Towards Smart Vehicular Environments via Deep Learning and Emerging Technologies

Ahmed Al-Hilo

A Thesis

in

The Department

of

Information and Systems Engineering

Presented in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy (Information and Systems Engineering) at

Concordia University

Montréal, Québec, Canada

June 2021

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Ahmed Al-Hilo**

Entitled: **Towards Smart Vehicular Environments via Deep Learning and Emerging Technologies**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Youmin Zhang

_____ External Examiner
Dr. Abdallah Shami

_____ External to Program
Dr. Dongyu Qiu

_____ Examiner
Dr. Roch Glitho

_____ Examiner
Dr. Jamal Bentahar

_____ Supervisor
Dr. Chadi Assi

_____ Co-supervisor
Dr. Sanaa Sharafeddine

Approved by _____
Dr. Abdessamad Ben Hamza, Chair
Department of Information and Systems Engineering

June, 8, 2021

_____
Dr. Mourad Debbabi, Interim Dean,
Gina Cody School of Engineering and Computer Science

Abstract

Towards Smart Vehicular Environments via Deep Learning and Emerging Technologies

Ahmed Al-Hilo, Ph.D.

Concordia University, 2021

Intelligent Transportation Systems (ITS) embrace smart vehicular environments through a fully connected paradigm known as vehicular networks. Vehicular networks allow automobiles to stay online and connected with their surroundings while travelling. In that sense, vehicular networks enable various activities; for example, autonomous driving, road surveillance, data collection, content delivery, and many others. This leads to more efficient, safer, and comfort driving experiences and opens up new opportunities for many business sectors. As such, the networking industry and academia have shown great interests in advancing vehicular networks and leveraging relevant services.

In this dissertation, several vehicular network problems are addressed along with proposing novel ideas and utilizing effective solutions. As opposed to stationary or slow moving communications, vehicular networks experience more challenging environment as a result of vehicle mobility. Consequently, vehicular networks suffer from ever-changing topology, short contact times, and intractable propagation environments. In particular, this dissertation presents six works that participate in supplementing the literature as follows. First, a content delivery framework in the context of vehicular network is studied where digital contents are generated by different content providers (CP) and have distinct values. To this end, a prefetching technique along with vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communications are used to enable fast content delivery. Furthermore, a pricing model is proposed to deal with contents' values to attain a satisfactory Quality of Experience (QoE). Second, a more advanced system model is discussed to cache contents with the assistance of vehicles and to enable a disconnected and fixed Road-Side

Unit (RSU) to participate in providing content delivery services. The changing popularity of contents is investigated besides accounting for the limited RSU cache capabilities. Third, the stationary RSU proposed in the second work is replaced by a more flexible infrastructure, namely an aerial RSU mounted on an unmanned aerial vehicle (UAV). The mobility of the UAV and its constrained energy capacity are analyzed and Deep Reinforcement Learning is incorporated to aid in solving the challenges in leveraging UAVs. Fourth, the previous two studies are integrated by investigating the collaboration between a UAV and terrestrial RSUs in delivering large-size contents. A strategy to fill up the UAV cache is also suggested via mulling contents over vehicles. Fifth, the complexity of vehicular urban environments is addressed. In particular, the problem of disconnected areas in vehicular environments due to the appearance of high-rise buildings and other obstacles is studied. In details, a Reconfigurable Intelligent Surface (RIS) is exploited to provide indirect links between the RSU and vehicles travelling through such areas. Our sixth and final contribution deals with time-constrained Internet of Things (IoT) devices (IoTD) supporting ITS networks. In this regard, a UAV is dispatched to collect their data timely and fully while being assisted by a RIS to improve the wireless channel quality. In the end, this dissertation provides discussions that highlight open research directions worth of further investigations.

Acknowledgments

I would like to express my sincere gratitude to those who supported, guided, and motivated me throughout my PhD program. First and foremost, my biggest thank goes to my supervisors Dr. Chadi Assi and Dr. Sanaa Sharafeddine for their unconditional availability, endless guidance, and exemplary supervision during my PhD journey.

I am also thankful to the PhD committee members: Dr. Roch Glitho, Dr. Dongyu Qiu, and Dr. Jamal Bentahar. Their valuable and constructive feedback and comments were appreciably fruitful to better shape my works. Furthermore, I thank Dr. Abdallah Shami for accepting to serve as a delegate in my Ph.D. thesis examining committee. It is a pleasure to have you serving as a delegate in my committee.

Furthermore, I would like to thank Dr. Dariush Ibrahimi for his immense endorsement and insightful advises that helped me a lot to push my works forward.

Next, I extend special thanks to my colleagues and labmates here in Montreal and in my hometown. Thank you for the valuable remarks and worthy discussions. On this occasion, I cannot forget my colleague, Dr. Moataz Samir, who helped me a lot during my study trip. I would like also to address Mohamed Elhattab and thank him for the significant and fruitful cooperation I had with him during the second half of my PhD program.

Last but not least, I would like to thank my family. Here, I address my parents, without their support, I would not have been able to continue my studies. Thank you from my deepest heart and I hope that I have achieved your dreams. In addition, I also thank my brother, sisters, and aunt for their unconditional love and assistance.

"Anyone who has never made a mistake has never tried anything new."

