.1, all the eNBs in $\mathcal{B}_1$ reuse the spectrum resource $\beta_1 R^{\max}$ for downlink transmission. Denote $\mathcal{M}_j/M_j$ as the set/number of AVs within the coverage of eNB S_j. Then AV k, under the coverage of eNB S_1, i.e., $k \in \mathcal{M}_1$, experiences two kinds of interference to the corresponding downlink: from

transmissions of other eNBs in $\mathcal{B}_1$ and of Wi-Fi APs in the coverage of eNBs in $\mathcal{B}_2$. Then, the spectrum efficiency at AV k ($k \in \mathcal{M}_1$) from eNB S_1 can be expressed as

$$r_1^k = \log_2(1 + \frac{P_1 G_1^k}{\sum\limits_{S_j \in \mathcal{B}_1, j \neq 1} P_j G_j^k + \sum\limits_{S_j \in \mathcal{B}_2} \sum\limits_{W_i \in \mathcal{A}_j} P_i' G_i'^k + \sigma^2}), \tag{4.1}$$

where G_j^k ($G_i'^k$) is the channel power gain between eNB S_j (Wi-Fi AP W_i) and AV k, and σ^2 is the power of the additive white Gaussian noise (AWGN). Similarly, the spectrum efficiency at AV k ($k \in \mathcal{M}_2$) from eNB S_2, r_2^k, can be obtained. Let R_j^k be the amount of spectrum allocated for AV k from eNB S_j. Then, the achievable transmission rates of AV k associated with eNBs S_1 (or S_2) can be expressed as

$$\gamma_1^k = R_1^k r_1^k \ (\text{or } \gamma_2^k = R_2^k r_2^k). \tag{4.2}$$

Denote $\mathcal{N}_i/N_i$ as the set/number of AVs within the coverage of Wi-Fi AP W_i. Let $R_{2,g}'^k$ and $R_{w,g}'^k$ be the amount of spectrum allocated to AV k from $\beta_2 R^{\max}$ and $\beta_w R^{\max}$, respectively, by Wi-Fi AP W_g under the coverage of eNB S_1 (i.e., $W_g \in \mathcal{A}_1$). Then the spectrum efficiencies at AV k from Wi-Fi AP W_g include the following two parts,

$$\begin{aligned} r_{2,g}'^k &= \log_2(1 + \frac{P_g' G_g'^k}{\sum\limits_{W_i \in \mathcal{A}_1, i \neq g} P_i' G_i'^k + \sum\limits_{S_j \in \mathcal{B}_2} P_j G_j^k + \sigma^2}) \\ r_{w,g}'^k &= \log_2(1 + \frac{P_g' G_g'^k}{\sum\limits_{W_i \in \{\mathcal{A}_1 \cup \mathcal{A}_2\}, i \neq g} P_i' G_i'^k + \sigma^2}). \end{aligned} \tag{4.3}$$

And the achievable transmission rate of a tagged AV k associated with Wi-Fi AP W_g, i.e., $k \in \cup_{W_g \in \mathcal{A}_1} \mathcal{N}_g$, can be expressed as

$$\gamma_g'^k = R_{2,g}'^k r_{2,g}'^k + R_{w,g}'^k r_{w,g}'^k. \tag{4.4}$$

Let $R_{1,h}'^k$ and $R_{w,h}'^k$ be the amount of spectrum allocated for AV k from $\beta_1 R^{\max}$ and $\beta_w R^{\max}$, respectively, by Wi-Fi AP W_h under the coverage of eNB S_2 (i.e., $W_h \in \mathcal{A}_2$), and $r_{1,h}'^k$ and $r_{w,h}'^k$ be the spectrum efficiencies at AV k from Wi-Fi AP W_h. Similarly, the achievable transmission rate of a tagged AV k associated with Wi-Fi AP W_h, i.e., $k \in \cup_{W_h \in \mathcal{A}_2} \mathcal{N}_h$, can be given by

$$\gamma_h'^k = R_{1,h}'^k r_{1,h}'^k + R_{w,h}'^k r_{w,h}'^k. \tag{4.5}$$

4.3 Resource Management Scheme

We consider two kinds of traffic for each AV: delay-sensitive traffic and delay-tolerant traffic. Examples of AV's delay-sensitive traffic include rear-end collision avoidance and platooning/convoying. The delay-tolerant traffic can be HD map information downloading and infotainment services. Denote p as the probability that an AV generates a delay-sensitive request. To accommodate the large amounts of data traffic generated by AVs while guaranteeing different QoS requirements for different applications, designing efficient resource management schemes is very important.

For downlink transmission to accommodate AVs' delay-sensitive requests, the transmission delay from eNB S_j or Wi-Fi AP W_i should be guaranteed statically. Let L_s and λ_s be the size and the arrival rate of the delay-sensitive packet. From [105], the maximum delay requirement, $D_{\max}$, can be transformed to a lower bound of the required transmission rate to guarantee that the downlink transmission delay exceeding $D_{\max}$ at most with probability ϱ, which can be expressed as

$$\gamma_{\min} = -\frac{L_s \log \varrho}{D_{\max} \log(1 - \log \varrho / (\lambda_s D_{\max}))}. \tag{4.6}$$

4.3.1 Spectrum Resource Allocation

To address complicated resource allocation, we will introduce a two-tier approach, including spectrum slicing among BSs and spectrum allocating among AVs, as following.

Spectrum slicing among BSs: Based on the dynamic spectrum management framework, the total available spectrum resources are sliced or divided according to the ratio set $\{\beta_1, \beta_2, \beta_w\}$ for different BSs. The main concern for spectrum slicing is fairness among BSs. To this end, a logarithmic utility function, which is concave and with diminishing marginal utility [105], is considered to achieve a certain level of fairness among BSs.

For AV k within the coverages of Wi-Fi APs, binary variables x_j^k and $x_i'^k$ represent the BS-vehicle association patterns, where $x_j^k = 1$ (or $x_i'^k = 1$) means AV k is associated with eNB S_j (or Wi-Fi AP W_i), $x_j^k = 0$ (or $x_i'^k = 0$) otherwise. Denote $\overline{\mathcal{M}}_j / \overline{M}_j$ as the set/number of AVs within the coverage of eNB S_j while outside of Wi-Fi APs. Then, the utility for AV k associated

with eNBs or Wi-Fi APs is

$$u_k = \begin{cases} u_1^k = \log(\gamma_1^k), & \text{if } k \in \overline{\mathcal{M}}_1 \cup \{k | x_1^k = 1\} \\ u_2^k = \log(\gamma_2^k), & \text{if } k \in \overline{\mathcal{M}}_2 \cup \{k | x_2^k = 1\} \\ u_g'^k = \log(\gamma_g'^k), & \text{if } k \in \mathcal{N}_g \cap \{k | x_g^k = 1\} \\ u_h'^k = \log(\gamma_h'^k), & \text{if } k \in \mathcal{N}_h \cap \{k | x_h^k = 1\}. \end{cases} \tag{4.7}$$

The aggregated network utility is defined as the summation of utility of each individual AV. Let $\mathbf{R} = \{R_1^k, R_2^k\}$ and $\mathbf{R}' = \{R_{2,g}'^k, R_{w,g}'^k, R_{1,h}'^k, R_{w,h}'^k\}$ be the matrices describing spectrum allocated to AVs by eNBs and by Wi-Fi APs, respectively. For given BS-vehicle association patterns with fixed transmit power of each Wi-Fi AP, the aggregated network-wide utility maximization problem can be given by

$$\mathbf{P1}: \max_{\substack{\beta_1,\beta_2,\beta_w \\ \mathbf{R},\mathbf{R}'}} \sum_{k \in \overline{\mathcal{M}}_1} u_1^k + \sum_{W_g} \sum_{k \in \mathcal{N}_g} (x_1^k u_1^k + x_g'^k u_g'^k) + \sum_{k \in \overline{\mathcal{M}}_2} u_2^k + \sum_{W_h} \sum_{k \in \mathcal{N}_h} (x_2^k u_2^k + x_h'^k u_h'^k) \tag{4.8}$$

$$\text{s.t.} \begin{cases} \beta_1, \beta_2, \beta_w \in [0,1] & \text{(4.8a)} \\ \beta_1 + \beta_2 + \beta_w = 1 & \text{(4.8b)} \\ \sum_{k \in \overline{\mathcal{M}}_1} R_1^k + \sum_{W_g} \sum_{k \in \mathcal{N}_g} x_1^k R_1^k = \beta_1 R^{\max} & \text{(4.8c)} \\ \sum_{k \in \overline{\mathcal{M}}_2} R_2^k + \sum_{W_h} \sum_{k \in \mathcal{N}_h} x_2^k R_2^k = \beta_2 R^{\max} & \text{(4.8d)} \\ \sum_{k \in \mathcal{N}_g} x_g'^k R_{l,g}'^k = \beta_l R^{\max}, \qquad l \in \{2, w\} & \text{(4.8e)} \\ \sum_{k \in \mathcal{N}_h} x_h'^k R_{l,h}'^k = \beta_l R^{\max}, \qquad l \in \{1, w\} & \text{(4.8f)} \\ R_1^k, R_2^k, R_{2,g}'^k, R_{w,g}'^k, R_{1,h}'^k, R_{w,h}'^k \geq 0. & \text{(4.8g)} \end{cases}$$

In problem (P1), the objective function is to maximize the aggregated network utility. Since β_1, β_2, and β_w are the only three slicing ratios, constraints (4.8a) and (4.8b) are considered in (P1). Constraints (4.8c)-(4.8g) indicate that spectrum resources allocated to AVs by a BS should be constrained by its available spectrum resources. According to problem (P1), each BS equally allocates the spectrum resources to AVs associated with it (will be discussed in detail in the next section). However, the downlink transmission rate required by an AV depends on its application request. For a BS with a fixed amount of available spectrum resources, equally allocating spectrum to AVs associated with it and simultaneously guaranteeing their heterogeneous QoS

requirements will reduce the number of accommodated AVs. Thus, QoS constraints on $\mathbf{R}$ and $\mathbf{R}'$ are not considered in problem (P1) and the optimal $\{\beta_1^\star, \beta_2^\star, \beta_w^\star\}$ is regarded as the only output to slice the total spectrum resources among BSs.

Spectrum allocating among AVs: To accommodate situations with high density AVs, a linear network utility function is considered in spectrum allocating among AVs associated with the same BS. For given slicing ratios β_1, β_2, and β_w, and transmit power of each Wi-Fi AP, a network throughput maximization problem can be formulated as

$$\mathbf{P2}: \max_{\substack{\mathbf{X},\mathbf{X}'\\ \mathbf{R},\mathbf{R}'}} \sum_{k\in\overline{\mathcal{M}}_1} \gamma_1^k + \sum_{W_g\in\mathcal{A}_1}\sum_{k\in\mathcal{N}_g}(x_1^k\gamma_1^k + x_g'^k\gamma_g'^k) + \sum_{k\in\overline{\mathcal{M}}_2}\gamma_2^k + \sum_{W_h\in\mathcal{A}_2}\sum_{k\in\mathcal{N}_h}(x_2^k\gamma_2^k + x_h'^k\gamma_h'^k) \tag{4.9}$$

$$\text{s.t.} \begin{cases} (4.8\text{c})-(4.8\text{g}) & & (4.9\text{a})\\ x_1^k, x_2^k, x_g'^k, x_h'^k \in \{0,1\}, & k\in\mathcal{N}_i & (4.9\text{b})\\ x_1^k + x_g'^k = 1, & k\in\cup_{W_g}\mathcal{N}_g & (4.9\text{c})\\ x_2^k + x_h'^k = 1, & k\in\cup_{W_h}\mathcal{N}_h & (4.9\text{d})\\ \gamma_l^k \geq \gamma_{min}, & l\in\{1,2\}, k\in\{\overline{\mathcal{M}}_1^s\cup\overline{\mathcal{M}}_2^s\} & (4.9\text{e})\\ x_1^k[\gamma_1^k-\gamma_{min}]\geq 0, & k\in\cup_{W_g}\mathcal{N}_g^s & (4.9\text{f})\\ x_2^k[\gamma_2^k-\gamma_{min}]\geq 0, & k\in\cup_{W_h}\mathcal{N}_h^s & (4.9\text{g})\\ x_i'^k[\gamma_i'^k-\gamma_{min}]\geq 0, & k\in\cup_{W_i\in\mathcal{A}_1\cup\mathcal{A}_2}\mathcal{N}_i^s & (4.9\text{h})\\ \gamma_l^k \geq \lambda_n L_n, & l\in\{1,2\}, k\in\{\overline{\mathcal{M}}_1^t\cup\overline{\mathcal{M}}_2^t\} & (4.9\text{i})\\ x_1^k[\gamma_1^k-\lambda_n L_n]\geq 0, & k\in\cup_{W_g}\mathcal{N}_g^t & (4.9\text{j})\\ x_2^k[\gamma_2^k-\lambda_n L_n]\geq 0, & k\in\cup_{W_h}\mathcal{N}_g^t & (4.9\text{k})\\ x_i'^k[\gamma_i'^k-\lambda_n L_n]\geq 0, & k\in\cup_{W_i\in\mathcal{A}_1\cup\mathcal{A}_2}\mathcal{N}_i^t & (4.9\text{l}) \end{cases}$$

where $\mathbf{X}=\{x_1^k, x_2^k\}$ and $\mathbf{X}'=\{x_g'^k, x_h'^k\}$ are the association pattern matrices between eNBs and AVs, and between Wi-Fi APs and AVs, respectively; L_n and λ_n are the corresponding packet size and the arrival rate for delay-tolerant service requests; $\overline{\mathcal{M}}_j^s/\overline{M}_j^s$ (or $\overline{\mathcal{M}}_j^t/\overline{M}_j^t$) are the set/number of AVs only within the coverage of eNB S_j and requesting for delay-sensitive (or delay-tolerant) services; $\mathcal{N}_i^s/N_i^s$ (or $\mathcal{N}_i^t/N_i^t$) are the set/number of AVs within the coverage of Wi-Fi AP W_i and requesting for delay-sensitive (or delay-tolerant) services.

In problem (P2), the first five constraints are same with problem (P1) and used to demonstrate the required spectrum for each AV allocated by its associated BS. Constraints (4.9b)-(4.9d) indicate that each AV is associated with either the eNB or the Wi-Fi AP closed to it. Constraints

(4.9e)-(4.9h) ensure the service rates from either eNBs and Wi-Fi APs to guarantee the delay requirements of delay-sensitive services. For AVs with delay-tolerant requests, constraints (4.9i)-(4.9l) indicate that the service rate from an eNB or a Wi-Fi AP should be not less than the periodic data traffic arrival rate at that eNB or Wi-Fi AP. Via solving problem (P2), the optimal association pattern matrices $\mathbf{X}^*$ and $\mathbf{X}^{*\prime}$, and local spectrum allocation matrices $\mathbf{R}^*$ and $\mathbf{R}^{*\prime}$ can be obtained, which maximize the network throughput with guaranteed QoS for different AV applications.

4.3.2 Transmit Power Control

In addition to spectrum slicing and allocating among BSs and among AVs, controlling the transmit power of Wi-Fi APs to adjust the inter-cell interference would further improve the spectrum utilization. Denote $\mathbf{P'} = \{P_i' | W_i \in \mathcal{A}_1 \cup \mathcal{A}_2\}$ as the transmit power matrix of Wi-Fi APs. Equations (4.1) and (4.3) indicate that the received signal-to-interference-plus-noise ratio (SINR) by AVs from either an eNB or a Wi-Fi AP change with Wi-Fi APs' transmit powers, and therefore, impacting the achievable transmission rates of the corresponding downlink. To obtain optimal transmit powers of Wi-Fi APs, the linear utility function is considered in this part similar to problem (P2). For a given slicing ratio set $\{\beta_1, \beta_2, \beta_w\}$, BS-vehicle association pattern matrices $\mathbf{X}$ and $\mathbf{X}'$, and local spectrum allocation matrices $\mathbf{R}$ and $\mathbf{R}'$, the network throughput maximization problem focusing on transmit power controlling can be formulated as

$$\mathbf{P3}: \max_{\mathbf{P'}} \sum_{k \in \overline{\mathcal{M}}_1} \gamma_1^k + \sum_{W_g \in \mathcal{A}_1} \sum_{k \in \mathcal{N}_g} (x_1^k \gamma_1^k + x_g'^k \gamma_g'^k) + \sum_{k \in \overline{\mathcal{M}}_2} \gamma_2^k + \sum_{W_h \in \mathcal{A}_2} \sum_{k \in \mathcal{N}_h} (x_2^k \gamma_2^k + x_h'^k \gamma_h'^k) \tag{4.10}$$

$$\text{s.t.} \begin{cases} (4.9e) - (4.9l), & \text{(4.10a)} \\ P_i' \in [0, P^{\max}], \qquad W_i \in \{\mathcal{A}_1 \cup \mathcal{A}_2\} & \text{(4.10b)} \end{cases}$$

where $P^{\max}$ is the maximum transmit power allowed by each Wi-Fi AP. In problem (P3), the first eight constraints in (4.10a) are same with problem (P2) and used to ensure the QoS requirements for delay-sensitive and delay-tolerant services. Constraint (4.10b) indicates that transmit power of each Wi-Fi AP is less than $P^{\max}$. Then the optimal transmit power for each Wi-Fi AP can be determined by solving problem (P3). From the above discussion, variables considered in problems (P1), (P2), and (P3) are coupled, thus the three problems should be solved jointly.

4.4 Problem Analysis and Suboptimal Solution

Due to the binary variable matrices $\mathbf{X}$ and $\mathbf{X}'$, problems (P2) and (P3) are combinatorial and difficult to solve. Thus, in this section, we first analyze each problem and then transform (P2) and (P3) to tractable forms before we jointly solving these three problems for the final optimal solutions.

4.4.1 Problem Analysis

Let $\mathcal{N}_i'$ be the set of AVs within and associated with Wi-Fi AP W_i, i.e., $\mathcal{N}_i' = \{k \in \mathcal{N}_i | x_i'^k = 1\}$ for $W_i \in \{\mathcal{A}_1 \cup \mathcal{A}_2\}$, and $|\mathcal{N}_i'| = N_i'$. Then, the objective function of (P1) can be transformed into,

$$\sum_{k \in \{\mathcal{M}_1 \setminus (\cup_{W_g} \mathcal{N}_g')\}} \log(R_1^k r_1^k) + \sum_{W_g} \sum_{k \in \mathcal{N}_g'} \log(\gamma_g'^k) + \sum_{k \in \{\mathcal{M}_2 \setminus (\cup_{W_h} \mathcal{N}_h')\}} \log(R_2^k r_2^k) + \sum_{W_h} \sum_{k \in \mathcal{N}_h'} \log(\gamma_h'^k) \tag{4.11}$$

where mathematical symbol, $\setminus$, describes the relative complement of one set with respect to another set. According to the constraints of (P1), the sets of spectrum allocating variables, $\{R_1^k\}$, $\{R_2^k\}$, $\{R_{2,g}'^k\}$, $\{R_{w,g}'^k\}$, $\{R_{1,h}'^k\}$, and $\{R_{w,h}'^k\}$, are independent with uncoupled constrains. Thus, similar to proposition 1 in [105], we can decompose problem (P1) into six subproblems and obtain the optimal fractions of spectrum allocated to AVs from the associated BSs as follows,

$$\begin{aligned}
R_1^* = R_1^{*k} &= \frac{\beta_1 R^{\max}}{M_1 - \sum_{W_g} N_g'} \\
R_2^* = R_2^{*k} &= \frac{\beta_2 R^{\max}}{M_2 - \sum_{W_h} N_h'} \\
R_{2,g}^{*\prime} = R_{2,g}^{*\prime k} &= \frac{\beta_2 R^{\max}}{N_g'} \\
R_{w,g}^{*\prime} = R_{w,g}^{*\prime k} &= \frac{\beta_w R^{\max}}{N_g'} \\
R_{1,h}^{*\prime} = R_{1,h}^{*\prime k} &= \frac{\beta_1 R^{\max}}{N_h'} \\
R_{w,h}^{*\prime} = R_{w,h}^{*\prime k} &= \frac{\beta_w R^{\max}}{N_h'}.
\end{aligned} \tag{4.12}$$

Equation (4.12) indicates that each BS equally allocates spectrum to AVs associated with it. By replacing the spectrum allocating variables with Equation (4.12), problem (P1) can be transformed into

$$\mathbf{P1}': \max_{\beta_1,\beta_2,\beta_w} \sum_{k\in\{\mathcal{M}_1\setminus(\cup_{W_g}\mathcal{N}'_g)\}} \log(\frac{\beta_1 R^{\max} r_1^k}{M_1-\sum_{W_g} N'_g}) + \sum_{W_g}\sum_{k\in\mathcal{N}'_g} \log(\frac{\beta_2 R^{\max} r'^k_{2,g} + \beta_w R^{\max} r'^k_{w,g}}{N'_g})$$
$$+ \sum_{k\in\{\mathcal{M}_2\setminus(\cup_{W_h}\mathcal{N}'_h)\}} \log(\frac{\beta_2 R^{\max} r_2^k}{M_2-\sum_{W_h} N'_h}) + \sum_{W_h}\sum_{k\in\mathcal{N}'_h} \log(\frac{\beta_1 R^{\max} r'^k_{1,h} + \beta_w R^{\max} r'^k_{w,h}}{N'_h}) \tag{4.13}$$

$$\text{s.t.}\{\ (4.8\text{a})-(4.8\text{b}) \tag{4.13a}$$

Due to the binary variable matrices $\mathbf{X}$ and $\mathbf{X}'$, using the brute force algorithm to solve problems (P2) and (P3) is with high complexity. To address this issue, we allow AVs within the overlapping area of a Wi-Fi AP and an eNB to associate with one or both the Wi-Fi AP and the eNB [108]. Thus, binary matrices $\mathbf{X}$ and $\mathbf{X}'$ are relaxed into real-valued matrices $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{X}}'$ with elements $\widetilde{x}_j^k \in [0,1]$ and $\widetilde{x}_i'^k \in [0,1]$, respectively. And then, we can transform problem (P2) into

$$\mathbf{P2}': \max_{\substack{\widetilde{\mathbf{X}},\widetilde{\mathbf{X}}'\\ \mathbf{R},\mathbf{R}'}} \sum_{k\in\overline{\mathcal{M}}_1} \gamma_1^k + \sum_{W_g\in\mathcal{A}_1}\sum_{k\in\mathcal{N}_g} (\widetilde{x}_1^k\gamma_1^k + \widetilde{x}_g'^k\gamma_g'^k) + \sum_{k\in\overline{\mathcal{M}}_2} \gamma_2^k + \sum_{W_h\in\mathcal{A}_2}\sum_{k\in\mathcal{N}_h} (\widetilde{x}_2^k\gamma_2^k + \widetilde{x}_h'^k\gamma_h'^k) \tag{4.14}$$

$$
\text{s.t.}\begin{cases}
(4.8\text{g}), (4.9\text{e}), (4.9\text{i}) & & \text{(4.14a)}\\
\sum\limits_{k\in\overline{\mathcal{M}}_1} R_1^k + \sum\limits_{W_g}\sum\limits_{k\in\mathcal{N}_g} \widetilde{x}_1^k R_1^k = \beta_1 R^{\max} & & \text{(4.14b)}\\
\sum\limits_{k\in\overline{\mathcal{M}}_2} R_2^k + \sum\limits_{W_h}\sum\limits_{k\in\mathcal{N}_h} \widetilde{x}_2^k R_2^k = \beta_2 R^{\max} & & \text{(4.14c)}\\
\sum\limits_{k\in\mathcal{N}_g} \widetilde{x}_g'^k R_{l,g}'^k = \beta_l R^{\max}, & l\in\{2,w\} & \text{(4.14d)}\\
\sum\limits_{k\in\mathcal{N}_h} \widetilde{x}_h'^k R_{l,h}'^k = \beta_l R^{\max}, & l\in\{1,w\} & \text{(4.14e)}\\
\widetilde{x}_1^k, \widetilde{x}_2^k, \widetilde{x}_g'^k, \widetilde{x}_h'^k \in [0,1], & k\in\mathcal{N}_i & \text{(4.14f)}\\
\widetilde{x}_1^k + \widetilde{x}_g'^k = 1, & k\in\cup_{W_g}\mathcal{N}_g & \text{(4.14g)}\\
\widetilde{x}_2^k + \widetilde{x}_h'^k = 1, & k\in\cup_{W_h}\mathcal{N}_h & \text{(4.14h)}\\
\widetilde{x}_1^k[\gamma_1^k - \gamma_{min}] \geq 0, & k\in\cup_{W_g}\mathcal{N}_g^s & \text{(4.14i)}\\
\widetilde{x}_2^k[\gamma_2^k - \gamma_{min}] \geq 0, & k\in\cup_{W_h}\mathcal{N}_h^s & \text{(4.14j)}\\
\widetilde{x}_i'^k[\gamma_i'^k - \gamma_{min}] \geq 0, & k\in\cup_{W_i\in\mathcal{A}_1\cup\mathcal{A}_2}\mathcal{N}_i^s & \text{(4.14k)}\\
\widetilde{x}_1^k[\gamma_1^k - \lambda_n L_n] \geq 0, & k\in\cup_{W_g}\mathcal{N}_g^t & \text{(4.14l)}\\
\widetilde{x}_2^k[\gamma_2^k - \lambda_n L_n] \geq 0, & k\in\cup_{W_h}\mathcal{N}_g^t & \text{(4.14m)}\\
\widetilde{x}_i'^k[\gamma_i'^k - \lambda_n L_n] \geq 0, & k\in\cup_{W_i\in\mathcal{A}_1\cup\mathcal{A}_2}\mathcal{N}_i^t. & \text{(4.14n)}
\end{cases}
$$

To analyze the concavity property of problems (P1′) and (P2′), three definitions about concave functions [109, 110] and two concavity-preserving operations [109] are introduced as follows,

Second-order conditions: Suppose function f is twice differentiable, i.e., it has Hessian or second derivative, $\nabla^2 f$, at each point in its domain, $\mathbf{dom} f$. Then f is concave if and only if $\mathbf{dom} f$ is a convex set and its second derivative is negative semidefinite for all $y \in \mathbf{dom} f$, i.e., $\nabla^2 f \preceq 0$.

To express biconcave set and biconcave function, we define $A \subseteq \mathbb{R}^n$ and $B \subseteq \mathbb{R}^m$ as two non-empty convex sets, and let Y be the Cartesian product of A and B, i.e., $Y \subseteq A \times B$. Define a- and b-sections of Y as $Y_a = \{b \in B : (a,b) \in Y\}$ and $Y_b = \{a \in A : (a,b) \in Y\}$.

Biconcave set: Set $Y \subseteq A \times B$ is called as a biconcave set on $A \times B$, if Y_a is convex for every $a \in A$ and Y_b is convex for every $b \in B$.

Biconcave function: Define function $f\colon Y \to \mathbb{R}$ on a biconvex set $Y \subseteq A \times B$. Then function $f\colon Y \to \mathbb{R}$ is called a biconcave function on Y, if $f_a(b) = f(a,b) : Y_a \to \mathbb{R}$ is a

concave function on Y_a for every given $a \in A$, and $f_b(a) = f(a,b) : Y_b \to \mathbb{R}$ is a concave function on Y_b for every given $b \in B$.

Biconcave optimization problem: An optimization problem with form $\max\{f(a,b) : (a,b) \in Y\}$ is called as a biconcave optimization problem, if the feasible set Y is biconvex on Y_a and Y_b, and the objective function $f(a,b)$ is biconcave on Y.

Nonnegative weighted sums: A nonnegative weighted sum of concave functions is concave.

Composition with an affine mapping: Let function $\hbar : \mathbb{R}^n \to \mathbb{R}$, $E \in \mathbb{R}^{n\times m}$, and $e \in \mathbb{R}^n$. Define function $\ell : \mathbb{R}^m \to \mathbb{R}$ by $\ell(y) = \hbar(Ey + e)$ with **dom** $\ell = \{y|Ey + e \in$ **dom** $\hbar\}$. Then function ℓ is concave if $\hbar$ is concave.

The following two propositions summarize the concavity property of problems (P1′) and (P2′), respectively, The objective function of problem (P1′) is a concave function on the three optimal variables β_1, β_2, and β_w, and problem (P1′) is a concave optimization problem.

Proof: For problem (P1′), constraint (4.13a) indicates that $\{\beta_1, \beta_2, \beta_w\}$ is a closed set, i.e., the problem domain is a convex set, and the objective function of (P1′) is the summation of AVs' logarithmic utilities, where the logarithmic function is a concave function due to the non-positive second derivative. Moreover, for an AV associated with a BS, the utility is logarithm of the achievable transmission rate, and the corresponding achievable transmission rate is an affine function of β_1, β_2, or β_w. Thus, based on the above two operations, we can conclude that the objective function of problem (P1′) is a concave function on the three optimal variables β_1, β_2, and β_w. Furthermore, constraint (4.8a) can be rewritten into inequality concave constraints, such as $\beta_1 \in [0,1]$ can be as $-\beta_1 \leq 0$ and $\beta_1 \leq 1$, and constraint (4.8b) is an equality affine function. Therefore, problem (P1′) is a concave optimization problem.

The objective function of problem (P2′) is a biconcave function on variable set $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\} \times \{\mathbf{R}, \mathbf{R}'\}$, and problem (P2′) is a biconcave optimization problem.

Proof: Constraints (4.14b)-(4.14f) of problem (P2′) indicate that $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\}$ and $\{\mathbf{R}, \mathbf{R}'\}$ are convex sets, and Cartesian product is an operation that preserves convexity of convex sets [109]. Thus, the domain of (P2′)', $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\} \times \{\mathbf{R}, \mathbf{R}'\}$, is a convex set. Moreover, as stated before, the objective function of (P2′) is the summation of AVs' achievable transmission rates from the associated BSs, where the transmission rate achieved by an AV k is an affine function on elements of $\{\mathbf{R}, \mathbf{R}'\}$ for a given association pattern and is an affine function on the association pattern variable for a given resource allocation. Considering the affine function is both concave and convex, it can prove that the objective function of problem (P2′) is a biconcave function on variable set $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\} \times \{\mathbf{R}, \mathbf{R}'\}$. Moreover, constraints (4.14g) and (4.14h) are equality affine

on $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\}$, constraints (4.14b)-(4.14e) are equality biaffine on $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\} \times \{\mathbf{R}, \mathbf{R}'\}$, constraints (4.8g) and (4.14f) are respectively inequality affine on $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\}$ and $\{\widetilde{\mathbf{R}}, \widetilde{\mathbf{R}}'\}$, and constraints (4.9e), (4.9i), and (4.14i)-(4.14n) are inequality biaffine on $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\} \times \{\mathbf{R}, \mathbf{R}'\}$. Thus, we can conclude that (P2′) is a biconcave optimization problem.

Even though the integer-value variables in problem (P3) can be relaxed to real-value ones by replacing constraint (4.10a) by (4.9e), (4.9i), and (4.14i)-(4.14n), the non-concave or non-biconcave relations between the objective function and decision variable of problem (P3) makes it difficult to solve directly. Thus, we use the first-order Taylor series approximation, and introduce two new variable matrices, $\mathbf{C} = \{C_1^k, C_2^k\}$ and $\mathbf{C}' = \{C_{2,g}'^k, C_{w,g}'^k, C_{1,h}'^k, C_{w,h}'^k\}$ with elements that are linear-fractional function of P_i', to replace the received SINR on AVs within each BS's coverage area. Then, the downlink spectrum efficiency on an AV associated with a BS can be re-expressed as a concave function of C. For example, using C_1^k to replace the SINR received on AV k associated with eNB S_1, we can rewritten equation (5.4) as

$$r_1^k = \log_2(1 + C_1^k). \tag{4.15}$$

Therefore, problem (P3) can be transformed into

$$\begin{aligned}
\mathbf{P3}': \max_{\mathbf{P}',\mathbf{C},\mathbf{C}'} & \sum_{k\in\overline{\mathcal{M}}_1} R_1^k \log_2(1 + C_1^k) + \sum_{k\in\overline{\mathcal{M}}_2} R_2^k \log_2(1 + C_2^k) + \sum_{W_g\in\mathcal{A}_1} \sum_{k\in\mathcal{N}_g} (x_1^k R_1^k \log_2(1+ \\
& C_1^k) + x_g'^k (R_{2,g}'^k \log_2(1 + C_{2,g}'^k) + R_{w,g}'^k \log_2(1 + C_{w,g}'^k))) + \sum_{W_h\in\mathcal{A}_2} \sum_{k\in\mathcal{N}_h} (x_2^k R_2^k \\
& \log_2(1 + C_2^k) + x_h'^k (R_{1,h}'^k \log_2(1 + C_{1,h}'^k) + R_{w,h}'^k \log_2(1 + C_{w,h}'^k)))
\end{aligned} \tag{4.16}$$

$$\text{s.t.} \begin{cases}
(4.9e), (4.9i), (4.14i) - (4.14n) & (4.16a)\\
P_i' \in [0, P^{\max}], \qquad W_i \in \{\mathcal{A}_1 \cup \mathcal{A}_2\} & (4.16b)\\
C_1^k \le \xi_1^k & (4.16c)\\
C_2^k \le \xi_2^k & (4.16d)\\
C_{2,g}'^k \le \xi_{2,g}'^k & (4.16e)\\
C_{w,g}'^k \le \xi_{w,g}'^k & (4.16f)\\
C_{1,h}'^k \le \xi_{1,h}'^k & (4.16g)\\
C_{w,h}'^k \le \xi_{w,h}'^k & (4.16h)
\end{cases}$$

where ξ^k (or ξ'^k) are the received SINRs on AV k from its associated eNB (or Wi-Fi AP). The six additional constraints (4.16c)-(4.16h) are biaffine on $\{\mathbf{P}'\} \times \{\mathbf{C}, \mathbf{C}'\}$ and are considered in problem (P3′) to ensure the equivalent with problems (P3).

4.4.2 Algorithms Design

To jointly solve the three problems (P1′), (P2′), and (P3′), we first design an alternate algorithm for (P3′) and then an ACS algorithm is applied to jointly solve these three problems. For simplicity, the objective functions for the three problems are denoted by $\mathcal{U}_{(P1')}$, $\mathcal{U}_{(P2')}$, and $\mathcal{U}_{(P3')}$, respectively.

The objective function of problem (P3′), $\mathcal{U}_{(P3')}$, is concave on $\{\mathbf{C}, \mathbf{C}'\}$, while constraints (4.16c)-(4.16h) are biaffine on $\{\mathbf{P}'\} \times \{\mathbf{C}, \mathbf{C}'\}$. Through maximizing $\mathcal{U}_{(P3')}$, optimal $\{\mathbf{C}, \mathbf{C}'\}$ can be obtained for given $\mathbf{P}'$ with constraints (4.16c)-(4.16h). Moreover, through maximizing 0 with constraints (4.16a)-(4.16h), the feasible set of $\mathbf{P}'$ can be obtained. Thus, we first separate problem (P3′) into two subproblems as follows

$$\begin{aligned} \mathbf{P3'.SP1}: \max_{\mathbf{C},\mathbf{C}'} \ & \mathcal{U}_{(P3')} \\ \text{s.t. } & (4.16c) - (4.16h) \end{aligned}$$

and

$$\begin{aligned} \mathbf{P3'.SP2}: \max_{\mathbf{P}'} \ & 0 \\ \text{s.t. } & (4.16a) - (4.16h). \end{aligned}$$

It is obvious that there must be a solution to subproblem (P3′.SP1). Moreover, since subproblem (P3′.SP2) is a feasibility problem and the initial value of $\mathbf{P}'$ is always the solution for (P3′.SP2). Thus, problem (P3′) converges and can be solved by iteratively solving subproblems (P3′.SP1) and (P3′.SP2).

To jointly solve (P1′), (P2′), and (P3′) and obtain the final optimal decision variables, the ACS algorithm is summarized in Algorithm 1. $\{\widetilde{\mathbf{X}}^{(t)}, \widetilde{\mathbf{X}}^{(t)\prime}\}$ and $\mathbf{P}^{(t)\prime}$ are the values of $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\}$ and $\mathbf{P}'$ at the beginning of the tth iteration, and $\mathcal{U}^{(t)}_{(P2')}$ is the maximum objective function value of problem (P2′) with optimal decision variables $\{\beta_1^{(t)}, \beta_2^{(t)}, \beta_w^{(t)}\}$, $\{\widetilde{\mathbf{R}}^{(t)}, \widetilde{\mathbf{R}}^{(t)\prime}\}$, $\{\widetilde{\mathbf{X}}^{(t)}, \widetilde{\mathbf{X}}^{(t)\prime}\}$, and $\mathbf{P}^{(t)\prime}$. To enhance the convergence speed of Algorithm 1, the output at the $(t-1)$th iteration is regarded as a feedback to the input at the tth iteration [111], such as, the tth input $\mathbf{P}^{(t)\prime}$ is defined as

$$\mathbf{P}^{(t)\prime} = \mathbf{P}^{(t-1)\prime} + \theta(\mathbf{P}^{\dagger\prime} - \mathbf{P}^{(t-1)\prime}) \tag{4.17}$$

where, θ is the feedback coefficient. Moreover, considering that a lager θ may result in missing optimal output at each iteration while a small θ reduces the convergence speed, two coefficients

θ_1 and θ_2 are considered in Algorithm 1.

According to the analysis of each problem in Subsection 4.4.1, Algorithm 1 converges since:

1. The output of problems (P1′) and (P2′), $\{\beta_1, \beta_2, \beta_w\}$, $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\}$, and $\{\widetilde{\mathbf{R}}, \widetilde{\mathbf{R}}'\}$, are closed sets;
2. Both (P1′) and (P2′) are concave/biconcave optimization problems such that the optimal solution for each problem at the end of the kth iteration is unique when the input of the algorithm is the optimal results obtained from the $(k-1)$th iteration;
3. Problem (P3′) is always solvable.

The computational complexity of Algorithm 1 is calculated as follows: In t-th iteration, the convex optimization problem (P2′) is solved at step 2 for the optimal spectrum allocation results, where the number of decision variables is $2\sum_{j=1}^{B} M_j$; Then at step 3, solving (P2′) again for the optimal vehicle-eNB and vehicle-AP association patterns with $2\sum_{j=1}^{B} M_j$ decision variables; Then at step 4, iteratively solving (P3′) for optimal $\{\mathbf{C}^{(t+1)}, \mathbf{C}^{(t+1)\prime}\}$ and $\mathbf{P}^{(t+1)\prime}$ with numbers of decision variables $2\sum_{j=1}^{B} M_j$ and $\sum_{j=1}^{B} A_j$, respectively; And solving (P1′) with 3 decision variables for the optimal slicing ratios at the end of t-th iteration. Therefore, in each iteration, the convex optimization problems are solved sequentially by using interior-point methods, and thus the time complexity upper bound of Algorithm 1 is $\mathcal{O}\left[\widehat{N}((\sum_{j=1}^{B} M_j)^{3.5} + (\sum_{j=1}^{B} A_j)^{3.5})\right]$, where $\widehat{N}$ is the maximum number of iterations, B is the number of eNBs within the considered area, and M_j/A_j is the number of AVs/Wi-Fi APs within the coverage of eNB S_j. In practical, M_j and A_j are generally less than 100 and 10, respectively.

4.5 Simulation Results

To show the effectiveness of our proposed spectrum resource management framework, extensive simulation is carried out. We compare the proposed spectrum resource management scheme with two existing resource slicing schemes, i.e., the maximization-utility (max-utility) based resource slicing scheme proposed in [105], and the maximization-SINR (max-SINR) based resource slicing scheme proposed in [108]. The BS-vehicle association patterns and spectrum slicing ratios are optimized with the objective of maximizing the aggregated network utility in the max-utility scheme while AVs choose to associate with the BS providing higher SINR and only spectrum slicing ratios are optimized in the max-SINR scheme.

We consider two eNBs (eNB $S_1 \in \mathcal{B}_1$ and eNB $S_2 \in \mathcal{B}_2$) and four Wi-Fi APs (AP 1 and AP 2 in $\mathcal{A}_1$, and AP 3 and AP 4 in $\mathcal{A}_2$) are utilized for AV applications. Transmit power is

Input: Input parameters for (P1′), (P2′), and (P3′); initial values for $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\}$ and $\mathbf{P}'$; terminating criterion κ_1; feedback coefficient updating criterion κ_2 ($\kappa_2 > \kappa_1$); feedback coefficients θ_1 and θ_2; maximum iterations $\widehat{N}$.

Output: Optimal spectrum slicing ratios, $\{\beta_1^*, \beta_2^*, \beta_w^*\}$; Optimal local spectrum allocation matrix, $\{\mathbf{R}^*, \mathbf{R}^{*\prime}\}$; Optimal BS-vehicle association patterns, $\{\widetilde{\mathbf{X}}^*, \widetilde{\mathbf{X}}^{*\prime}\}$; Optimal transmit powers for APs, $\mathbf{P}^{*\prime}$; Optimal SINR matrices $\{\mathbf{C}^*, \mathbf{C}^{*\prime}\}$.

/* Initialization phase */

for *the first iteration,* $k = 0$ **do**

set initial values for $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}'\}$ and $\mathbf{P}'$, denoted by $\{\widetilde{\mathbf{X}}^{(0)}, \widetilde{\mathbf{X}}^{(0)\prime}\}$ and $\mathbf{P}^{(0)\prime}$, respectively; set $\mathcal{U}_{(P2')}^{(0)}$ to 0.

end

/* Solving iteratively phase */

repeat

foreach $k \leq \widehat{N}$ **do**

Step1: $\{\beta_1^\dagger, \beta_2^\dagger, \beta_w^\dagger\} \leftarrow$ solving (P1′) given $\{\widetilde{\mathbf{X}}^{(t)}, \widetilde{\mathbf{X}}^{(t)\prime}\}$ and $\mathbf{P}^{(t)\prime}$;

Step2: $\{\mathbf{R}^\dagger, \mathbf{R}^{\dagger\prime}\} \leftarrow$ solving (P2′) given $\{\beta_1^\dagger, \beta_2^\dagger, \beta_w^\dagger\}$, $\{\widetilde{\mathbf{X}}^{(t)}, \widetilde{\mathbf{X}}^{(t)\prime}\}$, and $\mathbf{P}^{(t)\prime}$;

Step3: $\{\widetilde{\mathbf{X}}^\dagger, \widetilde{\mathbf{X}}^{\dagger\prime}\} \leftarrow$ solving (P2′) given $\{\beta_1^\dagger, \beta_2^\dagger, \beta_w^\dagger\}$, $\{\mathbf{R}^\dagger, \mathbf{R}^{\dagger\prime}\}$, and $\mathbf{P}^{(t)\prime}$;

Step4: $\{\mathbf{C}^{(t+1)}, \mathbf{C}^{(t+1)\prime}\}$, $\mathbf{P}^{(t+1)\prime} \leftarrow$ solving (P3′) by iteratively solving (P3′.SP1) and (P3′.SP2) given $\{\beta_1^\dagger, \beta_2^\dagger, \beta_w^\dagger\}$, $\{\mathbf{R}^\dagger, \mathbf{R}^{\dagger\prime}\}$, and $\{\widetilde{\mathbf{X}}^\dagger, \widetilde{\mathbf{X}}^{\dagger\prime}\}$;

if *No feasible solutions for (P1′), (P2′), or (P3′)* **then**

Go to initialization phase and reset the initial values for related parameters until no feasible solutions found; Stop and no optimal solutions under current network setting;

else if $\| \mathcal{U}_{(P2')}^{(t)} - \mathcal{U}_{(P2')}^{(t-1)} \| \leq \kappa_2$ **then**

$\{\beta_1^{(t+1)}, \beta_2^{(t+1)}, \beta_w^{(t+1)}\} \leftarrow$
$\{\beta_1^{(t)}, \beta_2^{(t)}, \beta_w^{(t)}\} + \theta_2 * (\{\beta_1^\dagger, \beta_2^\dagger, \beta_w^\dagger\} - \{\beta_1^{(t)}, \beta_2^{(t)}, \beta_w^{(t)}\})$;

$\{\mathbf{R}^{(t+1)}, \mathbf{R}^{(t+1)\prime}\} \leftarrow \{\mathbf{R}^{(t)}, \mathbf{R}^{(t)\prime}\} + \theta_2 * (\{\mathbf{R}^\dagger, \mathbf{R}^{\dagger\prime}\} - \{\mathbf{R}^{(t)}, \mathbf{R}^{(t)\prime}\})$;

$\{\mathbf{X}^{(t+1)}, \mathbf{X}^{(t+1)\prime}\} \leftarrow \{\mathbf{X}^{(t)}, \mathbf{X}^{(t)\prime}\} + \theta_2 * (\{\mathbf{X}^\dagger, \mathbf{X}^{\dagger\prime}\} - \{\mathbf{X}^{(t)}, \mathbf{X}^{(t)\prime}\})$.

end

else

$\{\beta_1^{(t+1)}, \beta_2^{(t+1)}, \beta_w^{(t+1)}\} \leftarrow$
$\{\beta_1^{(t)}, \beta_2^{(t)}, \beta_w^{(t)}\} + \theta_1 * (\{\beta_1^\dagger, \beta_2^\dagger, \beta_w^\dagger\} - \{\beta_1^{(t)}, \beta_2^{(t)}, \beta_w^{(t)}\})$;

$\{\mathbf{R}^{(t+1)}, \mathbf{R}^{(t+1)\prime}\} \leftarrow \{\mathbf{R}^{(t)}, \mathbf{R}^{(t)\prime}\} + \theta_1 * (\{\mathbf{R}^\dagger, \mathbf{R}^{\dagger\prime}\} - \{\mathbf{R}^{(t)}, \mathbf{R}^{(t)\prime}\})$;

$\{\mathbf{X}^{(t+1)}, \mathbf{X}^{(t+1)\prime}\} \leftarrow \{\mathbf{X}^{(t)}, \mathbf{X}^{(t)\prime}\} + \theta_1 * (\{\mathbf{X}^\dagger, \mathbf{X}^{\dagger\prime}\} - \{\mathbf{X}^{(t)}, \mathbf{X}^{(t)\prime}\})$.

end

Obtain $\mathcal{U}_{(P2')}^{(t+1)}$ at the end of kth iteration with $\{\beta_1^{(t+1)}, \beta_2^{(t+1)}, \beta_w^{(t+1)}\}$, $\{\widetilde{\mathbf{R}}^{(t+1)}, \widetilde{\mathbf{R}}^{(t+1)\prime}\}$, $\{\widetilde{\mathbf{X}}^{(t+1)}, \widetilde{\mathbf{X}}^{(t+1)\prime}\}$, and $\mathbf{P}^{(t+1)\prime}$;

end

$k \leftarrow k + 1$;

until $\| \mathcal{U}_{(P2')}^{(t)} - \mathcal{U}_{(P2')}^{(t-1)} \| \leq \kappa_1$ *or* $k \geq \widehat{N}$;

Algorithm 1: The ACS algorithm for jointly solving (P1′), (P2′), and (P3′).

fixed at 10 watts (i.e., 40 dBm) for each eNB with a maximum communication range of 600 m. Since no transmit power control for both max-utility and max-SINR schemes, transmit powers of APs are set as 1 watt with communication range of 200 m, the same as in [105]. In our simulation, the minimum inter-vehicle distance is 5 m, and the AV density over one lane, i.e., the number of AVs on one lane per meter, varies within range of $[0.04, 0.20]$ AV/m. The downlink channel gains for eNBs and Wi-Fi APs are described as $L_e(d) = -30 - 35\log_{10}(d)$ and $L_w(d) = -40 - 35\log_{10}(d)$ [105], respectively, where d is the distance between an AV and a BS. We take platooning/convoying as an example to set the delay bound for delay-sensitive applications, i.e., 10 ms [40, 112], and downloading HD map is considered as an example for delay-tolerant applications [113]. Other important parameters in our simulation are listed in Table 4.1.

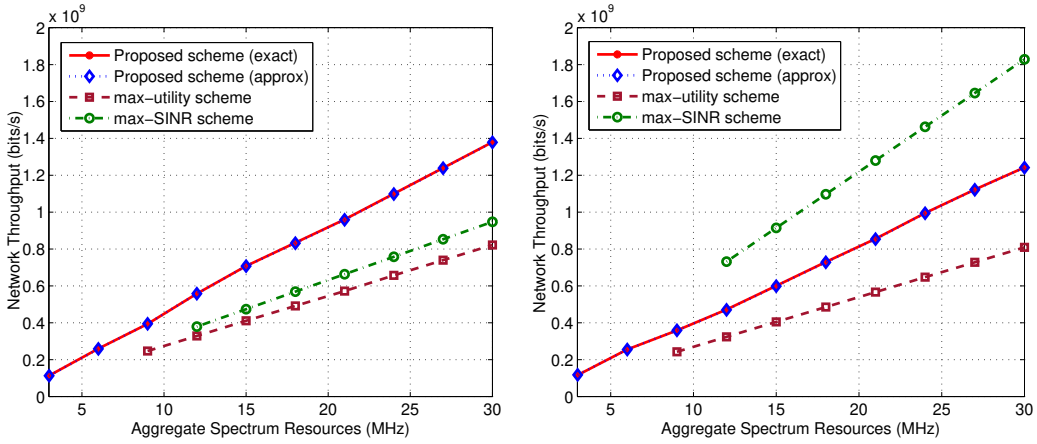
Table 4.1: Parameters values.

Parameter	**Value**
Maximum transmit power allowed by APs	2.5 watts
Background noise power	-104 dBm
HD map packet arrival rate	20 packet/s
HD map packet size	9000 bits
Safety-sensitive packet arrival rate	4 packet/s
Safety-sensitive packet size	1048 bits
Safety-sensitive packet delay bound	10 ms
Safety-sensitive request generating probability	$0.1 - 0.9$
Delay bound violation probability	10^{-3}
θ_1/θ_2	0.001/0.1
κ_1/κ_2	0.01/20

We use network throughput that is, the summation of achievable transmission rate by each individual AV from BSs, to measure performances of different spectrum resource management schemes. Considering the scarcity of spectrum resources, the different vehicular applications, and the high network dynamic, we evaluate the performance of the proposed scheme and compare with the max-utility and the max-SINR schemes under different amounts of aggregate spectrum resource (W_v), probabilities of generating a delay-sensitive request by AVs (p), and AV densities in Figure 4.2 to Figure 4.4.

Figure 4.2 demonstrates the network throughputs achieved by the three schemes with respect to different amounts of aggregate spectrum resources, W_v, where AV density is 0.05 AV/m and $p = 0.2$ and 0.8, respectively. With the increase of W_v, the transmission rate for each AV is increased due to the increase of the amount of allocated spectrum resources. From Figure

4.2, the minimum requirement for spectrum resources by the proposed scheme to support the downlink transmissions is 3 MHz while at least 9 MHz and 12 MHz spectrum is required by the max-utility scheme and the max-SINR scheme[1], respectively. Moreover, under different W_v, the network throughput achieved by the proposed scheme is on average over 70% and over 50% higher than that of the max-utility scheme for $p = 0.2$ and 0.8, respectively, and over 45% higher on average than that of the max-SINR scheme for $p = 0.2$. From Figure 4.2(a), with the increase of W_v, network throughput achieved by the proposed scheme increases more rapidly than the max-utility scheme.



(a) $p = 0.2$ (b) $p = 0.8$

Figure 4.2: Comparison of network throughput vs. aggregate spectrum resources under the same AV distribution with AV density 0.05AV/m.

Network throughputs of the three schemes under different p are evaluated in Figure 4.3. The effect of p on network throughput is mainly caused by the difference between the QoS requirements for delay-sensitive and delay-tolerant applications. According to Equation (4.6) and the parameter setting in Table 4.1, the transmission rate required by a delay-tolerant request is 180.00 kbits/s, which is higher than that for a delay-sensitive request, 140.37 kbits/s. A large p indicates a low total transmission rate required by all AVs to satisfy their applications' QoS requirements, therefore more remaining spectrum resources can be allocated to AVs with higher received SINRs in the proposed scheme. Thus, under the scenarios with the same AV density, 0.05 AV/m, network throughputs of the three schemes increase with p. For the max-SINR

[1]Note that, the range of the x-axis is set according to our proposed scheme, and no network throughput is shown for the max-utility scheme or the max-SINR scheme when the amount of resources is not enough to support the given scenario by these two schemes.

scheme, AVs associate with the BS providing higher SINR and each BS equally allocates its available spectrum resources to AVs. To guarantee the QoS requirements for AVs, the amount of spectrum resource allocated to AVs from the same BS fluctuates with the distribution of BS-vehicle SINR and p, resulting in drastic impacts on the achieved network throughput. Moreover, from Figure 4.3, the proposed scheme outperforms the max-SINR scheme when p is small and can achieve higher network throughput than the max-utility scheme for the scenario with different p.

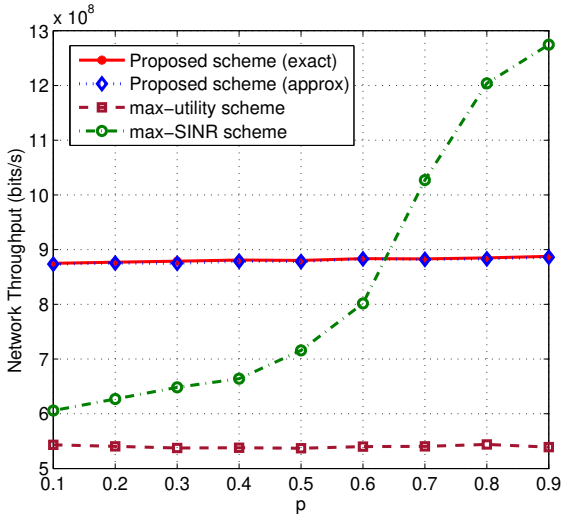
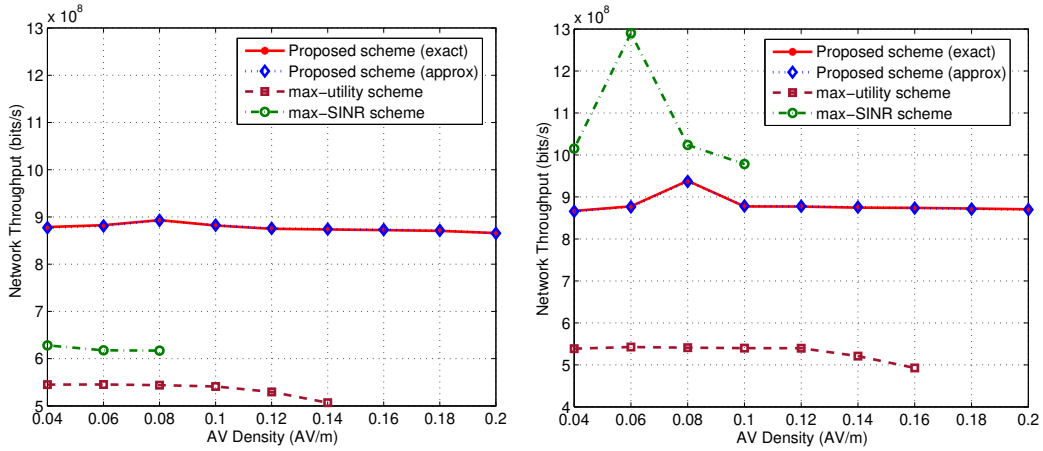


Figure 4.3: Average network throughput vs. p (AV density is 0.05AV/m).

Fig. 4.4 shows the network throughputs of the three schemes under different AV densities with $p = 0.2$ and 0.8, respectively, and 20 MHz aggregate spectrum resources. From the figure, the proposed scheme is more robust to AV density changing than the other two. For both the max-SINR and the max-utility schemes, only scenarios with small AV densities can be accommodated due to equal spectrum allocation among AVs and unbalance between the downlink data traffic and the available aggregate spectrum resources. Furthermore, the proposed scheme has over 50% increase in the achieved network throughput than the max-utility scheme with $p = 0.2$ and 0.8 and has over 40% increase than the max-SINR scheme when $p = 0.2$.

Fig. 4.4 also indicates the effect of AV density on the achieved network throughputs by the three schemes. In general, the network throughputs achieved by the three schemes overall decrease with AV density. To increase the network throughput, the proposed scheme and the max-SINR scheme prefer slicing high spectrum ratio to the BSs providing higher SINRs to its associated AVs once enough spectrum is allocated to each AV to guarantee the QoS requirements

for their applications. When the AV density is relatively lower (e.g., 0.04 AV/m), 20 MHz spectrum resource is more than enough to satisfy each request's QoS requirement and the average probability for AVs with high SINR increases with the AV density, therefore resulting in increased network throughput. However, the amount of spectrum resources needed to satisfy AV application's QoS requirements increases with the AV density for the three schemes, thus less spectrum resources can be used to increase network throughput, resulting in decreasing in network throughput.



(a) $p = 0.2$ (b) $p = 0.8$

Figure 4.4: Average network throughput vs. AV density.