–Albert Einstein–

Contents

List of Figures

List of Tables

List of Abbreviations

3GPP	3rd Generation Partnership Project
5G	Fifth-Generation
6G	Sixth-Generation
AC	Actor-Critic
AI	Artificial-Intelligence
B5G	Beyond Fifth-Generation
BRB	Branch, Reduce and Bound
BS	Base Station
C-V2X	Cellular Vehicle to Everything
CAPEX	Capital Expenditure
CPLEX	IBM ILOG CPLEX Optimization Studio
CP	Content Provider
CSI	Channel State Information
DDPG	Deep Deterministic Policy Gradient
DP	Dynamic Programming
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
eMBB	Enhanced Mobile Broadband

IoCV	Internet of Connected Vehicles
IoE	Internet of Everything
IoTDs	Internet of Things Devices
IoT	Internet of Things
ITS	Intelligent Transportation System
LiDARs	Light Detection and Ranging
LoS	Line of Sight
LP	Linear Programming
MDP	Markov Decision Process
MINLP	Mixed Integer Non Linear Program
ML	Machine Learning
mMTC	Massive Machine Type Communications
mmWave	Millimeter Wave
NLoS	Non Line of Sight
NN	Neural Network
NOMA	Non Orthogonal Multiple Access
OPEX	Operational Expenditure
PPO	Proximal Policy Optimization
QoS	Quality of Service
ReLU	Rectified Linear Unit
RIS	Reconfigurable Intelligent Surface
RSUs	Road Side Units
SCA	Successive Convex Approximation
SNR	Signal to Noise Ratio

SON	Self Organized Networks
SUMO	Simulation of Urban MObility
TDMA	Time Division Multiple Access
TRPO	Trust Region Policy Optimization
UAVs	Unmanned Aerial Vehicles
URLLC	Ultra Reliable Low Latency Communications
VR/AR	Virtual Reality and Augmented Reality

Chapter 1

Introduction

1.1 ITS Era in Future Wireless Networks

Future cellular network generations promise to foster revolutionary services, provide higher data rates, allow massive connectivity, and exchange huge traffic volumes. In terms of services, three leading-edge use case domains; enhanced mobile broadband (eMBB), ultra reliable low latency communications (URLLC), and massive machine type communications (mMTC). Under these domains exists a wide spectrum of fundamental services including, Extended Reality (XR), remote healthcare, smart homes, industry automation, to name a few. Moreover, future cellular networks involve various stakeholders and end-users, from fixed or slow moving entities such as smart phone users and industrial Internet of Things (IIoT) devices to highly mobile users associated with vehicular networks. Additionally, 5G and beyond technologies are forecast to enable a set of unprecedented services and applications related to the Intelligent Transportation Systems (ITS). ITS is a combination of advanced technologies that aim to make driving experiences more efficient, safer, and more convenient [1]. It is rapidly evolving to provide innovative services related to traffic management, road safety, infotainment applications and smart highways. For example, fully autonomous vehicles are expected to account for up to 66% of the passenger-kilometers traveled in 2040 [2, 3]. With reduced to non-existent human control of vehicles, the automotive industry is transforming the vehicle interior to an entertainment space supporting various services.

In general, two classes of technologies are identified for ITS; safety- and non-safety related.

Safety-related applications encompass traffic management, autonomous driving, cooperative maneuvering, safety message dissemination, and many others. While non-safety applications include a set of infotainment services that are consumed by on-road users such as 4K video streaming, video games, social media browsing, etc. Indeed, the dominance of video-based services spurs a paramount growth of traffic volume that places a significant load on the existing network infrastructure. Although, Road-Side Units (RSU) emerged to augment the capability of cellular networks and typically operate on selected spectrum bands and have limited coverage, the major issue of the lack of, or a highly congested, backhaul link is still unsolved [4, 5]. All in all, ITS applications and services desire extra low latency, ultra high reliability, high speed, and massive connectivity in order to satisfy the market demands. To answer these demands, vehicular communications have emerged as an appealing technology to connect vehicles with each other via vehicle to vehicle (V2V) and with remote entities via vehicle to infrastructure (V2I) [6].

To enable vehicular communications, Wireless Access in Vehicular Environments (WAVE) standard is used to specify the communication protocol and frequencies used in this kind of communications which is governed by dedicated short-range communications (DSRC). DSRC is a wireless channel for medium range communications that allows one- and two-way transmissions and is commonly used for vehicular networks. Apart from DSRC, the community of 3GPP suggested cellular vehicle to everything (C-V2X) to integrate vehicular communications with the cellular networks.

Vehicular communication is proposed as a proper medium to connect vehicles with their environments that contain miscellaneous actors. As demonstrated in Fig. 1.1, vehicular environments encompass many kinds of various entities where vehicles can communicate with, i.e., RSUs, base stations (BS), etc. In addition, establishing a decent communication platform may require assistance in case of vehicular networks where enormous numbers of objects appear such as buildings. Hence, introducing reconfigurable intelligent surface (RIS) to vehicular networks is appealing to enhance wireless communications. Given its ability to control and manipulate wireless environments, RIS has emerged as a key enabler technology for the six-generation (6G) cellular networks. RIS is an array composed of a number of passive low-cost elements, each of which has the ability to independently tune the phase-shift of the incident radio waves. Furthermore, UAVs are also suggested to aid vehicular networks owing to their mobility and flexibility which can significantly enhance the