Figure 4.2 to Figure 4.4 show that the proposed scheme outperforms the two comparisons in terms of network throughput. In addition to replacing the equality allocation with on-demand spectrum allocating among AVs, the performance improvement is also due to the transmit power controlling in the proposed scheme. Taking scenarios with four different AV densities, i.e., 0.05 AV/m, 0.10 AV/m, 0.15 AV/m, and 0.20 AV/m, as examples, the optimal transmit powers obtained by the proposed scheme are shown in Table 4.2. To avoid the impact of the initial APs' transmit powers on the network throughput, APs' transmit powers are fixed on 2.5 watts with a communication range of 260 m for both comparisons. With 0.05 AV/m AV density, the network throughputs achieved by the proposed, the max-utility, and the max-SINR schemes are 0.86 Gbits/s, 0.52 Gbits/s, and 1.12 Gbits/s, respectively. However, both the max-utility and the max-SINR schemes are ineffective to scenarios with 0.10 AV/m, 0.15 AV/m, and 0.20 AV/m, due to the high inter-cell interferences. From columns 2 to 5 in Table 4.2, the transmit powers of AP 2 and AP 3 for the proposed scheme have been adjusted, which helps to control inter-cell interference for both eNBs and the other two APs' transmissions. Despite the improvement in

network throughput, the computational complexity of the proposed scheme is higher than the other two, resulting in more iterations, as shown in columns 6 to 8 of Table 4.2. In which, N/A means not available and is used when a scheme is not working due to the limited amount of spectrum resources.

Table 4.2: Optimal transmit powers and number of iterations for the three schemes ($p = 0.8$).

AV Density (AV/m)	Optimal Transmit Powers P′(watts)				Number of Iterations		
	P_1'	P_2'	P_3'	P_4'	Proposed Scheme	max-utility Scheme	max-SINR Scheme
0.05	2.500	2.4054	2.4144	2.500	12	7	N/A
0.10	2.500	2.3840	2.3748	2.500	23	N/A	N/A
0.15	2.500	2.3761	2.3731	2.500	34	N/A	N/A
0.20	2.500	2.3699	2.3699	2.500	51	N/A	N/A

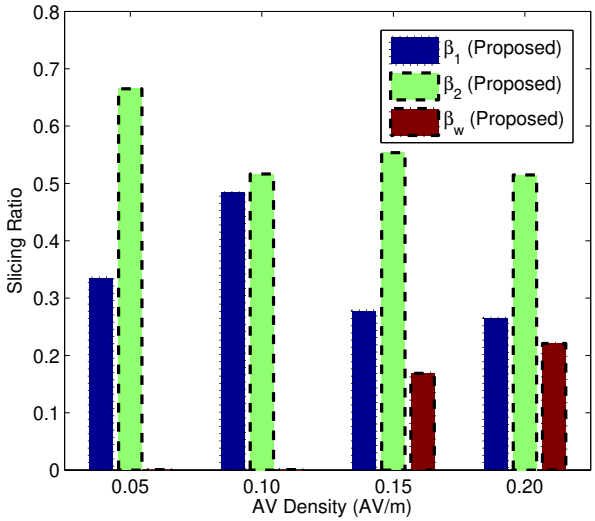


Figure 4.5: Spectrum slicing ratios under different AV density for the proposed scheme with $p = 0.8$.

In addition to adjusting the transmit powers for the APs, the spectrum slicing ratios among BSs are also adjusted by the proposed scheme, as shown in Figure 4.5. With AV density increasing from 0.05 AV/m to 0.20 AV/m, the amount of spectrum resources sliced to Wi-Fi APs, i.e., the spectrum slicing ratio β_w, is increased in the proposed scheme. This is because a large β_w indicates more spectrum resources can be reused among APs and therefore improving the spectrum efficiency.

4.6 Summary

In this chapter, we have proposed a dynamic spectrum management framework to enhance spectrum resource utilization in the MEC-assisted ADVNET with the consideration of cellular and Wi-Fi interworking. To maximize the aggregate network utility and provide QoS-guaranteed downlink transmission for delay-sensitive and delay-tolerant requests, three optimization problems have been investigated to slice spectrum among BSs fairly, to allocate spectrum among AVs associated with a BS in a QoS-guaranteed way, and to control transmit powers of Wi-Fi APs. In order to solve these three problems, we first use linear programming relaxation and first-order Taylor series approximation to transform them into tractable forms, and then design an ACS algorithm to jointly solve them. Based on the simulation results, the designed ACS algorithm has good convergence property within an acceptable number of iterations. Compared with two existing spectrum management schemes, the proposed framework is more robust to AV density changing and provides higher network throughput. In the next chapter, we will apply AI technology to jointly manage the multi-dimensional resources in the MVNET.

Chapter 5

Multi-Dimensional Resource Management in MVNETs

In this chapter, we study the joint allocation of the spectrum, computing, and caching resources in an MVNET. To support different vehicular applications, we consider two typical MEC architectures and formulate multi-dimensional resource optimization problems accordingly, which are usually with high computation complexity and overlong problem-solving time. Thus, we exploit RL to transform the two formulated problems and solve them by leveraging the DDPG and hierarchical learning architectures. Via off-line training, the network dynamics can be automatically learned and appropriate resource allocation decisions can be rapidly obtained to satisfy the QoS requirements of vehicular applications. From simulation results, the proposed resource management schemes can achieve high delay/QoS satisfaction ratios. The complete research results of this chapter can be seen in [54].

5.1 Background Information

To address the challenges in implementing MVNETs, many research works have been performed recently, including the design of architecture, task offloading scheme, resource management scheme, and so on. For example, the MEC-based hierarchical vehicular network framework, consisted of vehicle level's on-board computing/caching resources and server level's resources (resources placed at the MEC and cloud-computing servers), has been investigated in [13, 114–117]. To better manage the spectrum/computing/caching resources among and make task offloading decisions to vehicle users, task offloading and resource management schemes

have been proposed in [59, 115–117]. Since task offloading and spectrum/computing resource allocation are coupled with each other, the objectives of the most existing works have been achieved by jointly optimizing these two parts with traditional optimization methods [115, 116]. However, only one or two dimensions of resources have been considered in most of the existing schemes, which cannot be directly adopted to support some vehicular applications where high dimensional resources are involved, such as the computing tasks generated by the leading vehicle for platoon/convoy control [1]. Moreover, there are also some works focusing on multi-dimensional resources management in the scenarios with low mobility users [67, 68]. For MVNETs, the computational complexity of multi-dimensional resource management problems increases due to the high vehicle mobility and time-varying demand on resources, therefore increasing the time consumption on the resource management scheme itself. Therefore, it is infeasible to adopt the pure optimization approach-based schemes to achieve multi-dimensional resource management in MVNETs, especially for the scenarios with delay-sensitive applications. How to design practical and QoS-oriented multi-dimensional resource management schemes for the MVNETs still needs effort.

As is known, AI technology, especially RL, can be exploited to solve resource management problems quickly [118–121]. Q-learning [91, 117], deep Q-learning [90, 93, 122], actor-critic [67, 123], and other DRL algorithms have been widely exploited for resource management in wireless communication networks. Inspired by the existing works and considering the dynamic vehicular network environment caused by high vehicle mobility and heterogeneous applications, we investigate how to exploit DRL to jointly manage the spectrum, computing, and caching resources to support delay-sensitive applications in the MVNET [13] in this chapter. Specifically, the main contributions of this work can be summarized as follows,

1. According to the location of the MEC server, two typical multi-dimensional resource management frameworks are proposed with placing the MEC server at a macro-cell BS (MBS) and an edge node (EN)[1], respectively.

2. Leveraging optimization theory, optimization problems are formulated to maximize the number of offloaded tasks with satisfied QoS requirements and constrained total amounts of available spectrum, computing, and caching resources.

3. To rapidly solve the formulated problems and obtain optimal spectrum slicing among BSs and optimal spectrum/computing/caching allocation among vehicles, the formulated optimization problems are transformed with DRL.

[1] An edge node is the node placed at the edge of the core network, including gateway nodes, edge communication nodes, and so on.

4. A DDPG-based algorithm is proposed to solve the transformed RL problems. As the complexity of the transformed RL problems increases with the sizes of environment state and action, a hierarchical DDPG (HDDPG)-based algorithm is developed by combining the DDPG and the hierarchical learning architecture.

The rest of this chapter is organized as follows. First, the system model is presented in Section 5.2, including the spectrum management frameworks, communication model, and computing/caching resource allocation models. In Section 5.3, we formulate two optimization problems under two typical MEC architectures to jointly manage the multi-dimensional resources and transform them with DRL. Then, DDPG- and HDDPG-based algorithms are proposed in Section 5.4 to solve the transformed RL problems. Section 5.5 presents extensive simulation results to illustrate the performance of the proposed algorithms. Finally, we provide our concluding remarks in Section 5.6.

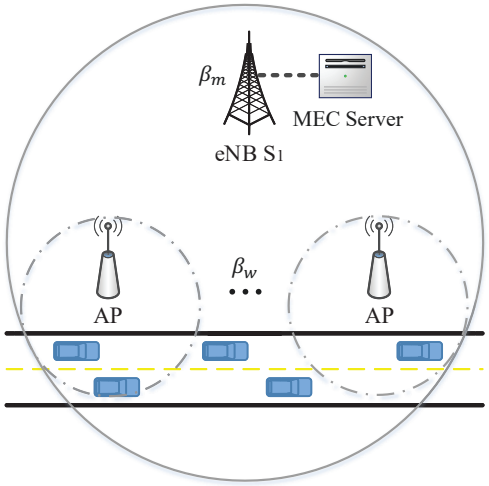
5.2 System Model

In this section, the models for spectrum management, vehicular communications, and computing/caching resource allocation under the considered MVNET are presented.

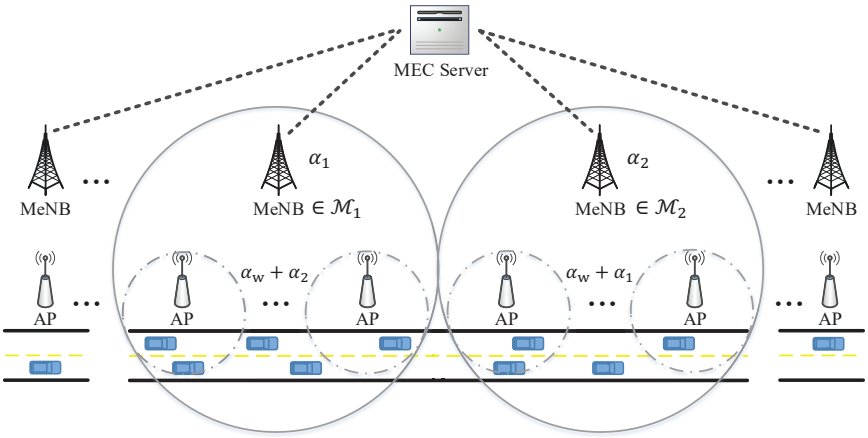
5.2.1 Spectrum Management Framework

The topology of MVNET and distribution of task offloading requests in the service area change frequently due to high vehicle mobility and heterogeneous applications. To offload vehicles' computing tasks to the MEC server with acceptable communication delay to satisfy the QoS requirements of each offloaded task, dynamic spectrum management frameworks are developed for two considered scenarios, placing the MEC server at an MeNB and at an EN, as shown in Figure 5.1. Considering vehicles on a two-lane straight country road with one lane for each direction, MeNBs and Wi-Fi APs are uniformly deployed on one side of the road with multiple Wi-Fi APs under the MeNB's coverage. In the following, we provide the detailed spectrum management procedures executed by the controllers installed at the MeNB- and EN-mounted MEC servers, including spectrum slicing among MeNBs and Wi-Fi APs and spectrum allocation among vehicles associated with the same BS.

MeNB-mounted MEC server: Denote $\mathcal{A}_m/A_m$ as the set/number of Wi-Fi APs under the service area of an MeNB-mounted MEC server, i.e., $\mathcal{A}_m = \{W_i, i = 1, ..., A_m\}$. Then the spectrum management procedures executed by the controller can be summarized into three steps:



(a) For the MeNB-mounted MEC server



(b) For the EN-mounted MEC server

Figure 5.1: Dynamic spectrum management frameworks.

(1) aggregating available spectrum resources from the MeNB and the A_m Wi-Fi APs into $R_{\max}$; (2) dynamically slicing the aggregated spectrum resources, $R_{\max}$, to the MeNB and the Wi-Fi APs with ratios β_m and β_w, respectively, where $\beta_m + \beta_w = 1$; and (3) allocating a proper fraction of spectrum to each vehicle's uplink transmission. To improve the spectrum efficiency with acceptable interference, spectrum reusing is enabled [124]. Vehicles associated with different Wi-Fi APs share spectrum $R_{\max}\beta_w$ to transmit computing tasks to the MeNB-mounted MEC server. Note that spectrum reusing among MeNBs is not considered in this scenario as it is beyond the scope of the control capability of the controller installed at the MeNB-mounted MEC server.

EN-mounted MEC server: To overcome the challenges caused by the limited serving time to each vehicle due to high vehicle mobility, the MEC server can be placed at an EN and with multiple MeNBs wired connected to it to enlarge its service area. According to the spatial adjacency relation among different MeNBs, the MeNBs under the service area of the EN-mounted MEC server are divided into two groups with sets/numbers $\mathcal{M}_1/M_1$ and $\mathcal{M}_2/M_2$, such that two MeNBs from the same group are not neighbored to each other. Spectrum slicing and spectrum reusing then are considered among the Wi-Fi APs and the two groups of MeNBs. By the controller installed at the EN-mounted MEC server, the aggregated spectrum resources, $R_{\max}$, are sliced with ratio set $\{\alpha_1, \alpha_2, \alpha_w\}$, and then reallocated to MeNBs in $\mathcal{M}_1$, MeNBs in $\mathcal{M}_2$, and Wi-Fi APs. In Figure 5.1(b), we take two adjacent MeNBs from $\mathcal{M}_1$ and $\mathcal{M}_2$ as the target MeNBs to illustrate the spectrum slicing results. Let $\mathcal{A}_j/A_j$ be the set/number of Wi-Fi APs within the coverage of MeNB S_j. Then for uplink transmission under the service area, considered spectrum reusing includes, (1) vehicles associated with MeNBs in $\mathcal{M}_1$ (or in $\mathcal{M}_2$) reuse the spectrum resource $R_{\max}\alpha_1$ (or $R_{\max}\alpha_2$); (2) vehicles associated with Wi-Fi APs reuse the spectrum resource $R_{\max}\alpha_w$; and (3) vehicles associated with a Wi-Fi AP reuse the spectrum resources sliced to the MeNBs that with no overlapping with this Wi-Fi AP, such as vehicles associated with the Wi-Fi APs in $\mathcal{A}_j$ (where $S_j \in \mathcal{M}_1$) reuse the spectrum resource $R_{\max}\alpha_2$ that is sliced to MeNBs in $\mathcal{M}_2$.

5.2.2 Communication Model

Assume the transmit powers of the MeNB and the Wi-Fi AP are fixed, full signal coverage is provided on the considered road segment by the MeNB, and there is no overlapping between any two Wi-Fi APs. Based on the duality theorem for the transmission in the same coherence interval [67], we assume the channels between a vehicle and a BS for uplink and downlink transmissions are symmetry. Let P be the transmit power of a vehicle. In the following, we analyze the spectrum efficiency and transmission rate achieved at the BS from the associating vehicles under the two dynamic spectrum management frameworks presented in Section 5.2.1.

Under the service area of the MeNB-mounted MEC server: As spectrum reusing is not considered among MeNBs in this scenario, the spectrum efficiency achieved at the MeNB from associating vehicle k can be given by

$$h_k^m = \log_2(1 + \frac{PG_k^m}{\sigma^2}), \tag{5.1}$$

where G_k^m (or $G_k'^i$) is the channel gain between vehicle k and the MeNB (or Wi-Fi AP W_i), which varies with the distance between vehicle k and the MeNB or Wi-Fi AP, and the calculating formulas are given in Section 5.5 in detail. σ^2 is the power of AWGN. Considering interference generated by the transmission under Wi-Fi AP $W_e \in \mathcal{A}_m$ ($e \neq i$), spectrum efficiency achieved at Wi-Fi AP W_i from vehicle k can be given by

$$h_k'^i = \log_2(1 + \frac{PG_k'^i}{\sum\limits_{W_e \in \mathcal{A}_m, e \neq i} PG_k'^e + \sigma^2}). \tag{5.2}$$

Let f_k^m (or $f_k'^i$) be the spectrum fraction allocated to vehicle k by the MeNB (or Wi-Fi AP W_i), namely, spectrum resource $R_{\max}\beta_m f_k^m$ (or $R_{\max}\beta_w f_k'^i$) are available for vehicle k's uplink transmission. Then the data rate of the uplink transmission from vehicle k to the MeNB (or Wi-Fi AP W_i) can be expressed as

$$R_k^m = R_{\max}\beta_m f_k^m h_k^m \ \ (\text{or } R_k'^i = R_{\max}\beta_w f_k'^i h_k'^i). \tag{5.3}$$

Under the service area of the EN-mounted MEC server: As shown in Figure 5.1(b), vehicles under the service area of the EN-mounted MEC server can access the MEC server through either an MeNB in $\mathcal{M}_1$ or $\mathcal{M}_2$, or a Wi-Fi AP. According to the considered spectrum reusing, we can obtain the spectrum efficiency achieved at the MeNB and Wi-Fi AP in this scenario accordingly. Taking the uplink transmission from vehicle k to MeNB $S_e \in \mathcal{M}_1$ as an example, experienced interference includes that from uplink transmission to MeNB $S_j \in \mathcal{M}_1$ ($j \neq e$) and from uplink transmission to Wi-Fi AP $W_i \in \mathcal{A}_j$ ($S_j \in \mathcal{M}_2$). And the spectrum efficiency achieved at MeNB S_e from vehicle k then can be given by

$$h_k^e = \log_2(1 + \frac{PG_k^e}{\sum\limits_{S_j \in \mathcal{M}_1, j \neq e} PG_k^j + \sum\limits_{S_j \in \mathcal{M}_2} \sum\limits_{W_i \in \mathcal{A}_j} PG_k'^i + \sigma^2}). \tag{5.4}$$

Let f_k^e be the spectrum fraction allocated to vehicle k by MeNB $S_e \in \mathcal{M}_1$. As the spectrum

resource sliced to MeNBs in $\mathcal{M}_1$ is $R_{\max}\alpha_1$, the transmission rate from vehicle k to MeNB S_e is

$$R_k^e = R_{\max}\alpha_1 f_k^e h_k^e. \tag{5.5}$$

Similarly, for the uplink transmission from vehicle k to MeNB $S_u \in \mathcal{M}_2$, we can also get the achievable spectrum efficiency h_k^u and transmission rate R_k^u accordingly.

For uplink transmission from vehicle k to the associated Wi-Fi AP, the achievable spectrum efficiency includes two parts since spectrum sharing is considered between a Wi-Fi AP and an MeNB and between two Wi-Fi APs. Taking Wi-Fi AP $W_i \in \mathcal{A}_j$ ($S_j \in \mathcal{M}_1$) as an example. According to spectrum reusing shown in Figure 5.1(b), both spectrum resource $R_{\max}\alpha_2$ and $R_{\max}\alpha_w$ are available for uplinks under Wi-Fi AP W_i. Using subscripts m and w to distinguish the reused spectrum resources that are sliced to MeNBs and Wi-Fi APs, respectively, i.e., $R_{\max}\alpha_2$ and $R_{\max}\alpha_w$. Then the corresponding spectrum efficiency achieved from these two parts can be described as

$$h_{m,k}^{\prime i} = \log_2(1 + \frac{PG_k^{\prime i}}{\sum\limits_{W_g \in \mathcal{A}_{m_1}, g \neq i} PG_k^{\prime g} + \sum\limits_{S_j \in \mathcal{M}_2} PG_k^j + \sigma^2}) \tag{5.6}$$

and

$$h_{w,k}^{\prime i} = \log_2(1 + \frac{PG_k^{\prime i}}{\sum\limits_{W_g \in \{\mathcal{A}_{m_1} \cup \mathcal{A}_{m_2}\}, g \neq i} PG_k^{\prime g} + \sigma^2}), \tag{5.7}$$

respectively, where $\mathcal{A}_{m_1}$ and $\mathcal{A}_{m_2}$ are the sets of Wi-Fi APs within the coverages of MeNBs in $\mathcal{M}_1$ and in $\mathcal{M}_2$, respectively. Let $f_{m,k}^{\prime i}$ and $f_{w,k}^{\prime i}$ be the spectrum fractions allocated to vehicle k by Wi-Fi AP W_i from $R_{\max}\alpha_2$ and from $R_{\max}\alpha_w$, respectively. Then, the transmission rate achieved by the uplink from vehicle k to Wi-Fi AP W_i can be given by

$$R_{j,k}^{\prime i} = R_{\max}\alpha_2 f_{m,k}^{\prime i} h_{m,k}^{\prime i} + R_{\max}\alpha_w f_{w,k}^{\prime i} h_{w,k}^{\prime i}, \tag{5.8}$$

where the subscript j of $R_{j,k}^{\prime i}$ indicates that Wi-Fi AP W_i is within the coverage of MeNB S_j.

5.2.3 Computing and Caching Resource Allocation Models

For vehicles under the service area of the MEC server, we assume each vehicle randomly generates computing tasks, such as image processing tasks for assisting in automatic driving [125], and periodically sends them to the MEC server with their driving state information

together. Denote $\{c_k^s, c_k^c, d_k\}$ as the computing task generated by vehicle k, where c_k^s is the data size of the computing task that needs to be transmitted to and cached by the MEC server, c_k^c denotes the required number of CPU cycles for task execution, and d_k is the maximum delay tolerated by the task. Without loss of generality, we assume a vehicle generates different computing tasks during different time slots and the vehicle distribution under the considered road segment changes with time.

Due to the high and stable data rate achieved by the wired connection between a BS and the MEC server, the transmission time in a wired link is relatively small and is neglected in this chapter. Assume each MEC server is equipped with computing capabilities of $C_{\max}^c$ CPU cycles per second (i.e., Hz) and caching capabilities of $C_{\max}^s$ kbits. Let f_k^c be the fraction of computing resources allocated to vehicle k. Then, under the service area of an MeNB-mounted MEC server, the total time consumption on offloading and executing the computing task generated by vehicle k can be expressed as

$$T_k = \begin{cases} \frac{c_k^s}{R_k^m} + \frac{c_k^c}{C_{\max}^c f_k^c}, & \text{if } v_k^m = 1 \\ \frac{c_k^s}{R_k'^i} + \frac{c_k^c}{C_{\max}^c f_k^c}, & \text{if } v_k'^i = 1, \end{cases} \tag{5.9}$$

where binary variable v_k^m (or $v_k'^i$) is the association pattern between vehicle k and the MeNB (or Wi-Fi AP W_i), which equals to 1 if vehicle k associates to the MeNB (or Wi-Fi AP W_i), and 0 otherwise. We assume a vehicle only associates with an MeNB when it is outside of the coverage of any Wi-Fi AP. For vehicles under the service area of an EN-mounted MEC server, let v_k^j (or $v_{j,k}'^i$) be the association pattern between vehicle k and MeNB S_j (or Wi-Fi AP W_i in $\mathcal{A}_j$). Then we can express the total time consumption on offloading vehicle k's task through MeNB S_j or Wi-Fi AP $W_i \in \mathcal{A}_j$ to and executing the task at the EN-mounted MEC server as

$$T_k = \begin{cases} \frac{c_k^s}{R_k^j} + \frac{c_k^c}{C_{\max}^c f_k^c}, & \text{if } v_k^j = 1 \\ \frac{c_k^s}{R_{j,k}'^i} + \frac{c_k^c}{C_{\max}^c f_k^c}, & \text{if } v_{j,k}'^i = 1. \end{cases} \tag{5.10}$$

Denote f_k^s as the fraction of caching resources allocated to vehicle k. Then, we call vehicle k's offloaded task is completed with satisfied QoS requirements when the following two conditions are satisfied: (1) at least c_k^s caching resources are allocated to vehicle k, i.e., $f_k^s C_{\max}^s \geq c_k^s$, and (2) the spectrum and computing resources allocated to vehicle k are enough for transmitting the c_k^s data to and executing the task at the MEC server with a total time cost less than d_k.

5.3 Problem Formulation and Transformation

In this section, we first formulate problems to jointly manage the spectrum, computing, and caching resources, and then transform the formulated problems based on RL.

5.3.1 Problem Formulation

In the MVNET, moving vehicles send their moving state, position, and task information, i.e., $\{c_k^s, c_k^c, d_k\}$, to the MEC server. Then based on the collected information, the controller centrally manages the spectrum, computing, and caching resources among vehicles with task offloading requests. How to efficiently allocate the limited three dimensions of resources to support as many task offloading requests with satisfied QoS requirements as possible is important to the MVNETs. To achieve this goal, problems are formulated to maximize the numbers of offloaded tasks that are completed with satisfied QoS requirements by the MeNB- and EN-mounted MEC servers. The one for the MeNB-mounted MEC server is as follows,

$$\max_{\substack{\beta_m,\beta_w \\ \mathbf{f},\mathbf{f}^c,\mathbf{f}^s}} \sum_{k\in\mathcal{N}} H(d_k - T_k)H(f_k^s C_{\max}^s - c_k^s), \tag{5.11}$$

$$\text{s.t.} \begin{cases} \beta_m, \beta_w \in [0,1] & \text{(5.11a)} \\ \beta_m + \beta_w = 1 & \text{(5.11b)} \\ f_k^m, f_k'^i, f_k^c, f_k^s \in [0,1], \quad k \in \mathcal{N} & \text{(5.11c)} \\ \sum\limits_{k\in\mathcal{N}_m} f_k^m = 1 & \text{(5.11d)} \\ \sum\limits_{k\in\mathcal{N}^i} f_k'^i = 1, \quad \forall i & \text{(5.11e)} \\ \sum\limits_{k\in\mathcal{N}} f_k^c = 1 & \text{(5.11f)} \\ \sum\limits_{k\in\mathcal{N}} f_k^s = 1, & \text{(5.11g)} \end{cases}$$

where T_k is given by equation (5.9), $\mathcal{N}$ is the set of vehicles under the service area of the MEC server, and $\mathcal{N}_m$ (or $\mathcal{N}^i$) is the set of vehicles associated with the MeNB (or Wi-Fi AP W_i). $\mathbf{f} = \{f_k^m, f_k'^i\}$ $(k \in \mathcal{N})$ is the spectrum allocation matrix to vehicles by the MeNB and Wi-Fi APs W_i. $\mathbf{f}^c$ and $\mathbf{f}^s$ are the matrices describing computing and caching resources allocated to vehicles by the MEC server, respectively. To describe an offloaded task with satisfied QoS requirements,

the Heaviside step function denoted by $H(\cdot)$ is involved in the objective function, which is 1 if the variable is larger or equal to 0, and 0 otherwise. Then, for an offloaded task that is generated by vehicle k and completed with satisfied QoS requirements, we have $H(d_k - T_k)H(f_k^s C_{\max}^s - c_k^s) = 1$. Similarly, according to equation (5.10) and the communication model corresponding to Figure 5.1(b), a joint resource management optimization problem can be also formulated for the scenario with EN-mounted MEC servers.

5.3.2 Problem Transformation with DRL

As the two formulated optimization problems are both non-convex due to the Heaviside step function, traditional optimization methods are infeasible without transforming the original objective functions. Due to the coupled relation among optimization variables β_m, β_w, $\mathbf{f}$, $\mathbf{f}^c$, and $\mathbf{f}^s$, the original problems have to be decomposed into subproblems and then leverage some alternate concave search algorithms to solve them [126]. Nevertheless, overlong solving time is resulted due to the alternating process among subproblems, which makes it impossible to satisfy the stringent delay requirements and would be further increased with the number of vehicles under the service area of the MEC server.

It is critical to rapidly obtain an optimal resource allocation decision for a given dynamic environment state with delay-sensitive tasks. Thus, we model the above resource allocation decision making problems as Markov decision processes (MDPs) [127], and then adopt DRL methods to solve them [128]. As shown in Figure 5.2, the fundamental DRL architecture consists of agent and environment interacting with each other [51]. The agent is implemented by the controller installed at each MEC server and everything beyond the controller is regarded as the environment. Through learning the best policy (called as resource allocation policy that maps the environment state to a resource allocation decision) to maximize the total accumulated reward, the above problems can be solved directly.

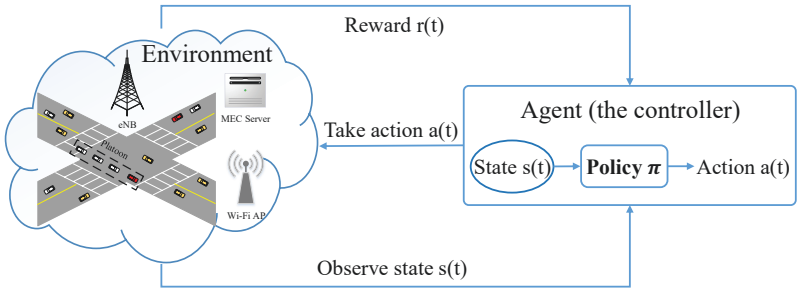


Figure 5.2: The fundamental DRL architecture in the MVNET.

Environment state: As discussed before, each vehicle periodically sends the driving state information and task information to the MEC server. By collecting such information, the agent (i.e., the controller) can obtain the environment state. Denote $\mathcal{S}$ as the state space, $N(t) = |\mathcal{N}|$ as the number of vehicles under the service area of the MEC server at time slot t, and $x_k(t)$ and $y_k(t)$ as the x- and y- coordinates of the position of vehicle k. Then, the environment state at time slot t, $s(t) \in \mathcal{S}$, can be described as

$$\begin{aligned} s(t) =&\{x_1(t), x_2(t), \ldots, x_{N(t)}(t), y_1(t), y_2(t), \ldots, y_{N(t)}(t), c_1^s(t), c_2^s(t), \ldots, \\ &c_{N(t)}^s(t), c_1^c(t), c_2^c(t), \ldots, c_{N(t)}^c(t), d_1(t), d_2(t), \ldots, d_{N(t)}(t)\}, \end{aligned} \tag{5.12}$$

According to each vehicle's position information, the uplink channel gain from a vehicle to a BS can be obtained by the agent. Note that the environment state should be adjusted according to the association patterns between vehicles and MeNBs (or Wi-Fi APs).

Action: Based on the observed environment states in $\mathcal{S}$, the agent will make resource allocation decisions according to the resource allocation policy π. Denote $\mathcal{A}$ as the action space. Then the action taken by the MeNB-mounted MEC server at time slot t, including the spectrum slicing ratio set $\{\beta_m(t), \beta_w(t)\}$, spectrum allocation fraction sets for the MeNB $f_k^m(t)$ and for each Wi-Fi AP $f_k'^i(t)$ ($W_i \in \mathcal{A}_m$), computing resource allocation fraction $f_k^c(t)$, and caching resource allocation fraction $f_k^s(t)$, can be given by

$$\begin{aligned} a(t) =&\{\beta_m(t), \beta_w(t), f_1^m(t), f_2^m(t), \ldots, f_{N(t)}^m(t), f_1'^i(t), f_2'^i(t), \ldots, f_{N^i(t)}'^i(t), f_1^s(t), \\ &f_2^s(t), \ldots, f_{N(t)}^s(t), f_1^c(t), f_2^c(t), \ldots, f_{N(t)}^c(t)\}, \forall W_i \in \mathcal{A}_m \end{aligned} \tag{5.13}$$

and that for the EN-mounted MEC server is

$$\begin{aligned} a(t) =&\{\alpha_1(t), \alpha_2(t), \alpha_w(t), f_1^m(t), f_2^m(t), \ldots, f_{N(t)}^m(t), f_1'^i(t), f_2'^i(t), \ldots, f_{N^i(t)}'^i(t), \\ &f_1^s(t), f_2^s(t), \ldots, f_{N(t)}^s(t), f_1^c(t), f_2^c(t), \ldots, f_{N(t)}^c(t)\}, \forall W_i \in \{\mathcal{A}_{m_1} \cup \mathcal{A}_{m_2}\}, \end{aligned} \tag{5.14}$$

where $N^i(t) = |\mathcal{N}^i|$ is the number of vehicles associated with Wi-Fi AP W_i at time slot t.

Reward: As shown in Figure 5.2, once the agent takes action $a(t-1)$ based on the observed environment state $s(t-1)$, the environment will return an immediate reward $r(t)$ to the agent[2]. Then in the learning stage, the agent updates the resource allocation policy, π, based on the received reward until the algorithm converged. Indicated by equation (5.11), the delay requirement and requested caching resources should be simultaneously satisfied to guarantee the QoS requirements of an offloaded task. Thus, to maximize the number of offloaded tasks that

[2]In RL, the immediate reward at time slot t, $r(t)$, is the consequence of action token at the previous time slot, $a(t-1)$.

are completed with satisfied QoS requirements by the MEC server at time slot t, we define the following two reward elements for offloaded task k that are corresponding to $H(d_k - T_k)$ and $H(f_k^s C_{\max}^s - c_k^s)$ of equation (5.11),

$$r_k^d(t) = \log_2(\frac{d_k(t-1)}{T_k(t-1)} + 0.00095) \tag{5.15}$$

$$r_k^s(t) = \log_2(\frac{C_{\max}^s f_k^s(t-1)}{c_k^s(t-1)} + 0.00095), \tag{5.16}$$

where $r_k^d(t) \geq 0$ when the delay requirement $d_k(t-1)$ is satisfied by the allocated spectrum and computing resources, and $r_k^s(t) \geq 0$ when enough caching resources are allocated to task k, otherwise negative $r_k^d(t)$ and $r_k^s(t)$ are obtained. To improve the convergence performance of the RL algorithm and the fairness among vehicles, we use the logarithm function to define the reward elements. And a small value 0.00095 is added when calculating the logarithm reward to the base 2, such that the minimum value of each reward element is limited to -10 to avoid sharp fluctuation.

5.4 DDPG Algorithm based Solution

According to whether the agent can learn the environment for decision making in advance or not, DRL algorithms can be classified into two categories: model-based and model-free. In the considered MVNET, the environment state dynamically changes over time due to the infinite channel states and dynamic task offloading requests. That is, the agent cannot make a resource allocation decision for the subsequent time slot according to the current observed environment state. Thus, we consider a model-free DRL algorithm with an uncertain environment in this work. Moreover, the environment state and action vectors given in the previous section indicate that the sizes of state space $\mathcal{S}$ and action space $\mathcal{A}$ are infinite. Hence, policy-based[3] and model-free DRL algorithms, such as policy gradient, actor-critic, deterministic policy gradient (DPG), and DDPG, should be adopted. In this work, we combine the DDPG with normalization and hierarchical learning architecture to solve the modeled MDPs.

DDPG is an improved actor-critic algorithm, which combines the advantages of policy gradient and deep Q-network (DQN) algorithms. As the DDPG architecture shown in Figure 5.3,

[3]As opposed to the value-based RL algorithm, the policy-based DRL algorithm makes an action decision according to actions' probabilities, and can be applied to scenarios with infinite state and action spaces.

the agent is more complex compared with the fundamental DRL architecture shown in Figure 5.2, and is mainly consisted of two entities, actor and critic. Similar to DQN, both actor and critic adopt target nets with soft updated parameters to achieve stable convergence. We use two DNNs with the same structure but different parameters, i.e., an evaluation net with real-time updated parameters and a target net with soft updated parameters, to realize the actor and critic. Let $\boldsymbol{\theta}^{\mu}$ and $\boldsymbol{\theta}^{\mu'}$ (or $\boldsymbol{\theta}^{Q}$ and $\boldsymbol{\theta}^{Q'}$) be the parameter matrices of the evaluation net and target net in the actor (or in the critic), respectively. Then the parameters of the target nets are soft updated with that of the evaluation nets as follows,

$$\boldsymbol{\theta}^{\mu'} = \kappa_a \boldsymbol{\theta}^{\mu} + (1 - \kappa_a)\boldsymbol{\theta}^{\mu'} \tag{5.17}$$

$$\boldsymbol{\theta}^{Q'} = \kappa_c \boldsymbol{\theta}^{Q} + (1 - \kappa_c)\boldsymbol{\theta}^{Q'} \tag{5.18}$$

with $\kappa_a \ll 1$ and $\kappa_c \ll 1$. Denote $\mu(\cdot)$ and $\mu'(\cdot)$ (or $Q(\cdot)$ and $Q'(\cdot)$) as the network functions of the evaluation and target nets in the actor (or in the critic). Same as the fundamental RL algorithm, the agent first makes a resource allocation action $a(t) = \mu(s(t))$ according to the observed environment state $s(t)$, and then waits for reward $r(t)$ and next subsequent state s' from the environment.

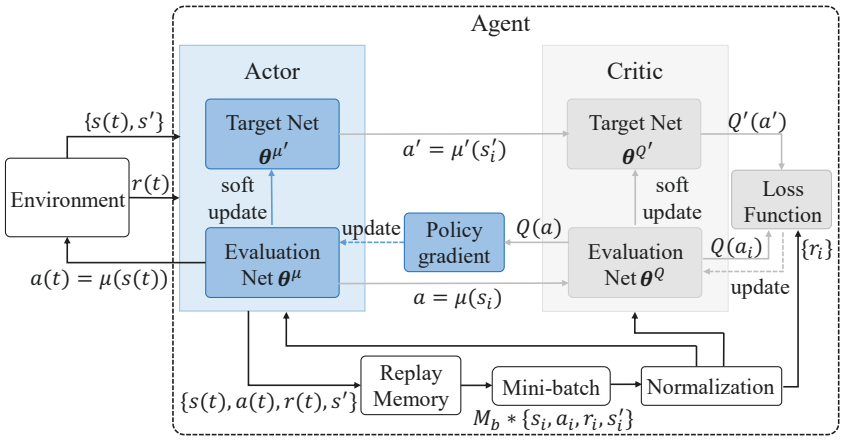


Figure 5.3: The architecture of the DDPG learning.

To improve the performance and stability of the evaluation and target nets and accelerate the convergence rate of DDPG, experience replay and input normalization are considered. As shown in Figure 5.3, each experience of the agent, denoted by $\{s(t), a(t), r(t), s'\}$, is saved in the replay memory. Assume up to M_r experiences can be saved in the replay memory. Then once the

number of experiences saved in the replay memory reaches to M_r, the experience first saved will be replaced by the new coming experience and the learning stage of the DDPG starts. In each step during the learning stage, the agent randomly chooses a mini batch of M_b experiences from the replay memory, denoted by $\{s_i, a_i, r_i, s_i'\}$ $(i = 1, \ldots, M_b)$, to update $\boldsymbol{\theta}^{\mu}$ and $\boldsymbol{\theta}^{Q}$. By randomly choosing experiences from the replay memory, the correlation among experiences are disrupted, and therefore accelerating the convergence rate. Moreover, considering a great difference among data elements of an experience would result in a large number of artificial neurons with inactive outputs[4], each experience is locally normalized before inputted to the actor and critic. Based on the architecture shown in Figure 5.3, two algorithms, i.e., DDPG- and HDDPG-based algorithms, are designed to solve the MDPs in the MVNET.

5.4.1 DDPG-based Algorithm

The DDPG-based algorithm is with exactly the same architecture in Figure 5.3. For the two considered scenarios with MeNB- and EN-mounted MEC servers, the environment state vector at time slot t is given by equation (6.8). According to the basic idea of DDPG [129], the goal of the agent is to find an optimal resource allocation policy π with the maximum achievable long-term average reward, $E\{Q^{\pi}(s, a)\}$, and can be approached by the learning stage step by step. In which, $Q^{\pi}(s, a)$ is the standard Q-value function used to evaluate the state-action value under policy π, and can be given by

$$Q^{\pi}(s, a) = E\left\{\sum_{\tau=0}^{\infty} \gamma^{\tau} r(t+\tau) \big| \pi, s = s(t), a = a(t)\right\}, \tag{5.19}$$

where $r(t)$ is the immediate reward received by the agent at time slot t, $\gamma \in (0, 1)$ is a discount factor on $r(t)$, and τ is a time slot counting symbol. In order to achieve the objective functions of the formulated optimization problems, we define the reward achieved by vehicle k at time slot t, $r_k(t)$, as

$$r_k(t) = r_k^d(t) + r_k^s(t), \tag{5.20}$$

such that a positive reward can be guaranteed once vehicle k's offloaded task is completed with satisfied QoS requirements by the MEC server. Considering the vehicle density changes over

[4]An output that is on the boundary of the output range is called as an inactive output of the artificial neuron, which depends on the input data and the enabled activation function. For example, -1 and 1 are inactive outputs of an artificial neuron with the tanh activation function. A large number of artificial neurons with inactive outputs would reduce the convergence rate of the learning algorithm.

time, the immediate reward, $r(t)$, then is defined as the average reward over vehicles within the service area of the MeNB- or EN-mounted MEC server, i.e., $r(t) = \frac{1}{N(t)} \sum_{k \in \mathcal{N}} r_k(t)$.

For the DDPG-based algorithm, the evaluation net of the actor is the one that makes the resource allocation action for each given environment state. Thus, how to update the parameter matrix, $\boldsymbol{\theta}^{\mu}$, during the learning stage is the key to find the optimal π in DDPG. As mentioned, in each step during the learning stage, a mini batch of experiences are randomly chosen from the replay memory and then inputted to the agent one by one. With each inputted experience, the actor and critic interact with each other to update the parameter matrices of their evaluation nets, $\boldsymbol{\theta}^{\mu}$ and $\boldsymbol{\theta}^{Q}$. Taking the i-th experience from the mini batch of experiences, $\{s_i, a_i, r_i, s_i'\}$, as an example, the interaction can be summarized as, (1) the actor makes resource allocation actions according to the two adjacent environment states, $a = \mu(s_i)$ and $a' = \mu'(s_i')$, by using the evaluation and target nets, respectively; (2) the critic evaluates a and a_i with its evaluation net[5], $Q(a)$ and $Q(a_i)$, and evaluates a' with its target net, i.e., $Q'(a')$; (3) the actor updates $\boldsymbol{\theta}^{\mu}$ according to the policy gradient to maximize $E\{Q(a)\}$, and the critic updates $\boldsymbol{\theta}^{Q}$ according to the loss function to minimize the temporal difference (TD)-error for each inputted experience. Here, the TD-error describes the difference between the estimated Q-value $Q(a_i)$ and the target Q-value $r_i + \gamma Q'(a')$, which can be given by

$$\varepsilon_i = Q(a_i) - (r_i + \gamma Q'(a')). \tag{5.21}$$

And then, the loss function can be described as $L(\boldsymbol{\theta}^{Q}) = E\{(\varepsilon_i)^2\}$.

From the critic perspective, the network functions of the evaluation and target nets are in charge of estimating the Q-value functions. As the achievable Q-value under the optimal policy π is $Q^{\pi}(s, a)$, i.e., $\pi(a|s) = \arg\max_{\mathrm{a}} \mathrm{Q}^{\pi}(\mathrm{s}, \mathrm{a})$ [130], the parameter updating in the critic during the learning stage is similar to DQN to make $Q(a)$ approximate $Q^{\pi}(s, a)$. As mentioned, a mini batch of M_b experiences are adopted in each step during the learning stage. Assume that the loss function, $L(\boldsymbol{\theta}^{Q})$, is continuously differentiable with respect to $\boldsymbol{\theta}^{Q}$. Then the critic can update $\boldsymbol{\theta}^{Q}$ with the gradient of $L(\boldsymbol{\theta}^{Q})$ as follows,

$$\Delta \boldsymbol{\theta}^{Q} = \lambda_c \frac{1}{M_b} \sum_{\mathcal{M}_b} \varepsilon_i \nabla_{\boldsymbol{\theta}^{Q}} Q(a_i), \tag{5.22}$$

where λ_c is the learning rate of the critic and $\nabla_{\boldsymbol{\theta}^{Q}} Q(a_i)$ is the derivative of $L(\boldsymbol{\theta}^{Q})$ with respect to $\boldsymbol{\theta}^{Q}$.

The role of the actor is to make a resource allocation action for each given environment state.

[5]Even though s_i and a_i are the state-action pair and we have $a = \mu(s_i)$, a_i and a may be two different actions due to the updating on the parameter matrix $\boldsymbol{\theta}^{\mu}$. Thus, $Q(a)$ and $Q(a_i)$ may also be different.

Considering the goal of the agent, the actor should update its parameter matrix to make $\mu(s_i)$ approximate the optimal resource allocation policy π to achieve the maximum $E\{Q^{\pi}(s,a)\}$. Thus, the policy objective function, used to evaluate the policy under a given parameter matrix $\boldsymbol{\theta}^{\mu}$, can be defined as $J(\boldsymbol{\theta}^{\mu}) = E\{Q(a)\}$ as in [131], where $a = \mu(s_i)$. The actor adopts the policy gradient method to update the parameter matrix $\boldsymbol{\theta}^{\mu}$ during the learning stage. Similar to the critic, when the policy objective function $J(\boldsymbol{\theta}^{\mu})$ is continuously differentiable with respect to $\boldsymbol{\theta}^{\mu}$, $\boldsymbol{\theta}^{\mu}$ then can be updated with the gradients of $J(\boldsymbol{\theta}^{\mu})$ as follows,

$$\Delta\boldsymbol{\theta}^{\mu} = \lambda_a \frac{1}{M_b} \sum_{\mathcal{M}_b} \nabla_a Q(a)|_{a=\mu(s_i)} \nabla_{\boldsymbol{\theta}^{\mu}} \mu(s_i), \tag{5.23}$$

where λ_a is the learning rate of the actor, $\nabla_a Q(a)|_{a=\mu(s_i)}$ is the derivative of $Q(a)$ with respect to a with $a = \mu(s_i)$, and $\nabla_{\boldsymbol{\theta}^{\mu}} \mu(s_i)$ is the derivative of $\mu(s_i)$ with respect to $\boldsymbol{\theta}^{\mu}$. Note that equations (5.22) and (5.23) are obtained by using the derivative method for compound function.

Combining with the parameter updating processes in the critic and actor, the DDPG-based algorithm can be summarized in Algorithm 2. To better observe the convergence of the algorithm, the total rewards per episode, r_{ep}, are considered in the DDPG-based algorithm. Let M_s be the number of steps in each episode, then for the episode from time slot t_0 to $t_0 + M_s$, we have $r_{ep} = \sum_{t=t_0}^{M_s+t_0} r(t)$.

When leveraging the DDPG-based algorithm to solve the MDPs in the scenarios with MeNB- and EN-mounted MEC servers, the state and action vectors are given by equations (6.8), (5.13), and (5.14). The three equations indicate that the sizes of state and action vectors increase linearly with the number of vehicles under the service area. That is more complex MDP would be modeled due to the increase of vehicle density in or service area of an MEC server, and therefore resulting in longer convergence time of the DDPG-based algorithm. Moreover, the total time consumption on offloading and executing vehicle k's task, $T_k(t)$ (given by equations (5.9) or (5.10)), is co-determined by the spectrum and computing resource allocation results. In an action, the spectrum slicing ratios and the spectrum/computing allocation fractions are the elements related to spectrum and computing resource allocation. And the impacts of these elements on the reward achieved by vehicle k are only implied by r_k^d. With the MDP becoming more complex, it will be more difficult for the DDPG-based algorithm to learn the impacts of such elements, and therefore resulting in longer convergence time. To overcome the above issues, we combine the architecture shown in Figure 5.3 with the hierarchical learning architecture [132], and propose an HDDPG-based algorithm in the next subsection.

/* Initialization phase */
Initialize the evaluation and target nets for the actor and critic with parameter matrices $\boldsymbol{\theta}^{\mu}$, $\boldsymbol{\theta}^{\mu'}$, $\boldsymbol{\theta}^{Q}$, and $\boldsymbol{\theta}^{Q'}$;
Initialize the replay memory buffer.
/* Parameter updating phase */
foreach *episode* **do**
  Receive initial observation state s^0 and reset $r_{ep} = 0$.
  foreach *step* t **do**
    Make an action $a(t) = \mu(s(t))$ with the parameter matrix $\boldsymbol{\theta}^{\mu}$;
    Receive the subsequent state s' and reward $r(t)$;
    if *the number of experiences* $< M_r$ **then**
      Store the experience $\{s(t), a(t), r(t), s'\}$ into the replay memory;
    else
      Replace the first saved experience with $\{s(t), a(t), r(t), s'\}$ in the replay memory;
      Randomly select a mini batch of M_b experiences from the replay memory;
      Update the parameter matrices in the critic:
      $\boldsymbol{\theta}^{Q} \leftarrow \boldsymbol{\theta}^{Q} + \Delta\boldsymbol{\theta}^{Q}$
      $\boldsymbol{\theta}^{Q'} = \kappa_c \boldsymbol{\theta}^{Q} + (1 - \kappa_c)\boldsymbol{\theta}^{Q'}$;
      Update the parameter matrices in the actor:
      $\boldsymbol{\theta}^{\mu} \leftarrow \boldsymbol{\theta}^{\mu} + \Delta\boldsymbol{\theta}^{\mu}$
      $\boldsymbol{\theta}^{\mu'} = \kappa_a \boldsymbol{\theta}^{\mu} + (1 - \kappa_a)\boldsymbol{\theta}^{\mu'}$.
    end
    $r_{ep} = r_{ep} + r(t)$.
  end
end

Algorithm 2: The DDPG-based algorithm

5.4.2 HDDPG-based Algorithm

According to the DDPG architecture shown in Figure 5.3 and the hierarchical learning architecture, the main idea of the HDDPG-based algorithm is to decompose the original MDP into sub-MDPs. By involving a variable matrix $\boldsymbol{\nu} = \{\nu_1, \nu_2, \ldots, \nu_N\}$, defined as the proportion of delay allowed by each vehicle's data transmission, the original resource allocation decision making problem can be decomposed into two subproblems for spectrum allocation and computing/caching resource allocation. The spectrum allocation subproblem is in charge of spectrum slicing among BSs and spectrum allocation among vehicles while satisfying the communication delay requirement $\nu_k d_k(t)$ ($0 \leq \nu_k \leq 1$) of vehicle k's task. The computing/caching resource allocation subproblem is to manage the MEC server's computing and caching resources among the received tasks. Then we solve the two subproblems with two DDPG algorithms (spectrum DDPG algorithm and computing DDPG algorithm). For the spectrum DDPG algorithm, the state and action vectors can be given by

$$\begin{aligned} s_{sp}(t) =&\{x_1(t), x_2(t), \ldots, x_{N(t)}(t), y_1(t), y_2(t), \ldots, y_{N(t)}(t), c_1^s(t), c_2^s(t), \ldots, \\ & c_{N(t)}^s(t), \nu_1 d_1(t), \nu_2 d_2(t), \ldots, \nu_{N(t)} d_{N(t)}(t)\} \end{aligned} \tag{5.24}$$

and

$$\begin{aligned} a_{sp}(t) =&\{\alpha_1(t), \alpha_2(t), \alpha_w(t), f_1^m(t), f_2^m(t), \ldots, f_{N(t)}^m(t), f_1'^i(t), f_2'^i(t), \ldots, \\ & f_{N^i(t)}'^i(t)\}, \forall W_i \in \{\mathcal{A}_{m_1} \cup \mathcal{A}_{m_2}\} \end{aligned} \tag{5.25}$$

respectively. At time slot t, each vehicle's time consumption on data transmission, denoted by $\boldsymbol{T'} = \{T_1'(t), T_2'(t), \ldots, T_{N(t)}'(t)\}$, can be obtained by taking action $a_{sp}(t)$ on the current state. Then, for the spectrum DDPG algorithm, we define the reward achieved by vehicle k as

$$r_k'(t) = \log_2(\frac{\nu_k d_k(t-1)}{T_k'(t-1)} + 0.00095). \tag{5.26}$$

And the immediate reward received by the agent at time slot t, $r_{sp}(t)$, can be given by $r_{sp}(t) = \frac{1}{N(t)} \sum_{k \in \mathcal{N}} r_k'(t)$.

Similarly, we can define the environment state and action vectors for the computing DDPG algorithm as follows,

$$\begin{aligned} s_{cp}(t) =&\{T_1'(t), T_2'(t), \ldots, T_{N(t)}'(t), c_1^s(t), c_2^s(t), \ldots, c_{N(t)}^s(t), c_1^c(t), \\ & c_2^c(t), \ldots, c_{N(t)}^c(t), d_1(t), d_2(t), \ldots, d_{N(t)}(t)\} \end{aligned} \tag{5.27}$$

and

$$a_{cp}(t) = \{f_1^s(t), f_2^s(t), \ldots, f_{N(t)}^s(t), f_1^c(t), f_2^c(t), \ldots, f_{N(t)}^c(t)\}. \tag{5.28}$$

By updating equation (5.10) with $T_k(t) = T'_k(t) + c_k^c(t)/(C_{\max}^c f_k^c(t))$, the reward achieved by vehicle k in the computing DDPG algorithm can be given by equation (5.20). And the immediate reward received by the agent of the computing DDPG is $r_{cp}(t) = \frac{1}{N(t)} \sum_{k \in \mathcal{N}} r_k(t)$.

As indicated by the action vector in the computing DDPG algorithm, the computing DDPG algorithm correlates with the spectrum DDPG algorithm. Thus, we regard the action made by the agent of the spectrum DDPG as a part of the environment of the computing DDPG algorithm. To distinguish the two DDPG architectures in the HDDPG, a subscript is added to the notations defined in Section 5.4.1. For example, $\{s_{sp}(t), a_{sp}(t), r_{sp}(t), s'_{sp}\}$ and $\{s_{cp}(t), a_{cp}(t), r_{cp}(t), s'_{cp}\}$ are denoted as the agent's experiences at time slot t in the spectrum DDPG and computing DDPG, respectively. Similar to the DDPG-based algorithm, the HDDPG-based algorithm can be summarized in Algorithm 3, where $r_{eps} = \sum_{t=t_0}^{M_s+t_0} r_{sp}(t)$ and $r_{epc} = \sum_{t=t_0}^{M_s+t_0} r_{cp}(t)$ are the episode rewards for the spectrum DDPG and computing DDPG.

5.5 Simulation Results and Analysis

To demonstrate the performance of the proposed DRL-based resource management schemes for the vehicular scenarios with MeNB- and EN-mounted MEC servers, simulation results are presented in this section. The simulation procedures of an RL-based algorithm can be summarized into two stages, i.e., learning stage to learn the model and inferring stage to test the learned model [133]. In this section, we first learn the DDPG- and HDDPG-based models for the two considered scenarios. Then, the learned models are tested under the scenarios with different amounts of available resources to measure the performance of the proposed resource management schemes.

We consider a two-lane straight country road with one for each direction, where the traffic flow on the road is generated by PTV Vissim [134]. For the scenario with the MeNB-mounted MEC servers, one MeNB and two Wi-Fi APs (AP 1 and AP 2) are deployed on one side of the road to support vehicular applications. And for the scenario with the EN-mounted MEC servers, we assume the MEC server is placed at an EN, where two adjacent MeNBs ($S_j \in \mathcal{M}_1$ and $S_{j+1} \in \mathcal{M}_2$) and four Wi-Fi APs (AP 1 and AP 2 in $\mathcal{A}_j$, and AP 3 and AP 4 in $\mathcal{A}_{j+1}$) are wired connected to the EN. In the simulation, we assume a vehicle chooses to associate with a Wi-Fi AP when it is under the coverage of that Wi-Fi AP, otherwise associate with the MeNB. The transmit power of each vehicle is set as 1 watt (i.e., 30 dBm), and the uplink channel gain

/* Initialization phase */

Initialize the evaluation and target nets for the actor and critic in the spectrum DDPG with parameter matrices $\boldsymbol{\theta}_{sp}^{\mu}$, $\boldsymbol{\theta}_{sp}^{\mu'}$, $\boldsymbol{\theta}_{sp}^{Q}$, and $\boldsymbol{\theta}_{sp}^{Q'}$;

Initialize the evaluation and target nets for the actor and critic in the computing DDPG with parameter matrices $\boldsymbol{\theta}_{cp}^{\mu}$, $\boldsymbol{\theta}_{cp}^{\mu'}$, $\boldsymbol{\theta}_{cp}^{Q}$, and $\boldsymbol{\theta}_{cp}^{Q'}$;

Initialize the replay memory buffers for both spectrum DDPG and computing DDPG.