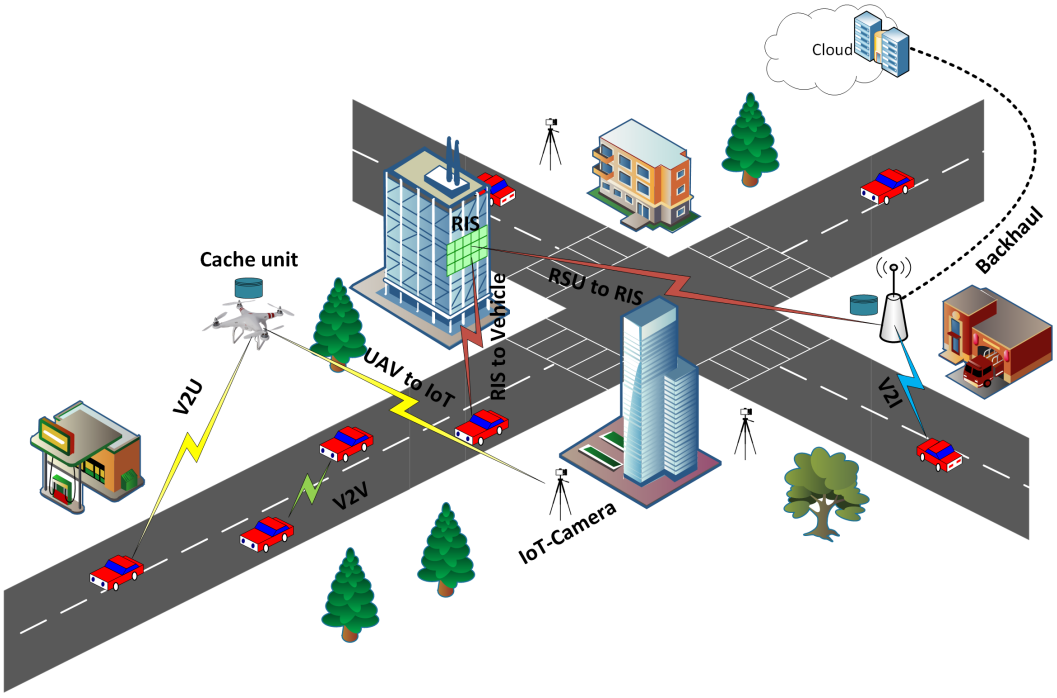


Figure 1.1: The vehicular network.

communications besides provisioning services for on-road users.

In this dissertation, we complement the literature by studying six scenarios where vehicular communications and services are notably enhanced. In order to do that, we study a group of technologies whose development and practical applications have not been largely realized and are commonly known as emerging technologies (such as Internet of Vehicles, UAVs, RIS, Artificial Intelligence, IoT, etc [7–9]). Dealing with such technologies imposes several challenges and concerns. The rest of this chapter discusses the principle issues raised by vehicular networks while considering various technologies' employment. Then, we provide a list of the limitations in the literature that we are going to cover. Finally, we briefly go over our contributions and provide a detailed organization for this dissertation.

1.2 Vehicular Network Challenges

As opposed to stationary or slow moving nodes, vehicular networks suffer from highly challenging environment, changing topology, and challenging propagation environments.

- **Ever-changing topology and road conditions:** Due to their high mobility, vehicles have limited connection times with external entities including, nearby vehicles and surrounding infrastructures. On the other hand, RSUs operated by WAVE protocols have limited coverage area [10]. Therefore, it is in many instances impossible to complete the transmission processes within one contact time especially if the wireless resources are not handled properly. Establishing decent wireless communication links for vehicles depends highly on the trajectory of vehicles, which is, in turn, specified by the vehicles variant speeds and directions. In the context of highways, the quality of V2V is determined based on the contact time between vehicles. Likewise, the ability of an infrastructure, i.e., RSU or UAV, to serve vehicles is influenced by the vehicles' residence times and distance between the vehicles and the RSU. On the other hand, the state of roads are inconstant, instead, there is peak and non-peak hours. For example, early morning, employees and students go to work, thus, the road density is high. In contrast, after midnight, the roads are almost empty. In addition, roads that lead to different destinations, have different road conditions based on the traffic and the characteristics of that region. In the context of vehicular networks, road density is one of the aspects that specifies the number of demands for contents and services. A high road traffic reflects larger numbers of requests and vise versa. Thus, a good planner for vehicular networks has to account for various road conditions. It is worth-mentioning that the techniques used to improve vehicular networks during dense road conditions are different from low- or mid-traffic events.

- **Content popularity and caching:** Today, the cloud offers enormous amounts of contents that have different features, requirements, and request patterns. On the other hands, people have different tastes towards contents which might be determined based on the culture, age, gender, or even the location. For example, people living in a particular region have different requests than people residing in other regions. In vehicular networks, this issue becomes much more complicated due to the mobility nature of vehicles as vehicles travel to different and perhaps unknown destinations. Thereby, determining content popularity profiles in vehicular networks is exceptionally strenuous. To further compound the problem, a recent study by Intel and Ovum reveals that video will account up to 90% of 5G traffic in 2028 [11]. Due to

the growing interest in video-based services, whether gaming, movies, and other infotainment applications, content providers strive to deliver high quality resolutions for premium customer experience. Consequently, the next cellular generation will have to provide seamless service for big content such as 4K videos to satisfy the market demands. One solution is to cache video contents at the edge, however, owing to their massive sizes, such contents impose heavy burdens on the edge cache units.