/* Parameter updating phase */

foreach *episode* **do**

  Receive initial observation state s_{sp}^{0}, and reset $r_{eps} = 0$ and $r_{epc} = 0$.

  foreach *step* t **do**

    Make an action $a_{sp}(t) = \mu_{sp}(s_{sp}(t))$ with the parameter matrix $\boldsymbol{\theta}_{sp}^{\mu}$, receive the subsequent state s'_{sp} and reward $r_{sp}(t)$, and obtain $\boldsymbol{T'} = \{T'_1(t), T'_2(t), \ldots, T'_{N(t)}(t)\}$;

    Make an action $a_{cp}(t) = \mu_{cp}(s_{cp}(t))$ with the parameter matrix $\boldsymbol{\theta}_{cp}^{\mu}$, and receive the subsequent state s'_{cp} and reward $r_{cp}(t)$;

    if *the number of experiences* $< M_r$ **then**

      Store $\{s_{sp}(t), a_{sp}(t), r_{sp}(t), s'_{sp}\}$ and $\{s_{cp}(t), a_{cp}(t), r_{cp}(t), s'_{cp}\}$ into the replay memory buffers of the spectrum DDPG and computing DDPG, respectively;

    else

      Replace the first saved experiences with $\{s_{sp}(t), a_{sp}(t), r_{sp}(t), s'_{sp}\}$ and $\{s_{cp}(t), a_{cp}(t), r_{cp}(t), s'_{cp}\}$ in the two replay memory buffers;

      Randomly select two mini batches of M_b experiences from the replay memory buffers of the spectrum DDPG and computing DDPG;

      Update the parameter matrices in the critic of the spectrum DDPG:

      $\boldsymbol{\theta}_{sp}^{Q} \leftarrow \boldsymbol{\theta}_{sp}^{Q} + \Delta\boldsymbol{\theta}_{sp}^{Q}$

      $\boldsymbol{\theta}_{sp}^{Q'} = \kappa_c \boldsymbol{\theta}_{sp}^{Q} + (1 - \kappa_c)\boldsymbol{\theta}_{sp}^{Q'}$;

      Update the parameter matrices in the actor of the spectrum DDPG:

      $\boldsymbol{\theta}_{sp}^{\mu} \leftarrow \boldsymbol{\theta}_{sp}^{\mu} + \Delta\boldsymbol{\theta}_{sp}^{\mu}$

      $\boldsymbol{\theta}_{sp}^{\mu'} = \kappa_a \boldsymbol{\theta}_{sp}^{\mu} + (1 - \kappa_a)\boldsymbol{\theta}_{sp}^{\mu'}$;

      Update the parameter matrices in the critic of the computing DDPG:

      $\boldsymbol{\theta}_{cp}^{Q} \leftarrow \boldsymbol{\theta}_{cp}^{Q} + \Delta\boldsymbol{\theta}_{cp}^{Q}$

      $\boldsymbol{\theta}_{cp}^{Q'} = \kappa_c \boldsymbol{\theta}_{cp}^{Q} + (1 - \kappa_c)\boldsymbol{\theta}_{cp}^{Q'}$;

      Update the parameter matrices in the actor of the computing DDPG:

      $\boldsymbol{\theta}_{cp}^{\mu} \leftarrow \boldsymbol{\theta}_{cp}^{\mu} + \Delta\boldsymbol{\theta}_{cp}^{\mu}$

      $\boldsymbol{\theta}_{cp}^{\mu'} = \kappa_a \boldsymbol{\theta}_{cp}^{\mu} + (1 - \kappa_a)\boldsymbol{\theta}_{cp}^{\mu'}$.

    end

    $r_{eps} = r_{eps} + r_{sp}(t)$;

    $r_{epc} = r_{epc} + r_{cp}(t)$.

  end

end

Algorithm 3: HDDPG-based algorithm

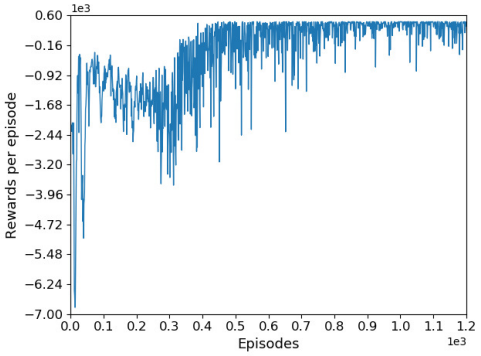
Table 5.1: Parameters for the learning stage.

Parameter	**Value**
Data size of a computing task	$[0.8, 1.2]$ kbits
Number of CPU cycles required to execute a computing task	$[80, 120]$ Mcycles/s
Amount of spectrum resources at an MeNB-/EN-mounted MEC server	$10/20$ MHz
Computational capability at an MeNB-/EN-mounted MEC server	$100/200$ GHz
Capacity of caching at an MeNB-/EN-mounted MEC server	$60/120$ kbits
Communication range of an MeNB/Wi-Fi AP	$600/150$ m
Straight-line distance between the MeNB (or Wi-Fi AP) and the road	225 m (or 3 m)
Service range of an EN-mounted MEC server	1100 m
Background noise power	-104 dBm
Discount factor on immediate reward	0.92
κ_a/κ_c	0.005
Replay memory size	10000
Size of a mini batch of experiences	32
Learning rate of the actor/critic	$0.00005/0.0005$

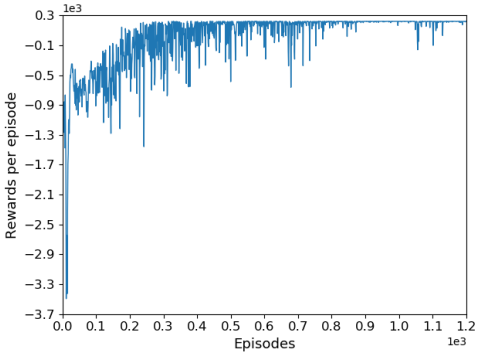
between a vehicle and an MeNB (or a Wi-Fi AP) is described as $L_m(d') = -30 - 35\log_{10}(\mathrm{d}')$ (or $L_w(d') = -40 - 35\log_{10}(\mathrm{d}')$) [126], where d' is the distance between the vehicle user and the MeNB (or the Wi-Fi AP). Taking a type of delay-sensitive computing task (e.g., the analysis of the surveillance content of special road segments for approaching vehicles [46, 135]) as an example, the delay bound for the offloaded computing task is set as 50 ms, i.e., $d_k = 50$ ms for $k \in \mathcal{N}$. Other parameters for the learning stage are listed in Table 5.1. Unless specified otherwise, parameters for the inferring stage are same to that of the learning stage.

Figures 5.4 and 5.5 demonstrate the convergence performance of the DDPG- and HDDPG-based algorithms in the scenarios with MeNB- and EN-mounted MEC servers, respectively. As we can see from the six subfigures, among the $1,200$ episodes during the learning stage, the total rewards per episode fluctuate sharply and are relatively small in the first few hundreds episodes and then tend to a relatively stable and high value. As mentioned in Algorithms 2 and 3, all parameters of the actors and critics of the DDPG- and HDDPG-based algorithms are initialized by the TensorFlow. Once $10,000$ of experiences are saved in the replay memory buffers, the learning stages of the DDPG- or HDDPG-based algorithms start to update the parameters of the actor and critic. Thus, the total rewards per episode fluctuate sharply in the beginning of the learning stage and then increase with the parameters gradually being optimized. Moreover, to direct the agent to satisfy more tasks' QoS requirements, we clip $r_k^d(t)$ to be $[-8, 0.2]$ and

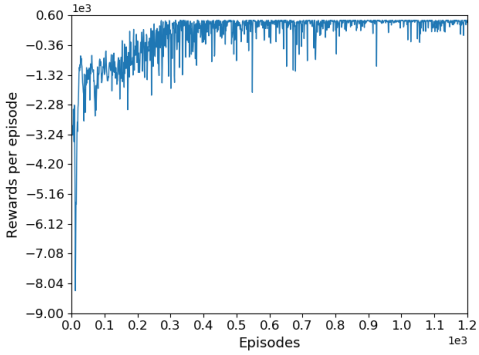
$r_s^d(t)$ to be $[-7, 0.2]$. And the average number of steps in each episode is 1100. Hence, the maximum total rewards per episode achieved by the DDPG-based algorithm and the computing DDPG of the HDDPG-based algorithm is 440, as shown in Figures 5.4(a), 5.4(c), 5.5(a), and 5.5(c). Similarly, the maximum total rewards per episode achieved by the spectrum DDPG of the HDDPG-based algorithm is 220, as shown in Figures 5.4(b) and 5.5(b) is 220.



(a) DDPG-based algorithm



(b) Spectrum DDPG in HDDPG-based algorithm

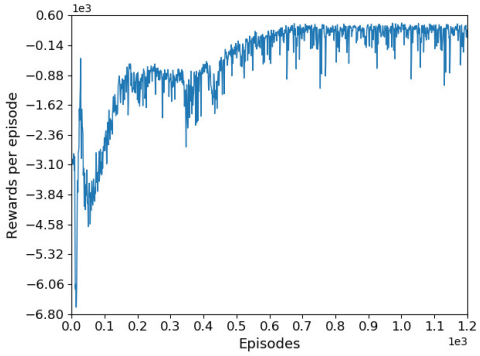


(c) Computing DDPG in HDDPG-based algorithm

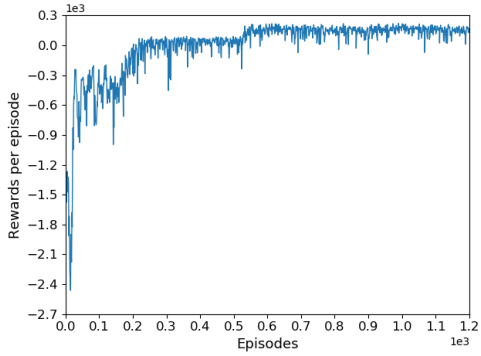
Figure 5.4: The total rewards of each episode in scenario with an MeNB-mounted MEC server.

As the outputs of the spectrum DDPG are parts of the inputs of the computing DDPG in the HDDPG-based algorithm, we regard the convergence performance of the computing DDPG as that of the HDDPG-based algorithm. From Figures 5.4(a) and 5.4(c) (or Figures 5.5(a) and 5.5(c)), the HDDPG-based algorithm converges within a smaller number of episodes than the DDPG-based algorithm in the scenario with an MeNB-mounted MEC server (or with an EN-mounted MEC server). For example, in the scenario with an MeNB-mounted MEC server, the

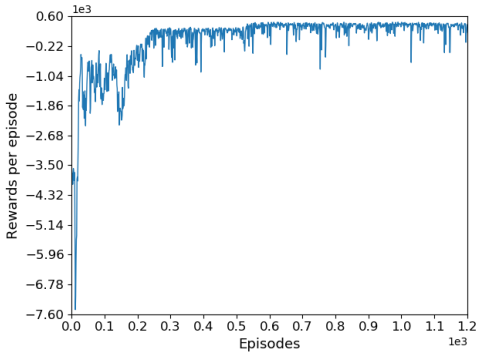
DDPG-based algorithm starts to converge after around 400 episodes and that of the HDDPG-based algorithm is around 300 episodes. That is because the impacts of the spectrum slicing and spectrum allocation on the final rewards are learned by the spectrum DDPG in the HDDPG-based algorithm. However, since two DDPG models need to be trained, the training time of 1200 episodes for the HDDPG-based algorithm is physically longer than that for the DDPG-based algorithm. Moreover, Figures 5.4 and 5.5 indicate that more episodes are required for the proposed algorithms to converge in the scenario with an EN-mounted MEC server comparing to the scenario with an MeNB-mounted MEC server. This is due to the more complex MDP caused by the increased numbers of vehicles and BSs under the EN-mounted MEC server.



(a) DDPG-based algorithm



(b) Spectrum DDPG in HDDPG-based algorithm



(c) Computing DDPG in HDDPG-based algorithm

Figure 5.5: The total rewards of each episode in scenario with an EN-mounted MEC server.

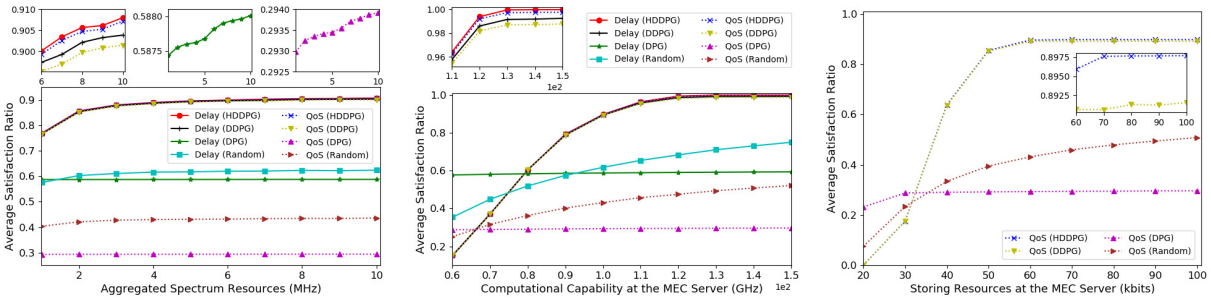
From the MEC server perspective, an efficient resource management scheme should be able to serve as many users as possible with the given available resources. And for a vehicle

user, satisfied delay or QoS requirements are critical. Thus, to measure the performance of the two proposed DRL-based resource management schemes, we define delay/QoS satisfaction ratio as the number of vehicles with satisfied delay/QoS requirements over the total number of vehicles within the service area and use them as the evaluation criterion in the inferring stage. Two comparisons are considered, the DPG-based scheme and the random resource management scheme. Considering the relatively large number of vehicles under the service area and the large size of a resource allocation action, it is unbefitting to discretize each action. Thus, we choose the DPG-based scheme, which is also applicable to MDPs with continuous state and action spaces as one of our comparisons. The other comparison is the random resource management scheme[6], which can obtain resource allocation decisions rapidly, same as our proposed schemes.

Figure 5.6 demonstrates the average delay/QoS satisfaction ratios over 5000 adjacent environment states in the scenario with the MeNB-mounted MEC server, with respect to different amounts of aggregated spectrum resources, computation capabilities, and capacity of caching, respectively. The three subfigures show that, except the cases with short supply available resources, higher average delay satisfaction ratios and doubled average QoS satisfaction ratios are achieved by the proposed DDPG- and HDDPG-based schemes compared with the DPG-based scheme[7] and the random resource management scheme. With the increase of the amount of the available spectrum/computing/caching resources, the average satisfaction ratios achieved by the proposed schemes and the two comparisons increase due to the increase of the amount of resources allocated to vehicle users. As mentioned before, the delay requirement is a part of its QoS requirement for an offloaded task, i.e., for a task with satisfied delay requirement, its QoS requirement would not be satisfied if no enough caching resources allocated to it. And the delay satisfaction of an offloaded task is co-determined by the amounts of spectrum and computing resources allocated to it. Thus, with the increase of the amount of spectrum resources (or computing/caching), the delay/QoS satisfaction ratios achieved by the proposed schemes tend to be saturated due to the fixed amounts of the computing and caching resources (or the spectrum and caching/computing resources). Moreover, the gaps between the delay satisfaction ratios and QoS satisfaction ratios of the proposed schemes are much smaller than that of the two comparisons. That is because the proposed schemes have overall managed the spectrum/computing/caching resources to satisfy the QoS requirements for as many vehicles' tasks as possible.

[6]With the random resource management scheme, the MEC server randomly allocates the spectrum, computing, and caching resources among vehicles under its service area.

[7]The DPG-based algorithm is trained with the same parameter setting as the two proposed schemes. Without deep learning, the convergence performance of the DPG-based algorithm cannot be guaranteed, especially in a scenario with large sizes of environment state and action vectors. Thus, low delay/QoS satisfaction ratios are obtained by the DPG-based scheme.



(a) Versus the aggregate spectrum resources

(b) Versus the computation capabilities

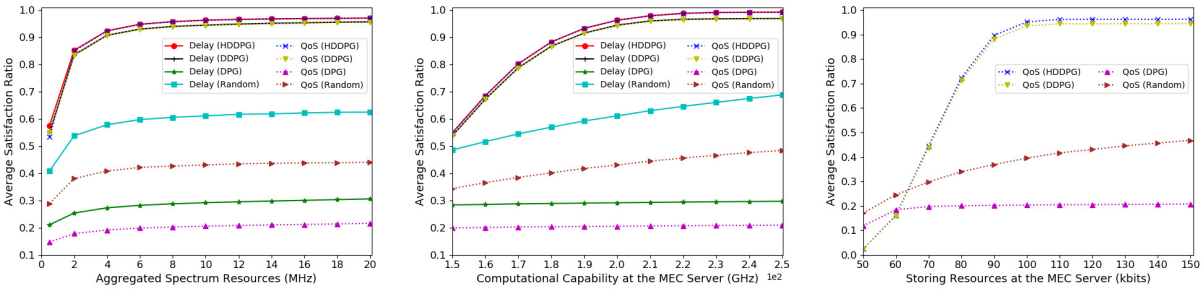
(c) Versus the capacity of caching

Figure 5.6: Average delay/QoS satisfaction ratio over the vehicles under the service area of the MeNB-mounted MEC server.

Figure 5.7 shows the average delay/QoS satisfaction ratios in the scenario with the EN-mounted MEC server. Similar to the scenario with the MeNB-mounted MEC server, the subfigures show that higher delay/QoS satisfaction ratios can be obtained by the proposed resource management schemes compared with the DPG-based scheme and random resource management scheme. Except for the scenarios with short supply available resources, such as the scenarios with $2\,\text{MHz}$ or less spectrum resources available for the EN-mounted MEC server with 100 or more vehicles under its service area, over 90% of tasks are with satisfied QoS requirements under different environment states with the proposed schemes. Also, from Figure 5.7 and the zoom in figures of Figure 5.6, the HDDPG-based scheme outperforms the DDPG-based scheme in terms of the delay/QoS satisfaction ratios in addition to improved convergence. That is because two DDPG models are adopted to optimally manage the spectrum and computing/caching resources in the HDDPG-based scheme. Moreover, as the HDDPG-based algorithm is more applicable to the scenario with complex MDPs, compared to the scenario with the MeNB-mounted MEC server, more performance enhancements are achieved by the HDDPG-based algorithm than the DDPG-based algorithm in the scenario with the EN-mounted MEC server.

5.6 Summary

In this chapter, we have investigated the joint spectrum, computing, and caching resource management problem to accommodate delay-sensitive applications in the MVNET. Particularly, we have considered two typical MEC architectures, i.e., with MeNB- and EN-mounted MEC servers, under which two resource optimization problems have been formulated to maximize the



(a) Versus the aggregate spectrum resources

(b) Versus the computation capabilities

(c) Versus the capacity of caching

Figure 5.7: Average delay/QoS satisfaction ratio over vehicles under the service area of the EN-mounted MEC server.

number of offloaded tasks that are completed with satisfied QoS requirements. As the formulated problems are computationally intractable in real time, we have exploited the DRL to transform and solve them and devised the DDPG- and HDDPG-based algorithms. Extensive simulation results have shown that our proposed DDPG- and HDDPG-based resource management schemes can converge within acceptable training episodes and outperform the DPG-based scheme and random resource management scheme in terms of delay/QoS satisfaction ratios. In the next chapter, we will consider UAV-mounted MEC servers to address the issues caused by uneven resource demanding in MVNETs.

Chapter 6

Multi-Dimensional Resource Management in UAV-Assisted MVNETs

In this chapter, we investigate multi-dimensional resource management for UAV-assisted MVNETs. To efficiently provide on-demand resource access, the MeNB and UAV, both mounted with MEC servers, cooperatively make association decisions and allocate proper amounts of resources to vehicles. Since there is no central controller, we formulate the resource allocation at the MEC servers as a distributive optimization problem to maximize the number of offloaded tasks while satisfying their heterogeneous QoS requirements, and then solve it with an MADDPG-based method. Through centrally training the MADDPG model offline, the MEC servers, acting as learning agents, then can rapidly make vehicle association and resource allocation decisions during the online execution stage. From our simulation results, the MADDPG-based method can converge within 200 training episodes, comparable to the single-agent DDPG (SADDPG)-based one. Moreover, the proposed MADDPG-based resource management scheme can achieve higher delay/QoS satisfaction ratios than the SADDPG-based and random schemes. The complete research results of this chapter can be seen in [69].

6.1 Background Information

To implement MEC- and/or UAV-assisted vehicular networks, many efforts have been made recently. Some of them have been focused on the deployment of MEC-mounted UAVs. For example, [136] has studied how to deploy and schedule the MEC-mounted UAVs to support vehicular applications. Resource management, as another research emphasis, has also attracted

lots of attention from the existing works, where most of them adopt the optimization and RL methods. In [58], the transmit powers of vehicles and the trajectories of UAVs have been jointly optimized to maximize the resource efficiency on MEC-mounted UAVs. In [137], a DRL based adaptive computation offloading method has been proposed to balance the tradeoff between energy consumption and data transmission delay in an MEC-assisted vehicular network. In [138], a framework using MEC-mounted UAVs has been proposed to support mobile users in the extended 5G network, and an RL method is adopted to manage the resources carried by the UAV. To jointly manage the spectrum, computing, and caching resources available to an MEC-mounted BS, DDPG- and HDDPG-based schemes have been proposed in chapter 5. However, only vehicular networks supported either by MEC-mounted BSs or UAVs have been studied by most of the existing works. How to perform efficient resource allocation to support applications with various resource demand and heterogeneous QoS requirements in vehicular networks with MEC-mounted BSs and UAVs still needs effort.

In this chapter, we investigate multi-dimensional resource management in the UAV-assisted MVNETs, where MEC servers are mounted at an MeNB and in some UAVs to provide resource access to vehicles with limited on-board resources. Inspired by existing works [137–140], we adopt RL methods to achieve real-time resource management in the considered scenario. Considering the sensitive delay requirements of some vehicular applications (e.g., autonomous driving) and the wireless transmission time between a UAV and a controller, installing a central controller either at the MeNB or an edge node to enable a centralized resource management scheme is infeasible sometimes. Thus, we develop a distributed cooperative scheme based on a multi-agent RL method to manage the multi-dimensional resources available to the MEC-mounted MeNB and UAVs. The main contributions of this work are summarized as follows,

1. To support as many offloaded tasks as possible while satisfying their QoS requirements, we formulate an individual optimization problem to each MEC server to jointly manage the MEC-mounted MeNB's and UAVs' spectrum, computing, and caching resources;

2. Because of the vehicle association pattern variables, the formulated problems are coupled with each other and non-convex. To rapidly solve these problems to satisfy the sensitive delay requirements of the offloaded tasks, we transform each formulated problem according to the main idea of RL;

3. We convert the transformed problems as a multi-agent problem by letting each MEC server act as an agent and develop an MADDPG algorithm to solve it. Through training the MADDPG model offline, the vehicle association and resource allocation decisions can be made in real time by each MEC server.

The rest of this chapter is organized as follows. In Section 6.2, the UAV-assisted MVNET architecture and the multi-dimensional resource management model are presented, followed with the formulated optimization problems. We develop an MADDPG algorithm to solve the formulated problems in Section 6.3 and provide simulation results in Section 6.4 to validate the performance of the MADDPG algorithm. This work is concluded in Section 6.5.

6.2 System Model and Problem Formulation

In this section, we first introduce a UAV-assisted MVNET architecture and a resource management model, and then formulate optimization problems to manage the multi-dimensional resources available to the MEC servers.

6.2.1 UAV-Assisted MVNET

Consider a UAV-assisted MVNET with MEC servers mounted at an MeNB and in multiple UAVs to support heterogeneous delay-sensitive vehicular applications, as illustrated in Figure 6.1. Each UAV flies at a constant speed under the coverage area of the MeNB and cooperatively provides resource access to vehicles. Vehicles drive either in cooperative states, such as in convoy and platoon forms, or in non-cooperative states [40]. Each vehicle periodically generates computing tasks with different QoS requirements and computing/caching resource demands. If demands to offload its task to the MEC server, the vehicle first sends a resource access request to the MeNB and/or a UAV covering it. After receiving the access permission and resource allocation results from the corresponding MEC servers, the computing task will be offloaded to the associated MEC server over the allocated spectrum resources. As task division is not considered here, we assume a vehicle under the overlapping area between the MEC-mounted MeNB and UAV can only associate with and offload its task to one of the MEC servers.

6.2.2 Resource Management Model

Due to the diversified applications and high vehicle mobility, the vehicular network topology and the distribution of resource access requests change over time frequently, thereby resulting in time-varying resource demand from vehicles under the service area of the MeNB. To allocate proper amounts of spectrum, computing, and caching resources to each resource access request to satisfy the offloaded tasks' QoS requirements, a multi-dimensional resource management model is developed in this section.

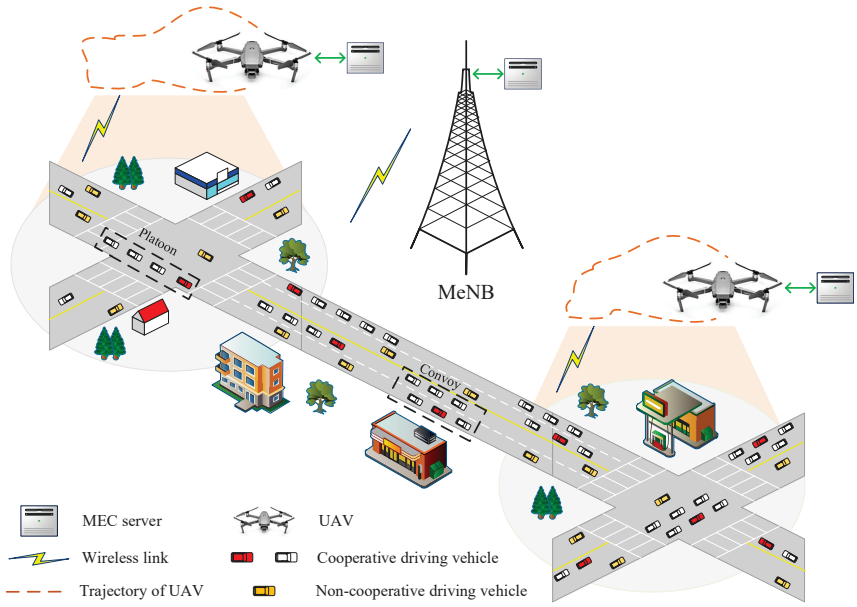


Figure 6.1: An illustration of the UAV-assisted MVNET.

Consider a two-lane and two-way country road segment, shown in Figure 6.2, an MEC-mounted MeNB is placed on one side of the road to provide full signal coverage to vehicles, and U MEC-mounted UAVs, denoted as $\mathcal{U}$, are flying above the road and under the coverage of the MeNB. Each UAV covers parts of the considered road segment and flies at a constant speed with no overlapping with other UAVs. Note that, the UAVs' flight trajectories are fixed in this work. The reason is, the range that each UAV is allowed to fly to is relatively small and the resource demanding from vehicles remains relatively stable within a small range, the impacts of trajectory design on the resource utilization improvement are small here.

Let $\mathcal{N}(t) = \{1, 2, \cdot, i, \cdot, N(t)\}$[1] be the set of vehicles within the coverage of the MeNB at time slot t, where $N(t) = |\mathcal{N}(t)|$. The computing task generated by vehicle i at time slot t is described as $\{c_i^s(t), c_i^c(t), c_i^d(t)\}$, where $c_i^s(t)$, $c_i^c(t)$, and $c_i^d(t)$ are the data size of, the number of CPU cycles required to execute, and the maximum delay tolerated by vehicle i's task, respectively. Once a computing task is generated, vehicle i sends a resource access request, containing the detailed information about this task, i.e., $\{c_i^s(t), c_i^c(t), c_i^d(t)\}$, and the driving state, e.g., moving direction and current position, to the MEC servers on demand. According to the collected information, each MEC server then makes vehicle association and resource allocation

[1]In this chapter, we add (t) at the end of some notations to distinguish the fixed parameters from the time-varying parameters. Yet the vehicular network is assumed to be static during each time slot [58].

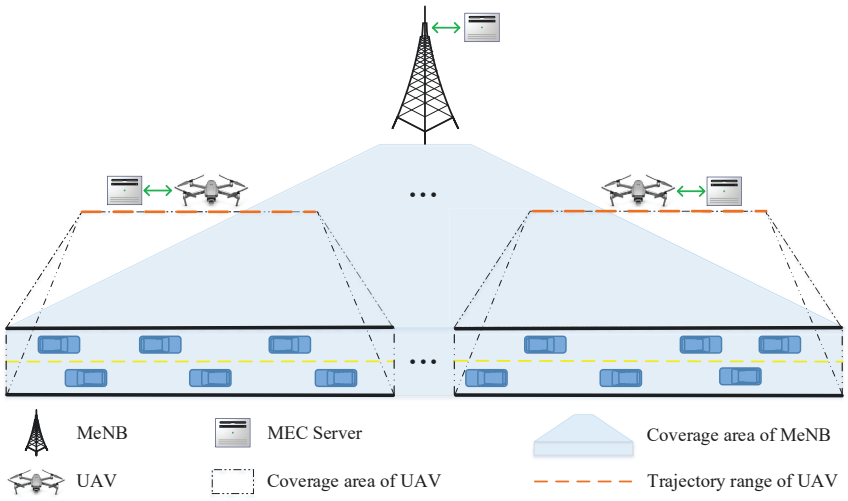


Figure 6.2: A simplified UAV-assisted MVNET scenario for multi-dimensional resource management.

decisions and returns them to vehicles, where each decision contains the vehicle-MeNB or vehicle-UAV association patterns and the fractions of the spectrum, computing, and caching resources allocated to vehicles.

Spectrum resource management: Denote the resource matrix for the MEC-mounted MeNB by $\{S_m, C_m^{co}, C_m^{ca}\}$, where S_m, C_m^{co}, and C_m^{ca} are the amounts of available spectrum, computing, and caching resources, respectively. Likewise, for MEC-mounted UAV $j \in \mathcal{U}$, the resource matrix is given by $\{S_u, C_u^{co}, C_u^{ca}\}$. To offload the computing tasks from each vehicle to its associated MEC server with an acceptable transmission delay, a proper amount of spectrum should be allocated to each vehicle. Let $\mathcal{N}'(t)/N'(t)$ be the set/number of vehicles under the coverage of the MeNB while outside of the U UAVs at time slot t, and $\mathcal{N}_j(t)/N_j(t)$ ($j \in \mathcal{U}$) is the set/number of vehicles under the coverage of UAV j. Namely, we have $\mathcal{N}(t) = \mathcal{N}'(t) \cup \{\mathcal{N}_j(t) : j \in \mathcal{U}\}$. For vehicle $i \in \mathcal{N}_j(t)$, let binary variables $b_{i,m}(t)$ and $b_{i,j}(t)$ be the vehicle-MeNB and vehicle-UAV association patterns, where $b_{i,m}(t) = 1$ (or $b_{i,j}(t) = 1$) if vehicle i associates with the MeNB (or UAV j) at time slot t, and $b_{i,m}(t) = 0$ (or $b_{i,j}(t) = 0$) otherwise. As the task division technology is not adopted, we have $b_{i,m}(t) + b_{i,j}(t) = 1$ for vehicles in $\mathcal{N}_j(t)$. Note that for vehicle $i \in \mathcal{N}'(t)$, $b_{i,j}(t)$ is null and we set $b_{i,m}(t) = 1$ since it is outside of the coverage of any UAV.

1) Vehicles associated with the MeNB: Let $\mathcal{N}_m(t)$ denote the set of vehicles associated with the MeNB, i.e., vehicles with $b_{i,m}(t) = 1$. Then, all uplink transmissions for offloading tasks from vehicles in $\mathcal{N}_m(t)$ to the MeNB share spectrum resource S_m. Considering the

duality transmission over the same coherence interval, the channels for uplink and downlink transmissions between a vehicle and the MeNB/UAV are assumed to be symmetry [67]. Let $G_{i,m}(t)$ denote the average channel gain between vehicle $i \in \mathcal{N}_m(t)$ and the MeNB at time slot t, which varies dynamically over the vehicle-MeNB distance. Then, we can express the achieved spectrum efficiency at the MeNB from vehicle i as

$$e_{i,m}(t) = \log_2 \left(1 + \frac{PG_{i,m}(t)}{\sigma^2}\right), \tag{6.1}$$

where P is the vehicle's transmit power and σ^2 is the power of the white noise. Denote $f_{i,m}(t)$ as the fraction of spectrum allocated to vehicle i by the MeNB at time slot t. Namely, vehicle i can occupy spectrum resource $S_m f_{i,m}(t)$ to offload its task to the MeNB at time slot t. The corresponding uplink transmission rate then can be given by

$$R_{i,m}(t) = S_m f_{i,m}(t) e_{i,m}(t). \tag{6.2}$$

2) Vehicles associated with UAVs: Considering the limitation of spectrum resources, we adopt spectrum reusing among UAVs with acceptable interference by pre-designing the fly trajectory of each UAV. Specifically, the uplink transmissions from vehicles to the U UAVs reuse spectrum resource S_u. As each UAV flies at a constant speed above the road, we assume there always exists a line-of-sight connection between a vehicle and a UAV. And the average channel gain between vehicle i and UAV $j \in \mathcal{U}$ at time slot t, $G_{i,j}(t)$, is defined similar to [141]. In addition to the white noise, UAV j also experiences the interference from the uplink transmission to UAV $v \in \mathcal{U} \backslash \{j\}$. Thus, the corresponding spectrum efficiency achieved at UAV j can be given by

$$e_{i,j}(t) = \log_2 \left(1 + \frac{PG_{i,j}(t)}{\sum_{v \in \mathcal{U} \backslash \{j\}} PG_{i,v}(t) + \sigma^2}\right). \tag{6.3}$$

Let $f_{i,j}(t)$ be the fraction of spectrum allocated to vehicle i by UAV j at time slot t. Then we can express the uplink transmission rate from vehicle i to UAV j as

$$R_{i,j}(t) = S_u f_{i,j}(t) e_{i,j}(t). \tag{6.4}$$

Computing/Caching resource management: As mentioned above, in addition to the vehicle-MeNB or vehicle-UAV association patterns and the spectrum allocation results, each MEC server also needs to allocate proper amounts of computing and caching resources to vehicles. As the transmit powers of the MeNB and UAVs are much higher than that of a vehicle and the data size of each task's process result is relatively small [142], the time consumption on

downlinking the task process result to each vehicle is neglected here [58]. Let $f_{i,m}^{co}(t)$ (or $f_{i,j}^{co}(t)$) be the fraction of computing resources allocated to vehicle i's task from the MeNB (or from UAV $j \in \mathcal{U}$) at time slot t. Then, the completion time of vehicle i's computing task, i.e., the time duration from the task is generated until vehicle i receives the task process result from its associated MEC server, can be expressed as

$$T_i(t) = \begin{cases} \frac{c_i^s(t)}{R_{i,m}(t)} + \frac{c_i^c(t)}{C_m^{co} f_{i,m}^{co}(t)}, & \text{if } b_{i,m}(t) = 1 \\ \frac{c_i^s(t)}{R_{i,j}(t)} + \frac{c_i^c(t)}{C_u^{co} f_{i,j}^{co}(t)}, & \text{if } b_{i,j}(t) = 1. \end{cases} \tag{6.5}$$

A proper amount of caching resource has to be pre-allocated to each resource access request, such that all data related to this task can be cached for task processing. For a resource access request sent by vehicle i, let $f_{i,m}^{ca}(t)$ (or $f_{i,j}^{ca}(t)$) be the fraction of caching resources allocated by the MeNB (or by UAV j). Then, a task offloaded from vehicle i to an MEC server is regarded to be completed with satisfied QoS requirements, if i) at least $c_i^s(t)$ caching resource is allocated to cache vehicle i's task, i.e., $f_{i,m}^{ca}(t) C_m^{ca} \geq c_i^s(t)$ or $f_{i,j}^{ca}(t) C_u^{ca} \geq c_i^s(t)$, and ii) the task's delay requirement is satisfied, namely, the task completion time is less than its maximum tolerance delay, $T_i(t) \leq c_i^d(t)$.

6.2.3 Problem Formulation

As resources available to both MeNB and UAV are finite, it is critical to efficiently allocate them. To avoid time consumption on exchanging information between a central controller and the MEC servers, a distributed resource management scheme is considered here. We formulate individual optimization problems to the MEC-mounted MeNB and UAVs to distributively manage their available spectrum, computing, and caching resources.

According to the resource management model presented in the previous subsection, the optimization problems formulated for the MeNB and UAVs can be described as

$$\max_{\substack{\mathbf{b}_m(t), \mathbf{f}_m(t), \\ \mathbf{f}_m^{co}(t), \mathbf{f}_m^{ca}(t)}} \sum_{i \in \mathcal{N}(t)} b_{i,m}(t) H[c_i^d(t) - T_i(t)] H[f_{i,m}^{ca}(t) C_m^{ca} - c_i^s(t)]$$

$$\text{s.t.} \begin{cases} (6.1), (6.2), (6.5) & (6.6a) \\ b_{i,m}(t) \in \{0,1\} \qquad i \in \mathcal{N}(t) & (6.6b) \\ b_{i,m}(t) + b_{i,j}(t) = 1 \qquad \forall i \in \cup_{j \in \mathcal{U}} \mathcal{N}_j(t) & (6.6c) \\ f_{i,m}(t), f_{i,m}^{co}(t), f_{i,m}^{ca}(t) \in [0,1], \quad i \in \mathcal{N}(t) & (6.6d) \\ \sum\limits_{i \in \mathcal{N}(t)} b_{i,m}(t) f_{i,m}(t) = 1 & (6.6e) \\ \sum\limits_{i \in \mathcal{N}(t)} b_{i,m}(t) f_{i,m}^{co}(t) = 1 & (6.6f) \\ \sum\limits_{i \in \mathcal{N}(t)} b_{i,m}(t) f_{i,m}^{ca}(t) = 1 & (6.6g) \end{cases}$$

and

$$\max_{\substack{\mathbf{b}_j(t), \mathbf{f}_j(t), \\ \mathbf{f}_j^{co}(t), \mathbf{f}_j^{ca}(t)}} \sum_{i \in \mathcal{N}_j(t)} b_{i,j}(t) H[c_i^d(t) - T_i(t)] H[f_{i,j}^{ca}(t) C_u^{ca} - c_i^s(t)]$$

$$\text{s.t.} \begin{cases} (6.3), (6.4), (6.5) & (6.7a) \\ b_{i,j}(t) \in \{0,1\} \qquad i \in \mathcal{N}_j(t) & (6.7b) \\ b_{i,j}(t) + b_{i,m}(t) = 1 \qquad \forall i \in \mathcal{N}_j(t) & (6.7c) \\ f_{i,j}(t), f_{i,j}^{co}(t), f_{i,j}^{ca}(t) \in [0,1], \quad i \in \mathcal{N}_j(t) & (6.7d) \\ \sum\limits_{i \in \mathcal{N}_j(t)} b_{i,j}(t) f_{i,j}(t) = 1 & (6.7e) \\ \sum\limits_{i \in \mathcal{N}_j(t)} b_{i,j}(t) f_{i,j}^{co}(t) = 1 & (6.7f) \\ \sum\limits_{i \in \mathcal{N}_j(t)} b_{i,j}(t) f_{i,j}^{ca}(t) = 1, & (6.7g) \end{cases}$$

respectively, where $\mathbf{b}_m(t) = \{b_{i,m}(t) : i \in \mathcal{N}(t)\}$ and $\mathbf{b}_j(t) = \{b_{i,j}(t) : i \in \mathcal{N}_j(t)\}$ are the vehicle association pattern matrices between vehicle $i \in \mathcal{N}(t)$ and the MeNB and between vehicle $i \in \mathcal{N}_j(t)$ and UAV $j \in \mathcal{U}$, respectively. As vehicles send their resource access requests to both MeNB and UAV, yet their computing tasks can be only offloaded to one of them, constraint $b_{i,j}(t) + b_{i,m}(t) = 1$ is considered in both formulated problems. $\mathbf{f}_m(t)$, $\mathbf{f}_m^{co}(t)$, and $\mathbf{f}_m^{ca}(t)$ (or $\mathbf{f}_j(t)$, $\mathbf{f}_j^{co}(t)$, and $\mathbf{f}_j^{ca}(t)$) are the spectrum, computing, and caching resource allocation matrices among vehicles associated with the MeNB (or with UAV j), respectively. The objective function of both formulated problems is to maximize the number of offloaded tasks completed

by the MEC server with satisfied QoS requirements, where the Heaviside step function, $H(\cdot)$, indicates whether the offloaded task's QoS requirements are satisfied.

6.3 MADDPG-based Resource Management Scheme

It is difficult to rapidly solve the above optimization problems using the traditional methods due to the following reasons,

1. The two problems are mixed-integer programming problems.
2. For each problem, spectrum resource management is coupled with computing resource management.
3. As indicated by constraints (6.6c) and (6.7c), the problem formulated for the MeNB is coupled with those for the UAVs in $\mathcal{U}$.
4. Considering the high network dynamic caused by the mobility of vehicles and UAVs and the sensitive delay requirements of different vehicular applications, each formulated problem has to be solved rapidly, and which is impossible to be achieved by the traditional optimization methods.

Thus, an RL approach is leveraged here. Considering the coupled relation among the formulated problems and there is no central controller, a multi-agent RL algorithm is designed, where each MEC server acts as an agent to learn the resource management scheme and solve the corresponding formulated problem. Specifically, we first re-model the resource management problems targeting the MeNB and the U UAVs as a multi-agent extension of MDPs [127, 140, 143], and then design an MADDPG algorithm to solve the MDPs.

6.3.1 Problem Transformation

We transform the formulated problems into a partially observable Markov game for $U + 1$ agents, including the MeNB agent and U UAV agents. Define the Markov game for $U + 1$ agents as a set of states $\mathcal{S}$, a set of observations $\mathcal{O} = \{\mathcal{O}_m, \mathcal{O}_1, \ldots, \mathcal{O}_j, \ldots, \mathcal{O}_U\}$, and a set of actions $\mathcal{A} = \{\mathcal{A}_m, \mathcal{A}_1, \ldots, \mathcal{A}_j, \ldots, \mathcal{A}_U\}$. The state set $\mathcal{S}$ describes the possible configurations of the road segment under the coverage of the MEC-mounted MeNB, including the mobility characteristics of vehicles and UAVs, and the time-varying tasks generated by vehicles. $\mathcal{O}_m$ and

$\mathcal{O}_j$ are observation spaces for the MeNB agent and UAV j agent ($j \in \mathcal{U}$), respectively, and the observation of each agent at time slot t is a part of the current state, $s(t) \in \mathcal{S}$. $\mathcal{A}_m$ and $\mathcal{A}_j$ ($j \in \mathcal{U}$) are action spaces for the MeNB and UAV j. For each given state $s \in \mathcal{S}$, the MeNB agent and UAV j agent use the policies, $\boldsymbol{\pi}_m$: $\mathcal{S} \mapsto \mathcal{A}_m$ and $\boldsymbol{\pi}_j$: $\mathcal{S} \mapsto \mathcal{A}_j$, to choose an action from their action spaces according to their observations corresponding to s, respectively.

Environment state: Let $x_i(t)$ and $y_i(t)$ be the x- and y- coordinates of vehicle $i \in \mathcal{N}(t)$, and $x'_j(t)$, $y'_j(t)$, and $z'_j(t)$ denote the x-, y-, and z- coordinates of UAV j at time slot t. Then, according to the resource management problems formulated for the UAV-assisted MVNET, the environment state at time slot t, $s(t) \in \mathcal{S}$, can be given by

$$\begin{aligned} s(t) =&\{x_1(t), x_2(t), \ldots, x_{N(t)}(t), y_1(t), y_2(t), \ldots, y_{N(t)}(t), c^s_1(t), c^s_2(t), \ldots, c^s_{N(t)}(t), \\ &c^c_1(t), c^c_2(t), \ldots, c^c_{N(t)}(t), c^d_1(t), c^d_2(t), \ldots, c^d_{N(t)}(t), x'_1(t), x'_2(t), \ldots, x'_U(t), y'_1(t), \\ &y'_2(t), \ldots, y'_U(t), z'_1(t), z'_2(t), \ldots, z'_U(t)\}. \end{aligned} \tag{6.8}$$

Observation: As the considered road segment is under the coverage of the MeNB and no information exchanging among different MEC servers, the observations of the MeNB and UAV j at time slot t, i.e., $o_m(t) \in \mathcal{O}_m$ and $o_j(t) \in \mathcal{O}_j$, can be described as

$$\begin{aligned} o_m(t) =&\{x_1(t), x_2(t), \ldots, x_{N(t)}(t), y_1(t), y_2(t), \ldots, y_{N(t)}(t), c^s_1(t), c^s_2(t), \ldots, \\ &c^s_{N(t)}(t), c^c_1(t), c^c_2(t), \ldots, c^c_{N(t)}(t), c^d_1(t), c^d_2(t), \ldots, c^d_{N(t)}(t)\} \end{aligned} \tag{6.9}$$

and

$$\begin{aligned} o_j(t) =&\{x_{1,j}(t), x_{2,j}(t), \ldots, x_{N_j(t),j}(t), y_{1,j}(t), y_{2,j}(t), \ldots, y_{N_j(t),j}(t), c^s_{1,j}(t), \\ &c^s_{2,j}(t), \ldots, c^s_{N_j(t),j}(t), c^c_{1,j}(t), c^c_{2,j}(t), \ldots, c^c_{N_j(t),j}(t), c^d_{1,j}(t), c^d_{2,j}(t), \ldots, \\ &c^d_{N_j(t),j}(t), x'_j(t), y'_j(t), z'_j(t)\}, j \in \mathcal{U}, \end{aligned} \tag{6.10}$$

respectively. For vehicle $i \in \mathcal{N}_j(t)$ under the coverage of UAV j, $x_{i,j}(t)$ and $y_{i,j}(t)$ denote the x- and y- coordinates, and $c^s_{i,j}(t)$, $c^c_{i,j}(t)$, and $c^d_{i,j}(t)$ are the detailed information about the offloaded computing tasks. Note that, for vehicle i under the overlapping area between the MeNB and UAV j, we have $\{x_{i,j}(t), y_{i,j}(t), z_{i,j}(t)\} = \{x_i(t), y_i(t), z_i(t)\}$ and $\{c^s_{i,j}(t), c^c_{i,j}(t), c^d_{i,j}(t)\} = \{c^s_i(t), c^c_i(t), c^d_i(t)\}$.

Action: According to the current policy, $\boldsymbol{\pi}_m$ or $\boldsymbol{\pi}_j$, and the corresponding observation, each MEC server chooses an action from its action space. The actions of the MeNB and UAV j at

time slot t, i.e., $a_m(t) \in \mathcal{A}_m$ and $a_j(t) \in \mathcal{A}_j$ ($j \in \mathcal{U}$), can be described as

$$\begin{aligned} a_m(t) =&\{b'_{1,m}(t), b'_{2,m}(t), \ldots, b'_{N(t),m}(t), f_{1,m}(t), f_{2,m}(t), \ldots, f_{N(t),m}(t), f^{co}_{1,m}(t), \\ &f^{co}_{2,m}(t), \ldots, f^{co}_{N(t),m}(t), f^{ca}_{1,m}(t), f^{ca}_{2,m}(t), \ldots, f^{ca}_{N(t),m}(t)\} \end{aligned} \tag{6.11}$$

and

$$\begin{aligned} a_j(t) =&\{b'_{1,j}(t), b'_{2,j}(t), \ldots, b'_{N_j(t),j}(t), f_{1,j}(t), f_{2,j}(t), \ldots, f_{N_j(t),j}(t), f^{co}_{1,j}(t), f^{co}_{2,j}(t), \\ &\ldots, f^{co}_{N_j(t),j}(t), f^{ca}_{1,j}(t), f^{ca}_{2,j}(t), \ldots, f^{ca}_{N_j(t),j}(t)\}, j \in \mathcal{U}, \end{aligned} \tag{6.12}$$

respectively, where the value ranges of $f_{i,m}(t)$, $f^{co}_{i,m}(t)$, $f^{ca}_{i,m}(t)$, $f_{i,j}(t)$, $f^{co}_{i,j}(t)$, and $f^{ca}_{i,j}(t)$ are same to constraints (6.6d) and (6.7d), namely, within $[0, 1]$. To address the challenge caused by the mixed integers, we relax the binary variables, $b_{i,m}(t)$ and $b_{i,j}(t)$, into real-valued ones, $b'_{i,m}(t) \in [0, 1]$ and $b'_{i,j}(t) \in [0, 1]$. As task division is not considered here, when measure the effect of an action at a given state, vehicle i under the overlapping area between the MeNB and UAV j will choose to offload its task to the MeNB if $b'_{i,m}(t) \geq b'_{i,j}(t)$, otherwise to UAV j. Also, we do additional processing on each action's elements to guarantee the total amount of resources allocated to all the associated vehicles is no more than $\{S_m, C^{co}_m, C^{ca}_m\}$ or $\{S_u, C^{co}_u, C^{ca}_u\}$, corresponding to the constraints (6.6e)-(6.6g) or (6.7e)-(6.7g) in the formulated problems. According to the actions defined by equations (6.11) and (6.12) and the value range of each element of $a_m(t)$ and $a_j(t)$, the action spaces for the MeNB and UAV $j \in \mathcal{U}$, $\mathcal{A}_m$ and $\mathcal{A}_j$, are continuous sets.

Reward: The reward is a function of state and action, which measures the effect of the action taken by an agent at a given state. Similar to any other learning algorithms [144], during the training stage, a corresponding reward will be returned to an agent at time slot t once the chosen action is taken by this agent at the previous time slot. Then according to the received reward, each agent updates its policy ($\boldsymbol{\pi}_m$ or $\boldsymbol{\pi}_j$) to direct to an optimal one, i.e., to a policy that the chosen actions at different environment states are always with high rewards. Denote the reward returned to the MeNB agent as r_m: $\mathcal{S} \times \mathcal{A}_m \mapsto \mathbb{R}$ and that to UAV j agent as r_j: $\mathcal{S} \times \mathcal{A}_j \mapsto \mathbb{R}$.

As the reward leads each agent to its optimal policy and the policy directly determines the association and resource allocation decision for the corresponding MEC server, the reward function should be designed based on the objectives of the original formulated problems. Thus, considering shaped rewards would help the algorithm to learn faster than sparse rewards [145, 146], the following two reward elements corresponding to $H(c^d_i(t) - T_i(t))$ and $H(f^{ca}_{i,m}(t)C^{ca}_m -$

$c_i^s(t)$) of equation (6.6) are designed for the MeNB agent,

$$r_{i,m}^d(t+1) = \log_2\left(\frac{c_i^d(t)}{T_i(t)} + 0.01\right), \tag{6.13}$$

$$r_{i,m}^s(t+1) = \log_2\left(\frac{f_{i,m}^{ca}(t)C_m^{ca}}{c_i^s(t)} + 0.01\right), \tag{6.14}$$

where $r_{i,m}^d(t+1)$ and $r_{i,m}^s(t+1)$ describe how far the delay and caching resources requested by vehicle i are satisfied by action $a_m(t)$, respectively. Specifically, we have $r_{i,m}^d(t+1) \geq 0$ and $r_{i,m}^s(t+1) \geq 0$ for vehicle i if its requested delay and caching resources are satisfied by action $a_m(t)$, respectively, and negative $r_{i,m}^d(t+1)$ and $r_{i,m}^s(t+1)$ are obtained otherwise. Similarly, the two reward elements for UAV j agent corresponding to equation (6.7) can be designed as

$$r_{i,j}^d(t+1) = \log_2\left(\frac{c_{i,j}^d(t)}{T_i(t)} + 0.01\right), \tag{6.15}$$

$$r_{i,j}^s(t+1) = \log_2\left(\frac{f_{i,j}^{ca}(t)C_u^{ca}}{c_{i,j}^s(t)} + 0.01\right). \tag{6.16}$$

As the formulated problems' objective is to maximize the number of offloaded tasks while satisfying their QoS requirements, the logarithmic function is adopted in the reward for fairness. The reward increases with the amounts of allocated resources. However, the incremental rate of a logarithm reward element slows once it reaches a positive value. Thus, instead of allocating more resources to parts of the offloaded tasks to achieve a higher reward, each agent is guided by the logarithm reward to allocate its resources to satisfy the QoS requirements of as many tasks as possible. Moreover, to avoid sharp fluctuation on the reward, a small value 0.01 is added to limit the minimum value of each reward element to $\log_2(0.01)$.

6.3.2 MADDPG-based Solution

To solve the above Markov game for $U+1$ agents, an MADDPG algorithm combines the DDPG algorithm with a cooperative multi-agent learning architecture. For each agent, indicated by equations (6.9), (6.10), (6.11), and (6.12), the observation and action spaces are continuous. Also, as the value range of each resource allocation element is $[0, 1]$ and their numerical

relationships must satisfy the original problems' constraints, disassembling the action space into discrete and efficient action vectors is difficult, especially for the cases with large size of action vectors. Hence, instead of RL algorithms for discrete state or action spaces, such as deep Q-network (DQN), the DDPG algorithm is adopted by each agent to address its corresponding MDP. Yet from the perspective of the whole Markov game, as the central controller is not considered in the network scenario, the RL algorithm with a single agent is infeasible. Moreover, to avoid spectrum and time cost on wireless communications among different MEC servers, we assume there is no information exchanging among different agents. Namely, only partial observation is available to each MEC server, and meanwhile, the decision made by one MEC server is unaware to others. Thus, considering the coupled relation among the formulated optimization problems, a cooperative multi-agent learning architecture with returning the same reward to the $U + 1$ agents is adopted to address the re-modeled Markov game to achieve the common objective of the original problems.

DDPG algorithm: The DDPG algorithm adopted by the MeNB agent is illustrated in the left of Figure 6.3, which combines the advantages of policy gradient and DQN. Here, we take the MeNB agent as an example to explain how to address the corresponding MDP with the DDPG algorithm, and which can be easily extended to the DDPG algorithm adopted by a UAV agent. Two main components, actor and critic, are included in the MeNB agent. According to policy $\boldsymbol{\pi}_m$, an action decision is made by the actor for each observation. Another component, critic, then uses a state-action function, $Q_m(\cdot)$, to evaluate the action chosen by the actor. Let s_m be the input state to the MeNB and γ be the discount factor to the immediate reward r_m. Then we have $Q_m(s_m, a_m) = \mathbb{E}[\sum_{\tau=0}^{\infty} \gamma^{\tau} r_m(t+\tau) | \boldsymbol{\pi}_m, s_m = s_m(t), a_m = a_m(t)]$, and which can be recursively re-expressed as $Q_m(s_m, a_m) = \mathbb{E}[(r_m|_{s_m,a_m}) + Q_m(s'_m, a'_m)]$. As in DQN, target networks and experience replay technology are adopted to improve the stabilization of DDPG. As shown in Figure 6.3, both the actor and the critic are implemented by two deep neural networks (DNNs), an evaluation network and a target network. And an experience replay buffer with size M_r is used to save transitions for the training stage.

As a type of policy gradient algorithm, the main idea of DDPG is to obtain an optimal policy $\boldsymbol{\pi}_m^*$ and learn the state-action function corresponding to $\boldsymbol{\pi}_m^*$, which is carried out by adjusting the parameters of the evaluation and target networks for the actor and the critic until convergence. In the above, the evaluation networks' parameters, $\boldsymbol{\theta}_m^{\mu}$ and $\boldsymbol{\theta}_m^{Q}$, are updated in real time. Specifically, a mini-batch of transitions with size M_b are randomly sampled from the replay buffer and inputted into the agent one by one. According to each inputted transition, the actor and the critic then update the parameters of the evaluation networks during the training stage.

Taking the i-th transition, $\{s_m^i, a_m^i, r_m^i, s_m^{i\prime}\}$, as an example, the critic adjusts the evaluation

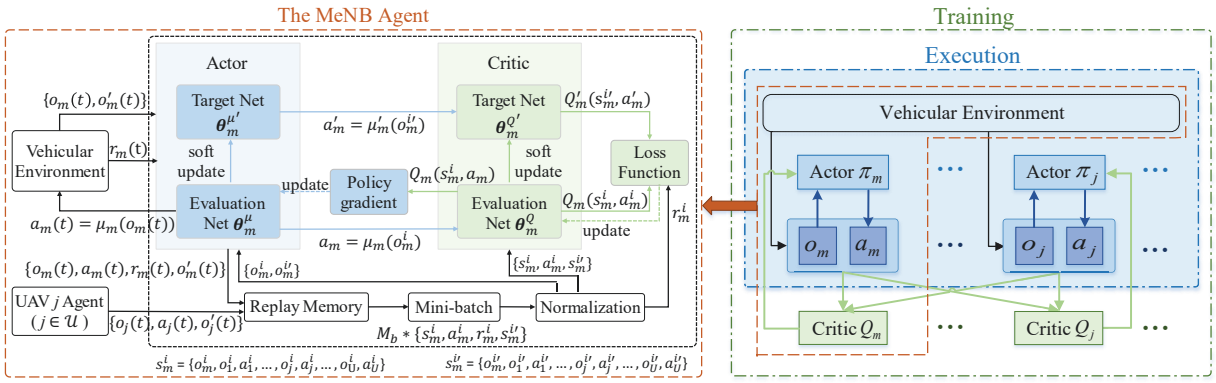


Figure 6.3: The MADDPG framework in the UAV-assisted MVNET.

network's parameters by minimizing the loss,

$$L(\boldsymbol{\theta}_m^Q) = \mathbb{E}[(Q_m(s_m^i, a_m^i) - (r_m^i + \gamma Q'_m(s_m^{i\prime}, a_m^{i\prime})))^2], \tag{6.17}$$

where $Q'_m(\cdot)$ is the state-action function for the target network. That is, if $L(\boldsymbol{\theta}_m^Q)$ is continuously differentiable, $\boldsymbol{\theta}_m^Q$ can be adjusted with the gradient of the loss function [147]. As the actor makes action decisions for each observation and each agent aims to maximize the cumulative reward, the evaluation network's parameters for the actor are updated by maximizing the policy objective function,

$$J(\boldsymbol{\theta}_m^\mu) = \mathbb{E}[Q_m(s_m^i, a_m)|a_m = \mu_m(o_m^i)], \tag{6.18}$$

where $\mu_m(\cdot)$ is the evaluation network function of the actor, which represents the deterministic policy $\boldsymbol{\pi}_m$: $\mathcal{O}_m \mapsto a_m$. As each association pattern variable is relaxed to $[0, 1]$, the action space of the MeNB agent, $\mathcal{A}_m$, is continuous, and so as $\mu_m(\cdot)$ is. Under this condition, we can conclude that $J(\boldsymbol{\theta}_m^\mu)$ is continuously differentiable according to [143], such that $\boldsymbol{\theta}_m^\mu$ can be adjusted in the direction of $\nabla_{\boldsymbol{\theta}_m^\mu} J(\boldsymbol{\theta}_m^\mu)$. With the real-time updated $\boldsymbol{\theta}_m^\mu$ and $\boldsymbol{\theta}_m^Q$, the parameters of the target networks, $\boldsymbol{\theta}_m^{\mu'}$ and $\boldsymbol{\theta}_m^{Q'}$, then can be softly updated as follows,

$$\begin{aligned} \boldsymbol{\theta}_m^{\mu'} &= \kappa_m^a \boldsymbol{\theta}_m^\mu + (1 - \kappa_m^a)\boldsymbol{\theta}_m^{\mu'}, \\ \boldsymbol{\theta}_m^{Q'} &= \kappa_m^c \boldsymbol{\theta}_m^Q + (1 - \kappa_m^c)\boldsymbol{\theta}_m^{Q'}, \end{aligned} \tag{6.19}$$

with $\kappa_m^a \ll 1$ and $\kappa_m^c \ll 1$.