- **Vehicular environment complexity:** In vehicular networks, large objects such as trucks and high-rise buildings often appear. Thereby, the Line-of-Sight (LoS) between the transmitters and vehicles is not always clear. Sometimes, the connection link quality falls bellow the intended threshold and the transmission processes become void. In this case, Reconfigurable Intelligent Surfaces (RIS) are employed acting as passive relays between the transmitter and destination. However, RIS deployment creates another problem owing to its hard phase-shift configuration [12]. In addition, determining phase-shift configuration for mobile users, such as vehicles, is very challenging.

- **Emerging technologies and ITS integration:** Setting up ITS in reality may encompass various kinds of unprecedented technologies that inherit their own challenges. For example, UAVs are commonly used in the literature as a key enabling technology for vehicular networks owing to their mobility that can cope with the dynamic nature of such environment. However, UAVs also suffer from a number of limitations, i.e., half-hour battery lifetime, that are strongly influenced by UAV velocity and trajectory [13]. Additional issue that hinders a UAV deployment for wireless communications is that the difficulty in maintaining strong backhaul connections with the core network while persistently flying to catch up with vehicles. Another enabling technology for ITS is IoT; IoT devices (IoTD) will be used largely in ITS to enable various applications and services [14]. However, due to their restricted capabilities, i.e., short battery lifetime and weak wireless transmissions, it is always challenging to collect the data generated by such devices. IoTDs are battery powered, hence, they are more inclined to save their power by alternating between sleep mode and active mode [15]. Where the IoTDs can only communicate with external entities when they are active. In addition, the

problem exacerbates when it comes to the sparsity distribution of these devices in urban areas which makes the communication with such devices very difficult.

1.3 Limitations of Existing Works

Despite there have been significant efforts devoted to enhance wireless communications and related services in vehicular environments, there are still several shortages in the literature. The following summarizes the existing gaps that are going to be covered by this dissertation:

- **Content providers with distinct QoE values:** Providing digital contents is one of the most demanded services not only for vehicular users but at the level of cellular networks in general. For example, the demands for video contents constitutes 90% of the entire cellular traffic volume. In the meantime, contents from different providers have various requirements. Nowadays, some platforms such as YouTube and Facebook offer free video streaming, yet, another class of content providers offer premium subscription plans for their costumers, i.e., Netflix and Amazon. Logically speaking, the customers of the latter type of content providers urge for better QoE for their users; for example, low latency and seamless video streaming. However, such problem has not been studied before in the context of vehicular networks.

- **Limited backhaul availability:** Deploying an infrastructure such as RSU or dispatching a UAV does not necessarily improve content experiences unless coupled with a solution to fetch the contents in the first place. RSUs and UAVs are deployed for various purposes including for safety and non-safety related applications; sometimes with computational capabilities to assist or enable vehicular networks. If the deployment happens to be done in an area where there is no infrastructure available or the operator desires to minimize the costs, a link to the backhaul may not be available. The RSU can then leverage the cached contents on the passing by vehicles to populate its cache in a cost-effective manner. In such a way, the RSU will be able to fill up its cache unit. However, that raises several challenges corresponding to the dynamic popularity of contents, vehicle mobility, and the incentives paid to the cooperative vehicles.

- **Massive contents and cooperative infrastructures:** One of the key problematic issue in vehicular networks is how large-size contents can be served to vehicles. As mentioned earlier, vehicles have limited contact times with remote infrastructures. Consequently, large contents often cannot be served via one infrastructure. Meanwhile, due to their high costs of deployment, it is quite undesirable to deploy a large number of RSUs on the roads. An alternate solution is realised via UAVs. Where UAVs mounted with cache capabilities are able to collaborate with the RSU operator in order to work as a supplement base station that continues serving vehicles after leaving the coverage area of the RSU. In this context, several concerns may arise owing to the UAV mobility, cache management, and resource scheduling among the RSU and the dispatched UAV.

- **Dead urban zones:** Although RSUs and UAVs are proposed as promising solutions to provide wide coverage for vehicles, there are still some areas in vehicular environments suffer from lack of connectivity due to blocked communication links caused by high buildings, big trucks, trees, and other objects that appear in urban areas. Thankfully, beyond 5G era presents one of the promising technologies that can cope with such problem. Namely, RIS is capable of manipulating wireless environments and eventually provides an indirect connection between a source and destination. Hence, introducing RIS to vehicular networks while considering the highly dynamic nature of the environment is a worthwhile topic to investigate.

- **Scattered IoT devices to enable ITS:** ITS depends on several emerging technologies, among them, IoT stands as a strong candidate to support various purposes. Road condition monitoring, traffic light status, parking lots, and autonomous driving, are all examples of applications that incorporate IoT principles. Owing to their sparse distribution in urban areas, UAV has been widely proposed to support IoT networks due to its flexibility to cope with such environment. However, both UAV and IoT devices suffer from several limitations related to their limited energy budget and short coverage. Meanwhile, the role that RIS can play in mitigating such issues has not yet addressed.

In light of the aforementioned limitations of existing works, this dissertation proposes six main contributions to bridge the respective gaps in realizing intelligent transportation systems.