MADDPG framework: As illustrated in the right figure of Figure 6.3, the MADDPG framework is composed of the vehicular environment and $U + 1$ agents, where each agent is

implemented by the DDPG algorithm similar to the MeNB agent. Benefiting from the actor and the critic compositions in the DDPG algorithm, centralized training and decentralized execution can be adopted directly in the MADDPG framework as in [148]. Next, we take the MeNB agent as an example to explain how to centrally train the MADDPG model and execute the learned model in a decentralized way.

In the centralized offline training stage, in addition to the local observation, extra information, i.e., observations and actions of all the UAV agents, is also available to the MeNB agent, $o_m(t)$ in (6.9). Namely, at time slot t, $\{o_j(t), a_j(t), o'_j(t)\}$ ($j \in \mathcal{U}$) is saved into the MeNB's replay buffer with $\{o_m(t), a_m(t), r_m(t), o'_m(t)\}$ together. For the i-th transition of the MeNB agent, $\{s^i_m, a^i_m, r^i_m, s^{i\prime}_m\}$, we have $s^i_m = \{o^i_m(t), o^i_1(t), a^i_1(t), \ldots, o^i_j(t), a^i_j(t), \ldots, o^i_U(t), a^i_U(t)\}$ and $s^{i\prime}_m = \{o^{i\prime}_m(t), o^{i\prime}_1(t), a^{i\prime}_1(t), \ldots, o^{i\prime}_j(t), a^{i\prime}_j(t), \ldots, o^{i\prime}_U(t), a^{i\prime}_U(t)\}$ as shown in Figure 6.3. When updating the parameters of the actor and the critic according to the inputted mini-batch of transitions, the actor chooses an action according to the local observation o^i_m, i.e., $a_m = \mu_m(o^i_m)$, and the chosen action and s^i_m then are valued by the critic. As the QoS satisfaction of each vehicle under the considered road segment is co-determined by the actions of the $U+1$ agents, using s^i_m which includes the information about other agents' actions to learn the state-action value function Q_m would ease training. Moreover, with the extra information, each agent allows to learn its state-action value function separately. Also, as aware of all other agents' actions, the environment is stationary to each agent during the offline training stage. Thus, the biggest concern to other multi-agent RL algorithms, i.e., the dynamic environment caused by other agents' actions, is addressed here. During the execution stage, as only local observation is required by the actor, each agent can obtain its action without aware of other agents' information.

Considering the common objective of the formulated optimization problems, the $U+1$ agents should cooperatively maximize the number of offloaded tasks while satisfying their QoS requirements. To achieve a cooperative Markov game for improving the overall resource utilization, we assume the same immediate reward r is returned to each agent [52], i.e., $r(t) = r_m(t) = r_j(t)$ ($j \in \mathcal{U}$). To avoid large fluctuation on reward, define $r(t) = \frac{1}{N(t)} \sum_{i \in \mathcal{N}(t)} r_i(t)$, where $r_i(t)$ is the reward achieved by vehicle i at time slot t and

$$r_i(t) = \begin{cases} r^d_{i,m}(t) + r^s_{i,m}(t), & \text{if } b_{i,m}(t) = 1 \\ r^d_{i,j}(t) + r^s_{i,j}(t), & \text{if } b_{i,j}(t) = 1. \end{cases} \tag{6.20}$$

MADDPG-based solution: According to the above discussion and Figure 6.3, the proposed MADDPG-based resource management scheme can be summarized in Algorithm 4. In Algorithm 4, continuous time slots are grouped into different episodes with M_s time slots included in each episode. To better describe the convergence performance, we let r' denote the total rewards achieved per episode. For example, for an episode starts at time slot t_0, we have

$r' = \sum_{t=t_0}^{M_s+t_0} r(t)$.

6.4 Simulation Results

In this section, we present simulation results to validate the proposed MADDPG-based resource management scheme. Specifically, we use PTV Vissim [134], a traffic simulation software, to simulate the vehicle mobility on a two-lane and two-way country road, where the behavior type is set to be "Freeway" and the number of inputs is 3. Environment states then can be obtained according to the vehicles' position information. With a mass of environment states, we train the MADDPG model[2] based on Algorithm 4 in the training stage. Then new environment states are used to test the performance of the learned model, i.e., the MADDPG-based resource management scheme, in the execution stage. In this section, we show the convergence performance of the MADDPG algorithm and compare it with a SADDPG algorithm [139]. Also, we compare the performance of the MADDPG-based resource management scheme with the SADDPG-based scheme and the random scheme. For the SADDPG algorithm, we assume a controller is installed at the MeNB and acts as the agent to centrally manage the resources available to the MeNB and UAVs. As this algorithm is based on DDPG and implemented with one agent, we call it as SADDPG to distinguish it from the MADDPG algorithm proposed in this work. For the random scheme, the association patterns and the amounts of resources allocated to vehicles under the coverage of an MEC server are randomly decided by the MeNB and/or a UAV.

Assume an MEC-mounted MeNB with 50 meters high is deployed on one side of the considered road segment [42]. On each side of the MeNB, two MEC-mounted UAVs are flying at speed 10 m/s in fixed altitude of 40 meters and parallel to the road to support vehicular applications [56, 65]. To guarantee an acceptable inter-UAV interference, assume the two UAVs are always flying with the same direction and the distance between them keeps at 600 meters. Vehicles under the coverage of a UAV can either associate with the UAV or the MeNB, and each vehicle's association pattern is co-determined by the association action elements of the MeNB and UAV. Similar to [141] and [126], the channel gains of uplinks from a vehicle to the MeNB and to the UAV are defined as $L_m(d'_m) = -40 - 35\log_{10}(d'_m)$ and $L_u(d'_u) = -30 - 35\log_{10}(d'_u)$, respectively, where d'_m (or d'_u) denotes the vehicle-MeNB (or vehicle-UAV) distance. As vehicles randomly generate heterogeneous computing tasks and the agents periodically manage the

[2]For the actors of the MeNB agent and the UAV agent, we respectively deploy two fully-connected hidden layers with $[1024, 512]$ and $[512, 256]$ neurons. And four fully-connected hidden layers with $[2048, 1024, 512, 256]$ neurons are deployed for their critics. Except for the output layer of each agent's actor which is activated by the *tanh* function, all other layers' neurons are activated by the *ReLU* function.

/* Initialization */
Initialize each agent's actor's and critic's evaluation and target networks to explore actions for the training stage;
Initialize the size of each agent's replay memory buffer.
/* Parameter updating */
foreach *episode* **do**
  Receive initial observations o_m and o_j ($j \in \mathcal{U}$), and set $r' = 0$.
  foreach *step* t **do**
    Each agent k selects action $a_k(t) = \mu_k(o_k(t))$ w.r.t. the current policy π_k and obtains the corresponding input state $s_k(t)$;
    Execute $a(t) = \{a_m(t), a_1(t), \ldots, a_U(t)\}$, receive reward $r(t)$, and obtain new observations o'_k and input states $s'_k(t)$ to each agent.
    foreach *each agent* k **do**
      if *the number of transitions* $< M_r$ **then**
        Store $\{s_k(t), a_k(t), r(t), s'_k\}$ into its replay buffer.
      else
        Replace the earliest saved transitions in the buffer with $\{s_k(t), a_k(t), r(t), s'_k\}$;
        Randomly select a mini-batch of transitions $\{s^i_k(t), a^i_k(t), r^i(t), s^{i\prime}_k\}$ with size $\mathcal{M}_b$ from the replay buffer;
        Update the parameter matrix of critic's evaluation network by minimizing the loss $L(\boldsymbol{\theta}^Q_k) = \frac{1}{M_b}\sum_{\mathcal{M}_b}(Q_k(s^i_k, a^i_k) - (r^i + \gamma Q'_k(s^{i\prime}_k, a^{i\prime}_k)))^2$;
        Update the parameter matrix of actor's evaluation network by maximizing the policy objective function $J(\boldsymbol{\theta}^\mu_k) = \frac{1}{M_b}\sum_{\mathcal{M}_b}(Q_k(s^i_k, a_k)|a_k = \mu_k(o^i_k))$;
        Update actor's and critic's target networks' parameters according to equation (6.19).
      end
    end
    $r' = r' + r(t)$.
  end
end

Algorithm 4: MADDPG-based Solution

available resources among the received resource access requests, we assume the task generation rate of each vehicle to be one task per time slot, and the computing task generated by vehicle i at time slot t is with $c_i^s(t) \in [0.5, 1]$ kbits, $c_i^c(t) \in [50, 100]$ MHz, and $c_i^d(t) \in [10, 50]$ ms. During the training stage, we fix the amounts of spectrum, computing, and caching resources available to the MeNB and UAVs to be $\{10\,\text{MHz}, 250\,\text{GHz}, 50\,\text{kbits}\}$ and $\{2\,\text{MHz}, 30\,\text{GHz}, 6\,\text{kbits}\}$, respectively. As a high learning rate speeds up the convergence of the RL algorithm while impacts the convergence stability, we take exponential decay/augment on the actors' and critics' learning rates, as well as on κ_a/κ_c and the reward discount factor. Unless otherwise specified, other parameters used in the training and execution stages are listed in Table 6.1.

Table 6.1: Parameters for the learning stage.

Parameter	**Value**
Vehicle's transmit power	1 watt
Communication range of the MeNB/UAV	$600/100$ m
Length of the considered road segment	1200 m
Background noise power	-104 dBm
Size of the replay buffer	10000
Size of a mini-batch	32
Actor's learning rate	Decaying from 0.0002 to 0.0000001
Critic's learning rate	Decaying from 0.002 to 0.000001
κ_a/κ_c	Decaying from 0.02 to 0.0001
Reward discount factor	Augmenting from 0.8 to 0.99

Figure 6.4 shows the rewards achieved per episode in the training stages of the MADDPG and SADDPG algorithms. As all parameters of the MeNB and the two UAV agents are globally initialized by the TensorFlow based on the initial state in the first episode, the rewards achieved by one episode are small and fluctuate dramatically in the first 200 episodes of the MADDPG algorithm. Yet, with the training going, the actors and critics adjust their evaluation and target networks' parameters to gradually approximate the optimal policies and the state-action functions corresponding to the optimal policies, respectively. Therefore, relatively high and stable rewards are achieved starting from the 200-th episode. Comparing with the SADDPG algorithm, the MADDPG algorithm converges as quickly as the SADDPG algorithm does although the achieved rewards are a little bit less stable.

As discussed above, three agents, i.e., one MeNB agent and two UAV agents, are trained centrally by the MADDPG algorithm. To demonstrate the convergence performance of the three agents, we show the varying tendency of the Q-values obtained from the evaluation network

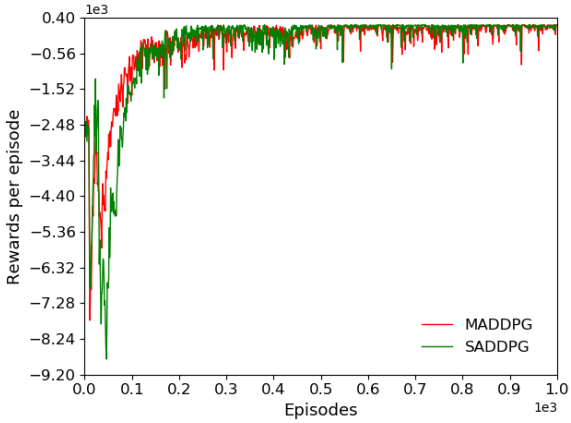


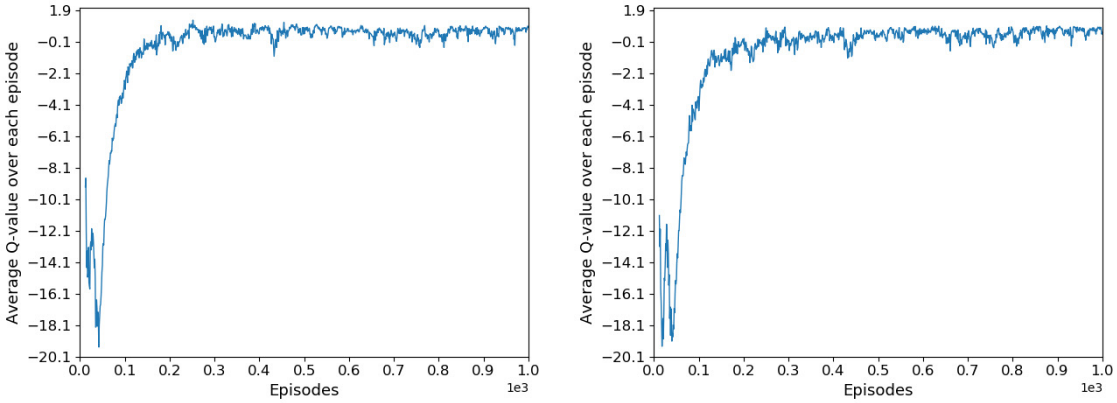
Figure 6.4: Rewards achieved per episode in the training stage.

of each agent's critic in Figure 6.5. With the training going, positive and stable Q-values are obtained by the three agents starting from the 200-th episode, which implies that, the training processes converge within 200 episodes and their convergence rates are consistent with the rewards achieved per episode of the MADDPG algorithm.

Similar to [54], we use the delay/QoS satisfaction ratios, defined as the proportions of the offloaded tasks with satisfied delay/QoS requirements, to measure the performance of different resource management schemes. As the amounts of resources carried at the MEC-mounted MeNB and UAVs are always pre-allocated, we use fixed amounts of resources to train the MADDPG model. During the online execution stage, the learned model is tested over $10,000$ continuous environment states.

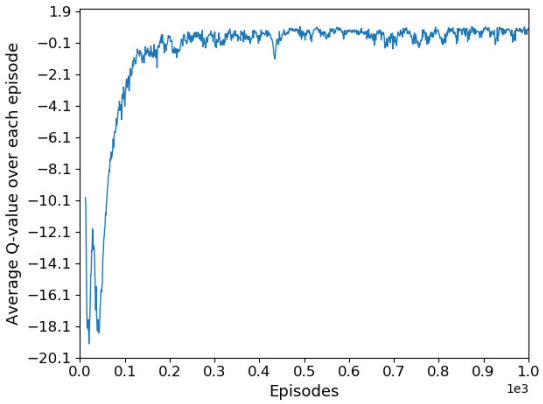
Figure 6.6 shows the average delay/QoS satisfaction ratios achieved by different schemes versus the amounts of spectrum, computing, and caching resources available to the MEC-mounted MeNB and UAVs, respectively[3]. Different from the random scheme, both MADDPG- and SADDPG-based schemes jointly manage the multi-dimensional resources available to the MeNB and UAVs to satisfy the offloaded tasks' QoS requirements. Thus, under the scenarios with different amounts of available spectrum, computing, and caching resources, more than doubled delay or QoS satisfaction ratios are achieved by the MADDPG- and SADDPG-based schemes than the random one. As the delay satisfaction ratio is not affected by the allocation of caching resources, only the QoS satisfaction ratio curves are described in Figures 6.6(c)

[3]Unless specified in the x-coordinate of each figure, the amounts of spectrum, computing, and caching resources available to the MeNB and UAVs during each test are fixed to $\{6\,\text{MHz}, 250\,\text{GHz}, 50\,\text{kbits}\}$ and $\{0.5\,\text{MHz}, 25\,\text{GHz}, 5\,\text{kbits}\}$, respectively.



(a) MeNB agent

(b) UAV 1 agent



(c) UAV 2 agent

Figure 6.5: The averaged Q-value over each episode under the MADDPG algorithm.

and 6.6(f). Constrained by the achieved delay satisfaction ratios, the QoS satisfaction ratios of the MADDPG- and SADDPG-based schemes tend to stable and reach the corresponding delay satisfaction ratios with the increasing of caching resources. However, for the random scheme, as the gap between the delay and QoS satisfaction ratios is relatively large, the QoS satisfaction ratio increases slowly when the amounts of caching resources at the MeNB and UAVs increase from 10 MHz to 100 MHz and from 1 MHz to 10 MHz, respectively. Moreover, as the delay requirement is a part of the QoS requirement, i.e., satisfying the delay requirement is indispensable for a task with satisfied QoS requirement, the delay satisfaction ratio is always higher than the corresponding QoS satisfaction ratio for the three schemes.

As we know, when leveraging learning-based methods to solve an optimization problem, only sub-optimal results can be obtained in most cases [149]. In the MADDPG-based scheme, three agents are cooperatively trained to solve three MDPs to achieve a maximum common reward. Allowing each agent to learn other agents' policies during the training stage, issues caused by the coupled relation among the optimization problems are partially addressed. The central optimization problem is much more complex than the ones formulated for the MeNB agent or the UAV agent, and which makes solving it by the SADDPG algorithm more challenging. Thus, in addition to avoiding extra spectrum and time cost on exchanging information between the central controller and the UAV[4], a higher delay/QoS satisfaction ratio is even achieved by the MADDPG-based scheme than the SADDPG-based one under most of the scenarios, as shown in the zoom-in figures in Figure 6.6. Moreover, the gap between the delay and QoS satisfaction ratios achieved by the MADDPG-based scheme is smaller than that of the SADDPG-based one, which indicates that the MADDPG-based scheme can better manage the multi-dimensional resources.

As discussed in subsection 6.3.1, reward element r_d is defined to measure how far the task's delay requirement is satisfied, which is co-determined by the spectrum and computing resource management results. However, for both MADDPG and SADDPG algorithms, allocating more spectrum resources to satisfy the delay requirement of an offloaded task is the same as allocating more computing resources. Namely, performance difference would be resulted between the spectrum and computing resource management for both learned MADDPG and SADDPG models. From the zoom in figures in Figures 6.6(a) and 6.6(b), under the scenarios with a small amount of spectrum and a large amount of computing resources at the MeNB, higher delay satisfaction ratios are achieved by the SADDPG-based scheme than the MADDPG-based one, which means that more optimal spectrum management is obtained by the SADDPG-based scheme[5] while more optimal computing management is obtained by the MADDPG-based scheme. As most of the vehicles choose to offload tasks to the MeNB, the amounts of resources

[4]For the SADDPG-based scheme, we assume extra spectrum resources are used for wireless communications between the UAV and the MeNB and time cost on them are not counted in Figure 6.6.

[5]The communication delay satisfaction ratio, defined as the proportions of tasks with $\frac{c_i^s(t)}{R_{i,m}(t)} \leq c_i^d(t)$ or

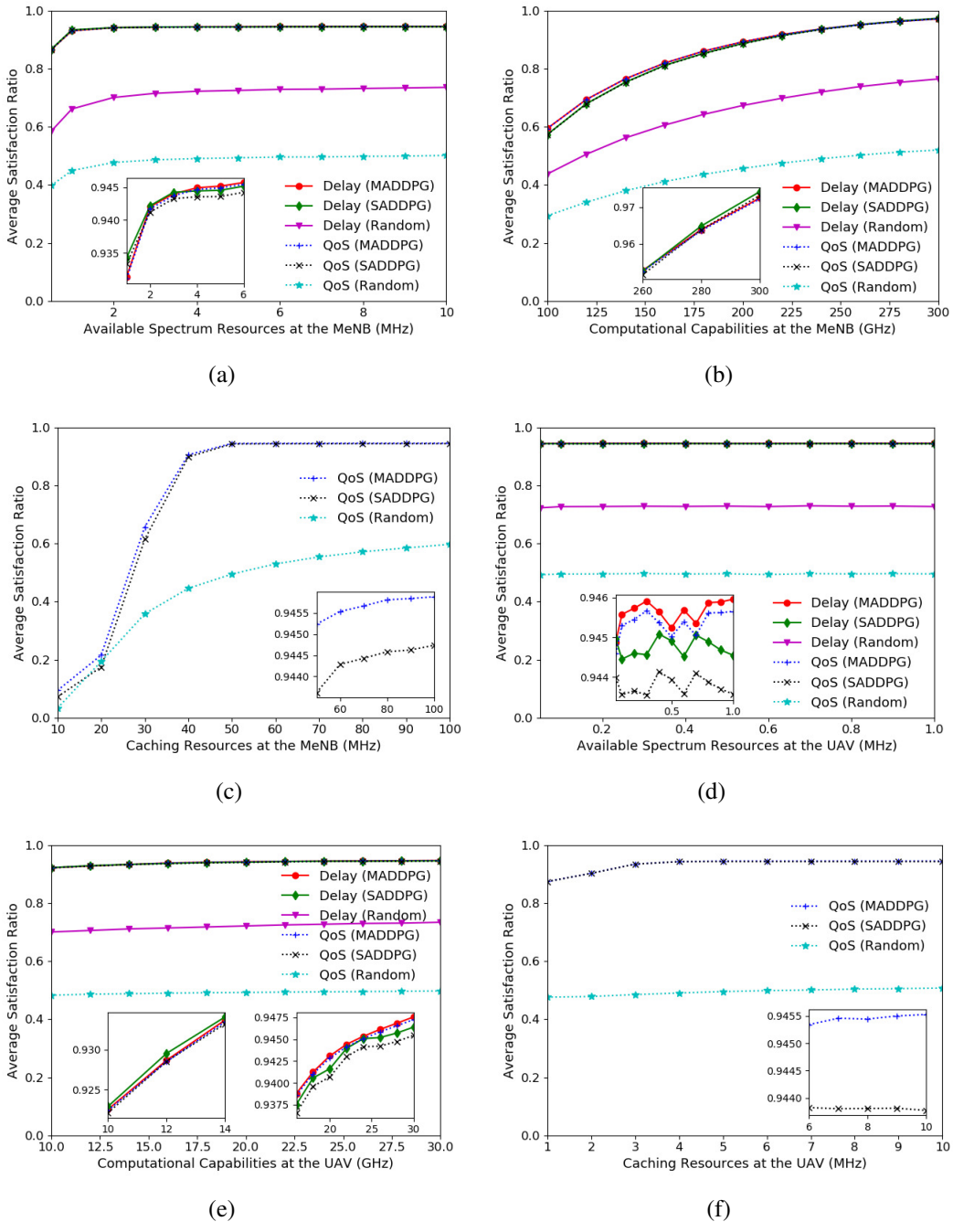


Figure 6.6: The average delay/QoS satisfaction ratios achieved by different schemes versus the amounts of spectrum, computing, and caching resources at the MeNB (subfigures (a), (b), and (c)) and UAVs (subfigures (d), (e), and (f)).

at the MeNB mainly determine the delay/QoS satisfaction ratios achieved by the three schemes when very small amounts of resources at the UAV, as shown in Figures 6.6(d), 6.6(e), and 6.6(f). And with the increase of resources at the UAV, the varying tendencies of the delay/QoS satisfaction ratios achieved by the three schemes are then gradually similar to Figures 6.6(a), 6.6(b), and 6.6(c).

6.5 Summary

In this chapter, we have studied multi-dimensional resource management in the UAV-assisted MVNETs. To cooperatively support the heterogeneous and delay-sensitive vehicular applications, an MADDPG-based scheme has been proposed to distributively manage the spectrum, computing, and caching resources available to the MEC-mounted MeNB and UAVs. For the high dynamic vehicular scenarios with delay-sensitive and computing-intensive applications, the MADDPG-based scheme can rapidly make vehicle association decisions and allocate proper amounts of multi-dimensional resources to vehicle users to achieve high delay/QoS satisfaction ratios. In the next chapter, we will conclude this thesis and show our future research directions.

$\frac{c_i^s(t)}{R_{i,j}(t)} \leq c_i^d(t)$, achieved by the SADDPG-based scheme is always higher than that of the MADDPG-based scheme.

Chapter 7

Conclusions and Future Works

In this chapter, we summarize the main results and contributions of this thesis and present our future research directions.

7.1 Main Research Contributions

In this thesis, we have investigated the resource management issues in MEC- and/or UAV-assisted vehicular networks. Specifically, we have proposed an MEC-assisted vehicular network architecture by integrating the SDN and NFV concepts. Under the proposed network architecture, the spectrum and multi-dimensional resource management issues have been studied. To overcome the issue of imbalance between supply and demand of resources remaining in the vehicular scenarios supported by only MEC-mounted BSs, we have also extended the proposed MVNET architecture into a UAV-assisted MVNET and studied the multi-dimensional resource management issue under it. The main contributions of this thesis are summarized as follows.

1. A novel vehicular network architecture by integrating the SDN and NFV concepts into MEC has been proposed. In which, functions supporting different applications can be host on servers flexibly with reduced function provisioning cost. The SDN control functionality is applied to achieve intelligent traffic steering and efficient multi-resource management. The proposed architecture can be used to efficiently support the emerging vehicular applications and services;

2. A dynamic two-tier spectrum management framework has been proposed to improve the spectrum resource utilization under the MVNET architecture, which considers the trade-

off between spectrum resource utilization and inter-cell interference. In specific, by leveraging logarithmic and linear utility functions, we have formulated three aggregate network utility maximization problems to fairly slice spectrum resources among BSs connected to the same MEC server, optimize BS-vehicle association patterns and resource allocation, and control the transmit power of BS. To jointly solve the three formulated optimization problems, linear programming relaxation and first-order Taylor series approximation have been used and an ACS algorithm has been designed;

3. We have investigated the joint spectrum, computing, and caching resource management problem to accommodate delay-sensitive applications in the MVNET. Particularly, two typical MEC architectures, i.e., with MeNB- and EN-mounted MEC servers, have been considered, and under which we have formulated two resource optimization problems to maximize the number of offloaded tasks that are completed with satisfied QoS requirements. To solve the formulated problems in real time, we have exploited the DRL for problem transformation and devised the DDPG- and HDDPG-based algorithms;

4. We have studied multi-dimensional resource management in the UAV-assisted MVNETs. To distributively manage the spectrum, computing, and caching resources available to the MEC-mounted MeNB and UAVs to support the heterogeneous and delay-sensitive vehicular applications, we have proposed an MADDPG-based scheme. For the high dynamic vehicular scenarios with delay-sensitive and computing-intensive applications, the proposed MADDPG-based scheme can rapidly make vehicle association decisions and allocate proper amounts of multi-dimensional resources to vehicle users to achieve high delay/QoS satisfaction ratios.

The architecture and schemes proposed in this thesis should provide useful guidelines for future research in multi-dimensional resource management scheme designing and resource utilization enhancement in highly dynamic wireless networks with diversified applications and services.

7.2 Future Works

For future research, I plan to design a comprehensive satellite-terrestrial network architecture enabled by MEC and cloud computing to support vehicular applications, focusing on design AI-based multi-dimensional resource management schemes. Combining satellite networks with the MEC- and/or UAV-assisted vehicular cyber-physical system, i.e., enabling satellite-terrestrial vehicular networks, can provide seamless signal coverage, high-capacity communication, and more flexible Internet access to support a large number of vehicle applications and services. For

example, the satellite antennas can be embedded into the roof of an autonomous vehicle to allow it to acquire satellite signals even without terrestrial BS coverage. Moreover, due to the complex vehicular environments, constructing fiber-equipped or stable wireless backhaul links to enable idea backhaul capacity for every small BSs is usually impracticable. Embedding the satellite antennas into small BSs to enable satellite-based small cells can provide users with a high-capacity backhaul to access the cloud computing server. However, due to the mobility of vehicles, UAVs, and satellites, challenges arise in satisfying the high requirements of dynamic adaptation and rapid response for intelligent vehicular networks. How to achieve fast, adaptive, efficient, and privacy-preserved resource management is in urgent demand for the satellite-terrestrial vehicular network.

1. **Adaptive Multi-radio Access Design:** In the satellite-terrestrial vehicular network, vehicle users, including manually driving vehicles and autonomous vehicles, are allowed to access the network via either a satellite, UAV, or terrestrial BS. To support the V2X communications, different communication techniques, such as DSRC and LTE, and new radio allocation and access techniques, including orthogonal frequency-division multiplexing (OFDM), non-orthogonal multiple access (NOMA), multiple-input-multiple-output (MIMO), and beamforming, can be applied in the satellite-terrestrial vehicular network. Designing access schemes by intelligently scheduling the above-mentioned access techniques and adaptively allocating the spectrum resources (including spectrum resources for both satellite and terrestrial communications) among vehicles to improve the spectrum utilization while satisfying the delay-sensitive and high-throughput requirement of V2X communication is critical and is one of my research focuses shortly.