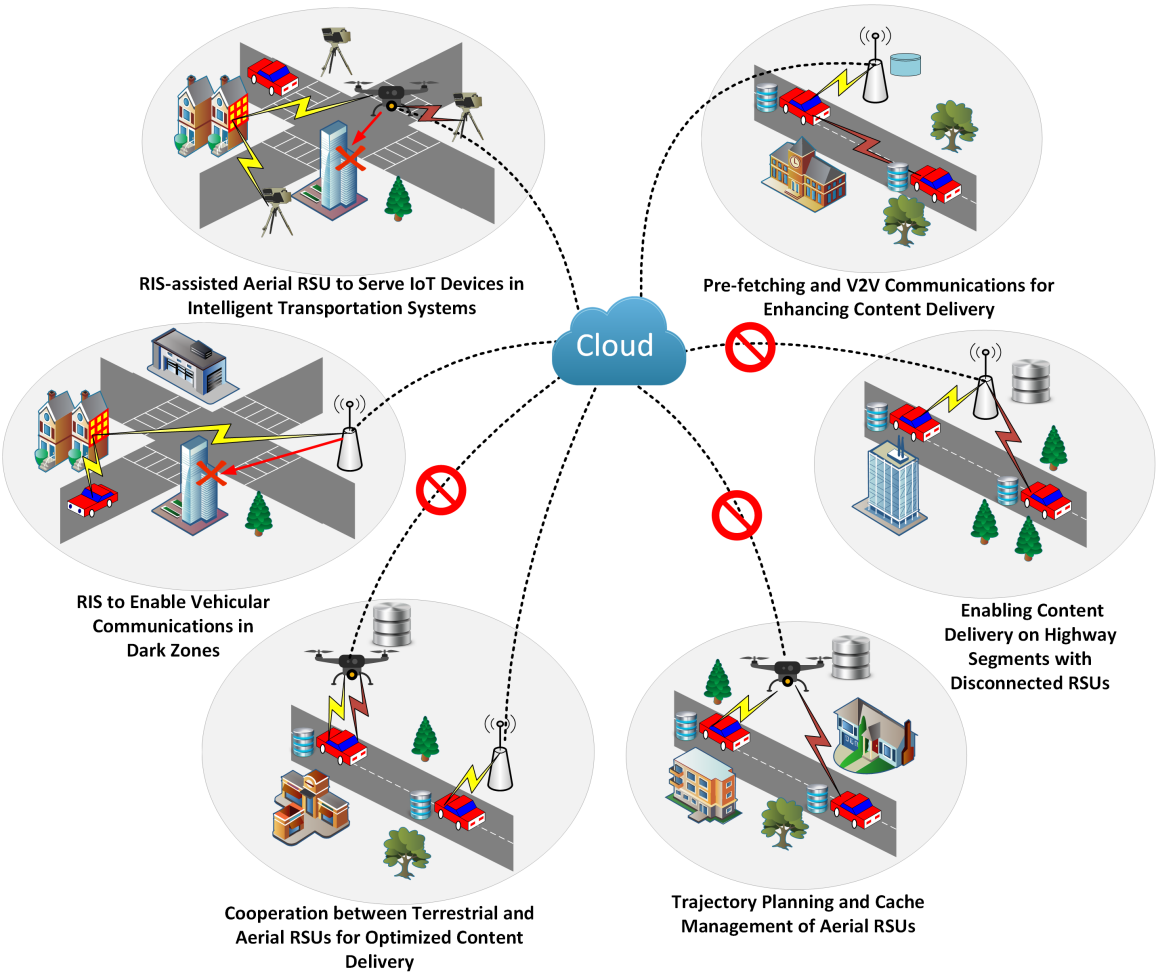


Figure 1.2: PhD thesis contributions overview.

1.4 Thesis Contributions

Our aim in this dissertation is to bridge the gap for the limitations presented in Section 1.3. Consequently, we contribute six works in improving content delivery and facilitate wireless link establishment in vehicular networks as depicted in Fig. 1.2. We consider various scenarios and leverage several appealing technologies including terrestrial and aerial RSUs, and connected and standalone infrastructures.

1.4.1 Pre-fetching and V2V Communications for Enhancing Content Delivery

In traditional highway scenarios of vehicular networks where digital contents are served, several RSUs are deployed alongside the road. Due to their limited range, there is a dark area between each two adjacent RSUs. The RSU operators, intuitively, are interested in increasing their revenue by providing improved services to the content providers' customers. In the context of entertainment services, there are several content providers having different requirements for their customers based on the quality of their service. For example, nowadays, some platforms broadcast their content in free-on-air mode while others accept only subscription plans for a predefined cost which varies from one provider to another. Also, in order to provide seamless service and avoid stall events or delays, the RSU operator prefetches enough chunks of contents that can be served within the vehicles residence time. The content chunks buffered on the RSU will later be served to the vehicles. The served chunks must be enough to playback until the vehicle reaches the next RSU.

Due to the limited period of time that a vehicle can stay within the RSU range, it is sometimes impossible to serve enough content to playback without interruptions. Hence, the RSU operator might also leverage V2V communication in a way that can relay some chunks through other vehicles. Another important factor that the RSU operator can exploit is the similarity among contents from similar content providers. For example, content providers might broadcast similar advertisement to all their customers. Such practice allows the RSU to make use of the V2V communications by serving these overlap chunks to only one vehicle while letting the the other ones receive these parts from that vehicle. The aforementioned scenarios are formulated mathematically to increase RSU operator revenue taking into account V2V and V2I scheduling as well as the values of contents and overlap contents. Then, three algorithms are proposed to solve the formulated problems with less complexity. The optimal solutions are also obtained via CPLEX for small instances of the problems.

1.4.2 Enabling Content Delivery on Highway Segments with Disconnected RSUs

In typical wireless networks, back-hauling constitutes one of the main bottleneck in the network. Every packet requested by a node has to be first fetched from the backhaul before it becomes ready

to be transmitted to the end user/node. When it comes to 5G and beyond, it is highly forecast that traffic volumes will exponentially increase because of the emergence of new technologies such as 4K streaming and Augmented Reality and Virtual Reality (AR/VR). This fact is more concerning in vehicular networks where RSUs are linked up with weak backhauls which can become congested during peak hours. Add to that, some cost-effective RSU are placed in remote and disconnected area for safety purposes, i.e., message dissemination. In such scenario, the RSU will not be able to provide content delivery services for vehicles unless a way to fetch contents is designed.

In contrary, in-vehicle caching (IV-Cache) is a another concept that allows vehicles to store and share contents among them. That is, a vehicle is equipped with a memory that can save the contents once they are downloaded. One can imagine that such contents can be harvested by the RSU once the vehicles become within the range of the RSU. However, as the RSU has a finite cache unit, it can only store a limited number of contents. Thus, it is better to select the most popular ones. Then, the RSU can serve vehicles which request one of these cached contents. In order to address the challenges of the problem laid out above, we resort to Deep Q Networks (DQN) to decide which vehicle to fetch its content. Also, an efficient method is designed to schedule downlink to serve vehicles.

1.4.3 Trajectory Planning and Cache Management of Aerial RSUs

An appealing upgrade for the system proposed in 1.4.2 can be realized through switching from terrestrial to aerial RSU, for example UAV. UAVs have some advantages over fixed RSUs in vehicular networks. Owing to their mobility and flexibility, UAVs can better cope with the dynamic nature of vehicular environments. Namely, UAVs can catch up with vehicles in need for service and expand the communication time until the service is fully provided. Additionally, UAVs can provide a cost-effective solution; thanks to their deployability, UAVs can be dispatched only when needed, for instance, during peak hours. In contrast, UAVs suffer from some limitations. First, UAVs have constrained battery capacity, therefore, they can only operate for a limited time. Further, UAV mobility incurs varying amounts of energy consumption based on the UAV velocity. Second, due to their mobility, it is challenging to maintain a good fronthaul link.

Therefore, in our third contribution, we propose a UAV equipped with cache units and is being

employed to serve vehicles on a highway. The objective is to maximize the energy efficiency of the UAV by trying to maximize the content delivery service while optimizing the energy consumption. This is done through an intelligent agent based on DRL that plans for the UAV trajectory. Namely, proximal policy optimization (PPO) is leveraged to decide the UAV trajectory while two algorithms have also been proposed to effectively tackle the problems of uplink and downlink between the UAV and vehicles.

1.4.4 Cooperation between Terrestrial and Aerial RSUs for Optimized Content Delivery

The demand for digital contents and advanced streaming services is experiencing unprecedented growth lately [16–18]. Hence, one RSU might not be able to cope with the massive demands for contents. Further, due to the short residence time of vehicles within the communication range of one RSU, it is highly possible that the RSU can only serve contents in part. Therefore, building upon our previous contributions, we further extend our system models by considering two types of infrastructure; aerial RSU (mounted on a UAV) and terrestrial RSU. The RSU is assumed to have a stable access to the cloud while the UAV has not. Hence, one way to enable the UAV to participate in serving contents to vehicles is through relaying contents from the RSU to the UAV via the vehicles themselves. This is done by push-carry-relay fashion where the RSU decides which contents to relay to the UAV and then pushes them on some vehicles.

In such context, several challenges arise. The first problem is to find the trajectory of the UAV to serve vehicles while fetching other content from the vehicles. The second problem is due to the wireless resource scheduling of the RSU. The third problem is due to the wireless resource scheduling of the UAV for the uplink and downlink. The fourth problem is due to the cache management of the UAV storage unit to replace existing contents once the cache unit is used up. Owing to the complexity of solving such problem, it is alternatively cast as an Markov decision process (MDP) whose solution is obtained through a Dual-Task Reinforcement Learning method to handle simultaneously RSU resource scheduling and UAV mobility.

1.4.5 Reconfigurable intelligent surfaces to Enable Vehicular Communications in Dark Zones

This contribution deals with more complicated scenario in vehicular networks where there is no clear LoS between the vehicles and RSU. Technically, due to the outdoor environment where high-rise buildings such as skyscraper appear, the transmitted signals will frequently experience bold distortions. Fortunately, a new emerging 6G-related technology is proposed to cope with such problem. Reconfigurable intelligent surfaces are a type of meta-surfaces that can receive and reflect transmission signals from the transmitter to the destination. To this end, one use case can be imagined in vehicular environment where the LoS between the RSU and one region/road is blocked by a particular object, say a building. In this case, we propose RIS to be deployed and situated on a building where it possesses LoS with the RSU and with the region of interest.

The objective of this work is to maximize the minimum average bit rate served to vehicles. This problem is formulated as a mixed integer non-convex program which is difficult to be solved. Also, there are some uncertainties owing to the dynamic nature of vehicular networks. Thereby, we resort to alternative methods based on DRL to determine RSU wireless scheduling and Block Coordinate Descent (BCD) to solve for the phase-shift matrix, i.e. passive beamforming, of the RIS. The Markov Decision Process (MDP) is defined and the complexity of the solution approach is discussed.