2. **Multi-Dimensional Resource Management Scheme:** In addition to providing computing capabilities for computing-intensive applications, MEC servers can also benefit many data-sensitive applications, such as video streaming and online gaming. Caching some popular content at the edge can help to mitigate the traffic congestion in the core network and improve content delivery efficiency. Optimally placing the content at the terrestrial, aerial, and space MEC servers to maximize the storage resource utilization while allocating a proper amount of spectrum for delivering the content to vehicles is one of the critical research problems in the satellite-terrestrial vehicular network. On the other hand, with more vehicular applications emerging, diversified tasks, such as computing task offloading and content delivery, are usually generated by vehicles and require multi-dimensional resources. Designing a multi-dimensional resource management scheme to effectively manage the uplink/downlink communication, computing, and storage resources is still urgent. Considering the heterogeneous quality of service requirements of different applications and the highly dynamic vehicular network environment, AI technologies can

be utilized to effectively manage the multi-dimensional resources here. For scenarios with multiple vehicular services that require multi-dimensional resource support, the procedure of multi-dimensional resource management can be summarized into two stages, multi-dimensional resource planning among services (also referred to as service slicing) and multi-dimensional resource scheduling among vehicles. We can slice the multi-dimensional resources among different services based on the statistical resource demanding from each service either via optimization or AI methods. For the multi-dimensional resource scheduling, to reduce the complexity of the learning algorithm and increase the convergence rate, we can decompose the resource scheduling problem into two sub-problems, computing task offloading and content placement/delivery, and then adopt the hierarchical learning architecture to solve them.

3. **Privacy-Preserved Resource Management Scheme:** Data-privacy preservation is always an important concern for vehicles. Vehicles have to share their data with the cloud or MEC server to support some AI-powered applications or services. For example, to enable automated driving, connected vehicles need to share their sensing data with the server for learning the optimal automated driving strategy. How to preserve the privacy of vehicles' data while allowing it to benefit from these applications is very important to the satellite-terrestrial vehicular network. One of the potential technologies to preserve data privacy in AI-powered applications is federated learning, also known as collaborative learning. With the federated learning technique, we can enable decentralized learning, namely, building the learning model at the MEC server while training it in vehicles without uploading raw data to the server, or building the learning model at the cloud computing server while training it in MEC servers.

References

[1] H. Peng, L. Liang, X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1064–1078, Feb. 2019.

[2] L. Liang, H. Peng, G. Y. Li, and X. Shen, "Vehicular communications: A physical layer perspective," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10647–10659, Dec. 2017.

[3] W. Zeng, M. Khalid, and S. Chowdhury, "In-vehicle networks outlook: Achievements and challenges," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 3, pp. 1552–1571, 3rd Quarter, 2016.

[4] M. Azees, P. Vijayakumar, and L. Deborah, "Comprehensive survey on security services in vehicular ad-hoc networks," *IET Intell. Transp. Syst.*, vol. 10, no. 6, pp. 379–388, Jul. 2016.

[5] L. Xu, L. Wang, G. Yin, and H. Zhang, "Communication information structures and contents for enhanced safety of highway vehicle platoons," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4206–4220, Nov. 2014.

[6] D. L. Guidoni, G. Maia, F. S. Souza, L. A. Villas, and A. A. Loureiro, "Vehicular traffic management based on traffic engineering for vehicular ad hoc networks," *IEEE Access*, vol. 8, pp. 45167–45183, March 2020.

[7] M. Berk, O. Schubert, H.-M. Kroll, B. Buschardt, and D. Straub, "Exploiting redundancy for reliability analysis of sensor perception in automated driving vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 12, pp. 5073–5085, Dec. 2020.

[8] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 1, pp. 263–284, 1st Quarter 2016.

[9] S. Zeadally, R. Hunt, Y. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges," *Telecommun. Syst.*, vol. 50, no. 4, pp. 217–241, Aug. 2012.

[10] X. Ma, J. Zhang, X. Yin, and K. Trivedi, "Design and analysis of a robust broadcast scheme for VANET safety-related services," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 46–61, Jan. 2012.

[11] S. Biswas, R. Tatchikou, and F. Dion, "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety," *IEEE Commun. mag.*, vol. 44, no. 1, pp. 74–82, Jan. 2006.

[12] Q. Ye and W. Zhuang, "Token-based adaptive MAC for a two-hop internet-of-things enabled MANET," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1739–1753, Oct. 2017.

[13] H. Peng, Q. Ye, and X. Shen, "SDN-based resource management for autonomous vehicular networks: A multi-access edge computing approach," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 156–162, Aug. 2019.

[14] M. Togou, A. Hafid, and L. Khoukhi, "SCRP: Stable cds-based routing protocol for urban vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1298–1307, May 2016.

[15] L. Zhu, C. Li, B. Li, X. Wang, and G. Mao, "Geographic routing in multilevel scenarios of vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7740–7753, Sep. 2016.

[16] SAE On-Road Automated Vehicle Standards Committee and others, "(R) Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," *SAE Standard J3016*, pp. 1–30, 2016.

[17] W. Li, X. Ma, J. Wu, K. S. Trivedi, X.-L. Huang, and Q. Liu, "Analytical model and performance evaluation of long-term evolution for vehicle safety services," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 1926–1939, March 2017.

[18] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 90–96, Spring 2017.

[19] P. Blyth, M. Mladenovic, B. Nardi, H. Ekbia, and N. Su, "Expanding the design horizon for self-driving vehicles: Distributing benefits and burdens," *IEEE Technol. Society Mag.*, vol. 35, no. 3, pp. 44–49, Sep. 2016.

[20] H. Marzbani, H. Khayyam, C. N. To, . V. Quoc, and R. N. Jazar, "Autonomous vehicles: Autodriver algorithm and vehicle dynamics," *IEEE Transa. Veh. Technol.*, vol. 68, no. 4, pp. 3201–3211, April 2019.

[21] E. Moradi-Pari, A. Tahmasbi-Sarvestani, and Y. Fallah, "A hybrid systems approach to modeling real-time situation-awareness component of networked crash avoidance systems," *IEEE Syst. J.*, vol. 10, no. 1, pp. 169–178, Mar. 2016.

[22] T. Nguyen, A. Islam, and Y. M. Jang, "Region-of-interest signaling vehicular system using optical camera communications," *IEEE Photonics J.*, vol. 9, pp. 1–21, Feb. 2017.

[23] M. N. Sial, Y. Deng, J. Ahmed, A. Nallanathan, and M. Dohler, "Stochastic geometry modeling of cellular V2X communication over shared channels," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11873–11887, Dec. 2019.

[24] S. R. Pokhrel and J. Choi, "Improving TCP performance over WiFi for internet of vehicles: A federated learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6798–6802, June 2020.

[25] G. Ding, J. Wang, Q. Wu, Y.-D. Yao, F. Song, and T. A. Tsiftsis, "Cellular-base-station-assisted device-to-device communications in TV white space," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 107–121, Jan. 2016.

[26] J. Kenney, "Dedicated short-range communications (DSRC) standards in the united states," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, July 2011.

[27] 1609 WG - Dedicated Short Range Communication Working Group, "1609.2-2016 - IEEE standard for wireless access in vehicular environments–security services for applications and management messages," 2016.

[28] L. Cheng, B. Henty, D. Stancil, F. Bai, and P. Mudalige, "Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1501–1516, Oct. 2007.

[29] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: a survey," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 148–157, May 2013.

[30] J. Mei, K. Zheng, L. Zhao, L. Lei, and X. Wang, "Joint radio resource allocation and control for vehicle platooning in LTE-V2V network," *IEEE Transa. Veh. Technol.*, vol. 67, no. 12, pp. 12218–12230, Dec. 2018.

[31] L. Liang, G. Li, and W. Xu, "Resource allocation for D2D-Enabled vehicular communications," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 3186–3197, July 2017.

[32] C.-s. Choi and F. Baccelli, "Modeling and analysis of vehicle safety message broadcast in cellular networks," *IEEE Trans. Wireless Commun.*, to appear.

[33] N. Lu, N. Cheng, N. Zhang, X. Shen, J. Mark, and F. Bai, "Wi-Fi hotspot at signalized intersection: Cost-effectiveness for vehicular internet access," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3506–3518, May 2016.

[34] B. Fan, H. Tian, S. Zhu, Y. Chen, and X. Zhu, "Traffic-aware relay vehicle selection in millimeter-wave vehicle-to-vehicle communication," *IEEE Wireless Commun. Letters*, vol. 8, no. 2, pp. 400–403, April 2019.

[35] W. Xu, H. Omar, W. Zhuang, and X. Shen, "Delay analysis of in-vehicle internet access via on-road WiFi access points," *IEEE Access*, vol. 5, pp. 2736–2746, Feb. 2017.

[36] H. Zhou, N. Cheng, Q. Yu, X. Shen, D. Shan, and F. Bai, "Toward multi-radio vehicular data piping for dynamic DSRC/TVWS spectrum sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2575–2588, Oct. 2016.

[37] J. Lim, W. Kim, K. Naito, J. Yun, D. Cabric, and M. Gerla, "Interplay between TVWS and DSRC: Optimal strategy for safety message dissemination in VANET," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 11, pp. 2117–2133, Nov. 2014.

[38] Y. Han, E. Ekici, H. Kremo, and O. Altintas, "Vehicular networking in the tv white space band: Challenges, opportunities, and a media access control layer of access issues," *IEEE Veh. Technol. Mag.*, vol. 12, pp. 52–59, June 2017.

[39] K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of DSRC and cellular network technologies for V2X communications: A survey," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9457–9470, Dec. 2016.

[40] H. Peng, L. Liang, X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1064–1078, Feb. 2019.

[41] J. Zhang, G. Lu, H. Yu, Y. Wang, and C. Yang, "Effect of the uncertainty level of vehicle-position information on the stability and safety of the car-following process," *IEEE Transa. Intell. Transp. Syst.*, to appear.

[42] H. Peng, D. Li, Q. Ye, K. Abboud, H. Zhao, W. Zhuang, and X. Shen, "Resource allocation for cellular-based inter-vehicle communications in autonomous multiplatoons," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11249–11263, Dec. 2017.

[43] W. Yang, B. Wan, and X. Qu, "A forward collision warning system using driving intention recognition of the front vehicle and V2V communication," *IEEE Access*, vol. 8, pp. 11268–11278, Jan. 2020.

[44] U. Hernandez-Jayo, A. Mammu, and I. De-la Iglesia, "Reliable communication in cooperative ad hoc networks," in *Contemporary Issues in Wireless Commun.*, InTech, Nov. 2014.

[45] K. Guenter, "Use of multi-access edge computing to power artificial intelligence for automotive." `https://iot-automotive.news/vodafone-use-of-multi-access-edge-computing-to-power-artificial-intelligence-for-automotive/`. Accessed Feb. 12, 2021.

[46] Z. Su, Y. Hui, and T. H. Luan, "Distributed task allocation to enable collaborative autonomous driving with network softwarization," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2175–2189, Oct. 2018.

[47] H. Peng, D. Li, K. Abboud, H. Zhou, H. Zhao, W. Zhuang, and X. Shen, "Performance analysis of ieee 802.11 p DCF for multiplatooning communications with autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2485–2498, Mar. 2017.

[48] C. Huang, M. Chiang, D. Dao, W. Su, S. Xu, and H. Zhou, "V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture," *IEEE Access*, vol. 6, pp. 17741–17755, Mar. 2018.

[49] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, Feb. 2020.

[50] S. Gurugopinath, P. C. Sofotasios, Y. Al-Hammadi, and S. Muhaidat, "Cache-aided non-orthogonal multiple access for 5G-enabled vehicular networks," *IEEE Trans. Veh. Tech.*, vol. 68, no. 9, pp. 8359–8371, Sept. 2019.

[51] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Tech.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[52] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, Oct. 2019.

[53] H. Zhou, H. Wang, X. Chen, X. Li, and S. Xu, "Data offloading techniques through vehicular Ad Hoc networks: A survey," *IEEE Access*, vol. 6, pp. 65250–65259, Nov. 2018.

[54] H. Peng and X. Shen, "Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2416–2428, Dec. 2020.

[55] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879–1892, Apr. 2019.

[56] X. Hu, K.-K. Wong, K. Yang, and Z. Zheng, "UAV-assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4738–4752, Oct. 2019.

[57] W. Chen, B. Liu, H. Huang, S. Guo, and Z. Zheng, "When UAV swarm meets edge-cloud computing: The QoS perspective," *IEEE Network*, vol. 33, no. 2, pp. 36–43, Apr. 2019.

[58] L. Zhang, Z. Zhao, Q. Wu, H. Zhao, H. Xu, and X. Wu, "Energy-aware dynamic resource allocation in UAV assisted mobile edge computing over social internet of vehicles," *IEEE Access*, vol. 6, pp. 56700–56715, Oct. 2018.

[59] H. Peng, Q. Ye, and X. Shen, "Spectrum management for multi-access edge computing in autonomous vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 3001–3012, July 2020.

[60] F. Lyu, J. Ren, N. Cheng, P. Yang, M. Li, Y. Zhang, and X. Shen, "LEAD: large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics," *IEEE Trans. Mobile Comput.*, to appear.

[61] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Trans. Veh. Tech.*, vol. 68, no. 10, pp. 10187–10200, Oct. 2019.

[62] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching," *Proc. IEEE*, vol. 108, no. 2, pp. 274–291, Feb. 2020.

[63] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1133–1146, Apr. 2020.

[64] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Tech.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[65] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, Sept. 2019.

[66] "Building an ecosystem for responsible drone use and development on Microsoft Azure." `https://azure.microsoft.com/en-ca/blog/building-an-ecosystem-for-responsible-drone-use-and-development-on-microsoft-azure/`. Accessed Jan. 1, 2021.

[67] Y. Wei, F. R. Yu, M. Song, and Z. Han, "Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural Actor-Critic deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2061 –2073, Apr. 2019.

[68] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, B. Hu, and V. C. Leung, "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4283–4294, June 2019.

[69] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, Jan. 2021.

[70] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5G network slicing for vehicle-to-everything services," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 38–45, Dec. 2017.

[71] J. Mei, X. Wang, and K. Zheng, "Intelligent network slicing for V2X services toward 5G," *IEEE Network*, vol. 33, no. 6, pp. 196–204, Nov./Dec. 2019.

[72] H. D. R. Albonda and J. Pérez-Romero, "An efficient RAN slicing strategy for a heterogeneous network with eMBB and V2X services," *IEEE Access*, vol. 7, pp. 44771–44782, April 2019.

[73] W. Wu, N. Chen, C. Zhou, M. Li, X. Shen, W. Zhuang, and X. Li, "Dynamic RAN slicing for service-oriented vehicular networks via constrained learning," *IEEE J. Sel. Areas Commun.*, to appear.

[74] Z. Mlika and S. Cherkaoui, "Network slicing with mec and deep reinforcement learning for the internet of vehicles," *IEEE Network*, to appear.

[75] Y. Yao, L. Rao, and X. Liu, "Performance and reliability analysis of IEEE 802.11 p safety communication in a highway environment," *IEEE trans. veh. technol.*, vol. 62, no. 9, pp. 4198–4212, 2013.

[76] Y. Han, E. Ekici, H. Kremo, and O. Altintas, "Throughput-efficient channel allocation algorithms in multi-channel cognitive vehicular networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 757–770, 2017.

[77] W. Xu, W. Shi, F. Lyu, H. Zhou, N. Cheng, and X. Shen, "Throughput analysis of vehicular internet access via roadside WiFi hotspot," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3980–3991, Apr. 2019.

[78] J. Eze, S. Zhang, E. Liu, and E. Eze, "Cognitive radio-enabled internet of vehicles: A cooperative spectrum sensing and allocation for vehicular communication," *IET Networks*, vol. 7, no. 4, pp. 190–199, 2018.

[79] Y. Han, E. Ekici, H. Kremo, and O. Altintas, "Resource allocation algorithms supporting coexistence of cognitive vehicular and ieee 802.22 networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1066–1079, 2017.

[80] L. Liang, Y. Li, and W. Xu, "Resource allocation for d2d-enabled vehicular communications," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 3186–3197, July 2017.

[81] F. Abbas, P. Fan, and Z. Khan, "A novel low-latency V2V resource allocation scheme based on cellular V2X communications," *IEEE Transa. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2185–2197, June 2019.

[82] Y. Chen, Y. Wang, M. Liu, J. Zhang, and L. Jiao, "Network slicing enabled resource management for service-oriented ultra-reliable and low-latency vehicular networks," *IEEE Transa. Veh. Technol.*, vol. 69, no. 7, pp. 7847–7862, July 2020.

[83] Intel, "Self-driving car technology and computing requirements."

[84] Y. Sun, X. Guo, J. Song, S. Zhou, Z. Jiang, X. Liu, and Z. Niu, "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3061–3074, Apr. 2019.

[85] J. Shi, J. Du, J. Wang, J. Wang, and J. Yuan, "Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning," *IEEE Transa. Veh. Technol.*, vol. 69, no. 12, pp. 16067–16081, Dec. 2020.

[86] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet Things*, vol. 6, no. 3, pp. 4377–4387, June 2019.

[87] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, "Efficient mobility-aware task offloading for vehicular edge computing networks," *IEEE Access*, vol. 7, pp. 26652–26664, Feb. 2019.

[88] Z. Zhou, J. Feng, Z. Chang, and X. Shen, "Energy-efficient edge computing service provisioning for vehicular networks: A consensus ADMM approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5087–5099, May 2019.

[89] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[90] L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10190–10203, Nov. 2018.

[91] T. Q. Dinh, Q. D. La, T. Q. Quek, and H. Shin, "Learning for computation offloading in mobile edge computing," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6353–6367, Dec. 2018.

[92] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, B. Hu, and V. C. Leung, "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching," *IEEE Internet Things*, vol. 6, no. 3, pp. 4283–4294, June 2019.

[93] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Tech.*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[94] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning based mode selection and resource management for green fog radio access networks," *IEEE Internet Things*, vol. 6, no. 2, pp. 1960–1971, Apr. 2019.

[95] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, B. Hu, and R. Y. Kwok, "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transa. Intell. Transp. Syst.*, to appear.

[96] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things*, vol. 5, no. 1, pp. 283–294, Feb. 2018.

[97] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "Joint resource allocation for software-defined networking, caching, and computing," *IEEE/ACM Trans. Net.*, vol. 26, no. 1, pp. 274–287, Feb. 2018.

[98] J. Su, J. Wu, P. Cheng, and J. Chen, "Autonomous vehicle control through the dynamics and controller learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5650 – 5657, July 2018.

[99] A. Fortelle, X. Qian, S. Diemer, J. Grégoire, F. Moutarde, S. Bonnabel, A. Marjovi, A. Martinoli, I. Llatser, A. Festag, and K. Sjoberg, "Network of automated vehicles: the AutoNet 2030 vision," in *ITS World Congr.*, 2014.

[100] Ş. Sabău, C. Oară, S. Warnick, and A. Jadbabaie, "Optimal distributed control for platooning via sparse coprime factorizations," *IEEE Trans. Auto. Control*, vol. 62, no. 1, pp. 305–320, Jan. 2017.

[101] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded vnf chains in 5g core networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 692–704, Feb. 2019.

[102] "ETSI GR MEC 017 V1.1.1 mobile edge computing (MEC); deployment of mobile edge computing in an NFV environment," tech. rep., European Telecommunications Standards Institute, Feb. 2018.

[103] "ETSI GS NFV-EVE 005 V1.1.1 (2015-12)," Dec. 2015.

[104] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, "Air-ground integrated mobile edge networks: Architecture, challenges and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26 – 32, Aug. 2018.

[105] Q. Ye, W. Zhuang, S. Zhang, A. Jin, X. Shen, and X. Li, "Dynamic radio resource slicing for a two-tier heterogeneous wireless network," *IEEE Trans. Veh. Tech.*, vol. 67, no. 10, pp. 9896 – 9910, Oct. 2018.

[106] Z. Su, Y. Wang, Q. Xu, M. Fei, Y.-C. Tian, and N. Zhang, "A secure charging scheme for electric vehicles with smart communities in energy blockchain," *IEEE Internet Things*, vol. 6, no. 3, pp. 4601–613, June 2019.

[107] W. Quan, K. Wang, Y. Liu, N. Cheng, H. Zhang, and X. Shen, "Software-defined collaborative offloading for heterogeneous vehicular networks," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–9, Apr. 2018.

[108] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, 2013.

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

[110] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: a survey and extensions," *Math. method oper. res.*, vol. 66, no. 3, pp. 373–407, June 2007.

[111] C. Guo, C. Shen, Q. Li, J. Tan, S. Liu, X. Kan, and Z. Liu, "A fast-converging iterative method based on weighted feedback for multi-distance phase retrieval," *Sci. rep-UK*, vol. 8, pp. 1–10, Apr. 2018.

[112] M. A. Lema, A. Laya, T. Mahmoodi, M. Cuevas, J. Sachs, J. Markendahl, and M. Dohler, "Business case and technology analysis for 5G low latency applications," *IEEE Access*, vol. 5, pp. 5917–5935, Apr. 2017.

[113] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Network*, vol. 32, no. 1, pp. 80–86, Jan. 2018.

[114] G. Qiao, S. Leng, K. Zhang, and Y. He, "Collaborative task offloading in vehicular edge multi-access networks," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 48–54, Aug. 2018.

[115] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

[116] J. Wang, D. Feng, S. Zhang, J. Tang, and T. Q. Quek, "Computation offloading for mobile edge computing enabled vehicular networks," *IEEE Access*, vol. 7, pp. 62624–62632, May 2019.

[117] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11158–11168, Nov. 2019.

[118] H. Li, K. Ota, and M. Dong, "Deep reinforcement scheduling for mobile crowdsensing in fog computing," *ACM Trans. Internet Technol.*, vol. 19, no. 2, pp. 1–18, Apr. 2019.

[119] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, Feb. 2020.

[120] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[121] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Tech.*, vol. 66, no. 11, pp. 10433–10445, Nov. 2017.

[122] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning-based mode selection and resource management for green fog radio access networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1960–1971, Apr. 2019.

[123] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[124] L. Kuang, X. Chen, C. Jiang, H. Zhang, and S. Wu, "Radio resource management in future terrestrial-satellite communication networks," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 81–87, Oct. 2017.

[125] L. Li, K. Ota, and M. Dong, "Humanlike driving: empirical decision-making system for autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 6814–6823, Aug. 2018.

[126] Q. Ye, W. Zhuang, S. Zhang, A. Jin, X. Shen, and X. Li, "Dynamic radio resource slicing for a two-tier heterogeneous wireless network," *IEEE Trans. Veh. Tech.*, vol. 67, no. 10, pp. 9896–9910, Oct. 2018.

[127] M. L. Puterman, *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[128] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, Fourthquarter 2019.

[129] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[130] I. Grondman, L. Busoniu, G. A. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[131] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor-critic algorithms," *Automatica*, vol. 45, no. 11, pp. 2471–2482, 2009.

[132] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," in *Adv. Neur. In.*, pp. 3675–3683, 2016.

[133] D. Zeng, L. Gu, S. Pan, J. Cai, and S. Guo, "Resource management at the network edge: A deep reinforcement learning approach," *IEEE Network*, vol. 33, no. 3, pp. 26–33, May/June 2019.

[134] "PTV vissim." `https://en.wikipedia.org/wiki/PTV_VISSIM/`. Accessed Feb., 2021.

[135] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," 2019.

[136] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, Aug. 2018.

[137] H. K. Ke, J. Wang, L. Deng, Y. Ge, and H. Wang, "Deep reinforcement learning-based adaptive computation offloading for MEC in heterogeneous vehicular networks," *IEEE Trans. Veh. Tech.*, vol. 69, no. 7, pp. 7916–7929, July 2019.

[138] G. Faraci, C. Grasso, and G. Schembra, "Design of a 5G network slice extension with MEC UAVs managed with reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2356–2371, Oct. 2020.

[139] H. Peng and X. Shen, "DDPG-based resource management for MEC/UAV-assisted vehicular networks," in *Proc. IEEE VTC 2020 Fall*, pp. 1–6, Oct. 2020.

[140] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[141] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.

[142] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.

[143] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent Actor-Critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 6379–6390, 2017.

[144] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Tech.*, vol. 1, pp. 45–66, Jan. 2020.

[145] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. ICML*, vol. 99, pp. 278–287, 1999.

[146] Bonsai, "Deep reinforcement learning models: Tips tricks for writing reward functions." `https://medium.com/@BonsaiAI/deep-reinforcement-learning-models-tips-tricks-for-writing-reward-functions-a84fe525e8e0/`. Accessed June 26, 2020.

[147] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proc. ICLR,* San Juan, Puerto Rico, 2016.

[148] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 2137–2145, 2016.

[149] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takác, "Reinforcement learning for solving the vehicle routing problem," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 9839–9849, 2018.

List of Publications

Journal Papers

[J1] **Haixia Peng**, Huaqing Wu, and Xuemin Shen, "Edge Intelligence for Multi-Dimensional Resource Management in Aerial-Assisted Vehicular Networks," submitted to IEEE Wireless Communication Magazine.

[J2] **Haixia Peng** and Xuemin Shen, "Agent Reinforcement Learning based Resource Management in MEC- and UAV-Assisted Vehicular Networks," IEEE Journal on Selected Areas in Communications - Series on Machine Learning in Communications and Networks, vol. 39, no. 1, pp. 131-141, Jan. 2021.

[J3] **Haixia Peng** and Xuemin Shen, "Deep Reinforcement Learning based Resource Management for Multi-Access Edge Computing in Vehicular Networks," IEEE Transaction on Network Science and Engineering, vol. 7, no. 4, pp. 2416-2428, Dec. 2020.

[J4] **Haixia Peng**, Qiang Ye, and Xuemin Shen, "Spectrum Resource Management for Multi-Access Edge Computing in Autonomous Vehicular Networks," IEEE Transaction on Intelligent Transportation Systems, vol. 21, no. 7, pp. 3001-3012, Jul. 2020.

[J5] **Haixia Peng**, Qiang Ye, and Xuemin Shen, "SDN-based Resource Management for Autonomous Vehicular Networks: A Multi-Access Edge Computing Approach," IEEE Wireless Communication Magazine, vol. 26, no. 4, pp. 156-162, Aug. 2019.

[J6] **Haixia Peng**, Le Liang, Xuemin Shen, and Geoffrey Ye Li, "Vehicular Communications: A Network Layer Perspective," IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1064–1078, Feb. 2019.

[J7] Weiting Zhang, Dong Yang, **Haixia Peng**, Wen Wu, Wei Quan, Hongke Zhang, and Xuemin Shen, "Deep Reinforcement Learning Based Resource Management for DNN Inference in Industrial IoT," accepted by IEEE Transactions on Vehicular Technology.

[J8] Weiting Zhang, Dong Yang, Wen Wu, **Haixia Peng**, Ning Zhang, Hongke Zhang, and Xuemin Shen, "Optimizing Federated Learning in Distributed Industrial IoT: A Multi-Agent Approach," submitted to IEEE Journal on Selected Areas in Communications - Special Issue on Distributed Learning over Wireless Edge Networks.

[J9] Le Liang, **Haixia Peng**, Geoffrey Ye Li, and Xuemin Shen, "Vehicular Communications: A Physical Layer Perspective," IEEE Transactions on Vehicular Technology, vol. 66, no. 12, pp. 10647-10659, Dec. 2017.

Conference Papers

[C1] **Haixia Peng** and Xuemin Shen, "DDPG-based Resource Management for MEC/UAV-Assisted Vehicular Networks," Proc. IEEE VTC 2020-Fall, Victoria, Canada, Oct. 4-7, 2020.

[C2] Dairu Han, **Haixia Peng**, Huaqing Wu, Wenhe Liao, and Xuemin Shen, "Joint Cache Placement and Content Delivery in Satellite-Terrestrial Integrated C-RANs," Proc. IEEE ICC'21, Virtual / Montreal, Canada, June 14-23, 2021.

[C3] Weiting Zhang, Dong Yang, Wen Wu, **Haixia Peng**, Hongke Zhang, and Xuemin Shen, "Resource Management for Federated Learning in Distributed Industrial IoT Networks," submitted to Proc. IEEE ICC'21 workshop.

[C4] Weiting Zhang, Dong Yang, **Haixia Peng**, Wen Wu, Wei Quan, Hongke Zhang, and Xuemin Shen, "Reinforcement Learning Based Resource Management for DNN Inference in IIoT," Proc. IEEE Globecom'20, Taipei, Taiwan, Dec. 7-11, 2020.

[C5] Haifeng Sun, Jun Wang, **Haixia Peng**, Lili Song, and Mingwei Qin, "Delay Constraint Energy Efficient Cooperative offloading in MEC for IoT," Proc. EAI CollaborateCom'20, Cyberspace, Oct. 16-18, 2020.