1.4.6 RIS-assisted Aerial RSU to Serve IoT Devices in Intelligent Transportation Systems

In our last contribution, we investigate into the scenario of data collection for IoT networks enabling ITS services. In details, a set of power-constrained IoT devices observe and generate time-constrained data spontaneously. A UAV is dispatched to collect these data timely (within the active periods of the IoT devices). However, due to some limitation related to the UAV small coverage area and limited-powered IoT devices, it is not always sufficient to rely solely on the UAV to provide high quality of service. Consequently, a RIS is proposed to improve the channel quality between the UAV and IoT devices. In such scenario, several challenges appear that associate with the UAV

trajectory, RIS phase-shift tuning, and IoT device scheduling. Hence, we cast the problem into two sub-problems where the first one is solved via DRL to control UAV trajectory and schedule the IoT devices. Then, RIS phase-shift is tackled based on BCD.

1.5 List of Publications

The research work of this dissertation produced 6 journal articles and one conference paper.

1.5.1 Journal Publications

(J1) Al-Hilo, Ahmed, et al. "Revenue-driven video delivery in vehicular networks with optimal resource scheduling." Elsevier Vehicular Communications, (2020).

(J2) Al-Hilo, Ahmed, et al. "Vehicle-Assisted RSU Caching Using Deep Reinforcement Learning." IEEE Transactions on Emerging Topics on Computing, (2021).

(J3) Al-Hilo, Ahmed, et al. "UAV-Assisted Content Delivery in Intelligent Transportation Systems-Joint Trajectory Planning and Cache Management." IEEE Transactions on Intelligent Transportation Systems, (2020).

(J4) Al-Hilo, Ahmed, et al. "A Cooperative Approach for Content Caching and Delivery in UAV-Assisted Vehicular Networks." Elsevier Vehicular Communications, Under Review, (2021).

(J5) Al-Hilo, Ahmed, et al. "Reconfigurable Intelligent Surface Enabled Vehicular Communication: Joint User Scheduling and Passive Beamforming." arXiv preprint arXiv:2101.12247 (2021).

(J6) Al-Hilo, Ahmed, et al. "RIS-Assisted UAV for Timely Data Collection in IoT Networks." arXiv preprint arXiv:2103.17162 (2021).

1.5.2 Conference and Workshop Publications

(C1) Al-Hilo, Ahmed, et al. "Cooperative content delivery in UAV-RSU assisted vehicular networks." Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond. 2020.

1.6 Organization of Thesis

The rest of this dissertation is organized as follows. In Chapter 2, we investigate a pricing model that allows different vehicle customers belong to different content providers to have particular QoE. As well as, the joint problem of resource allocation and content prefetching and scheduling while maximizing the revenues of the RSU operator is studied. Chapter 3 examines the ability of disconnected RSU to support content delivery services when equipped with cache capabilities. In Chapter 4, we investigate the benefits of leveraging aerial RSU (UAV) instead of a terrestrial one. Chapter 5 considers a scenario where stationary RSU collaborates with a UAV in serving contents to vehicles. The benefits of this collaboration is studied and a content transfer method to the UAV over vehicles is also proposed to facilitate this collaboration. Chapter 6 presents a system model based on RIS to provide wireless connectivity for vehicles residing in a dark zone. In particular, the communication link between the RSU and the vehicles is cut off due to the presence of high buildings. Chapter 7 proposes a UAV assisted by RIS to improve the wireless communication between the UAV and IoTDs enabling ITS. Here, the IoTDs are assumed to be alternating between two modes namely, active mode and sleep mode, where they can only communicate during the first mode. In the end, Chapter 8 concludes the dissertation and highlights a collection of open research directions for future consideration.

Chapter 2

Pre-fetching and V2V Communications for Enhancing Content Delivery

2.1 Introduction

In this chapter, we focus on video-based infotainment services (Video on demand) and propose optimized mechanisms for content prefetching and efficient management of Roadside Unit (RSU) resources to maximize the profit of RSU operators while guaranteeing high quality of experience with reduced costs. In a real case scenario, one can foresee that there are several RSUs providing intermittent coverage and vehicles arriving at different times and requesting contents from the cellular base station where the content providers pay the cellular operator in order to store their contents at the edge to minimize the latency. However, since cellular base stations might not be able to handle all the demands due to limited resources and huge traffic, RSUs are utilized to offload a great deal of requests partially or fully while maintaining a target quality level. Such a scenario expects RSUs to cooperate in serving the vehicles since one RSU is unable to satisfy all the vehicles' demands at the same time. In other words, each RSU should prefetch and provide enough video content for each vehicle to reach the next RSU without stall events and with less reliance on the cellular networks. However, the RSU should know in advance what content it can deliver to which vehicle such that it prefetches the desired content to avoid any backhaul delay. The management of this operation is challenging since vehicles arrive at different times with different speeds, thus, having

distinct residence periods. Moreover, vehicles request various kinds of video contents with different requirements. Management of this operation has, nonetheless, to account for increasing RSU's profit.

To handle the stringent requirements imposed by video traffic, we leverage cooperation between cellular and RSU networks as well as among vehicles. RSUs are provided with heads-up knowledge on the needed content so they prefetch and have the content ready upon vehicles' arrival. Moreover, the system designates vehicle(s) to relay content via V2V to requesting vehicles while the latter exit the RSU coverage. Many works have addressed content delivery in VANETs, however, to the best of our knowledge, no research considered the profit earned by efficient management of RSU resources to support content delivery services. Such scenario is expected to prevail in future communication networks. This can be in the form of Data Distribution Service (DDS), where the content providers pay to RSU operators in order to promote their contents and, subsequently, improve Quality of Experience (QoE) to their customers. Example services include, media-service (e.g. NetFlix), broadcasting and video sharing (e.g. YouTube), online music store (e.g. Amazon Music), and advertising agency (e.g. Google Adsense).

In this contribution, we show how RSUs can make profit from data delivery on one hand, and how content providers can improve their services on the other hand. Other work also explores different factors which can influence the revenue such as the impact of V2V communication to relay content between vehicles and how common content, e.g. advertisements, can improve the overall revenue [19, 20]. The main contributions of this work are:

(1) Addressing the joint problem of resource allocation, content prefetching, and scheduling of content delivery to vehicles while taking into account the set of vehicles within the RSU coverage, and requirements of requested content.

(2) Leveraging V2V communications and common content to generate more revenues and.

(3) Designing a business model that motivates both the RSUs and vehicles to deliver content.

(4) Studying and analyzing the the performance of the proposed approach as compared to the optimal solution and an alternative greedy approach.

The Chapter is structured as follows. Section 2.2 presents the related works, while section 2.3 explains our system model. In Section 2.4, we formulate our problem mathematically. Section 2.5 presents three algorithms that can solve our problem with less complexity. The numerical results and analysis are presented in section 2.6. Finally, section 2.7, summarizes the chapter.

2.2 Related Work

Data dissemination and content delivery have recently gained substantial momentum in the literature. We classify related research into three directions. First one deals with the gain and the second group works on resource allocation based on multi-channel. While the last one focuses on caching strategies to lower down backhaul utility which leads to high costs and delay and improve the QoS. However, to the best of our knowledge, work that concentrates on content delivery from the perspective of RSU revenue is very limited. [21] studies the utility of RSU to make profit by delivering advertisements. They focus only on the advertisement part and from the viewpoint of the advertisement agency (broker). In [22], RSUs compete against each other in order to attract the customers and sell their contents to maximize their revenues. [23] suggests a cooperation and competition among RSUs and parked vehicles to deliver contents. This competition is based on a pricing model where each party in the vehicular ad-hoc network tries to maximize its profit. [24] proposed a game model that helps to achieve adequate routing performance in vehicular networks by motivating vehicles to cooperate in order to improve their reputation. The reputation value can also be considered as a payment similarly to our system model. Yet, [25] also developed a distributed game-theoretical framework to improve the performance of the networks by using incentive to motivate nodes to cooperate together. They divided the nodes into two categories, sellers and buyers, where buyers leverage seller resources to improve their QoS. The work in this chapter is different from the existing literature in several ways. First, it works on vehicular network to find the optimal or sub-optimal resource allocation for RSU. Moreover, the proposed approach is broader and in the sense that it offloads video traffic from the cellular network while prioritizing content by maximizing RSU profit. The pricing model in this work considers content with different prices as opposed to existing literature that assumes a competition upon services by different plays such as the parked

vehicles, RSUs, ...etc.

2.3 System Model

We consider a multi-lane highway segment with cellular coverage and having several RSUs deployed alongside as depicted in Figure 2.1. Also, we assume each vehicle has a content request to stream from a content provider where the content are stored at the edge server near the cellular base station. We assume a collaborative environment between the cellular operator, the VANET network and the content provider where the objective is to deliver seamless experience to customers.

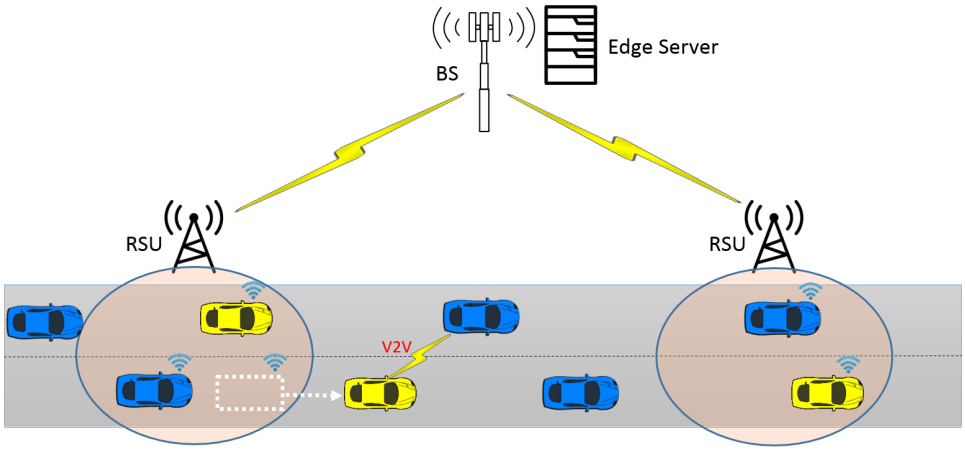


Figure 2.1: System Model.

2.3.1 Communication Model

We assume that RSUs use the 802.11p to communicate with the vehicles and the vehicles also use the 802.11p for V2V communication. The 802.11p uses the 5.9GHz radio spectrum and enables a multi-rate transmission (6-27Mbps) which varies according to the distance between a vehicle and the RSU, the channel quality, etc. In our work, we assume the coverage of the RSU is divided into several areas (S_0 to S_N), each has its own transmit bit rate.

Furthermore, we assume a time slotted system, i.e., vehicle i consumes N_i slots for its sojourn time in S0 while another vehicle with higher velocity will reside for fewer slots for the same distance. In addition, we assume there is a V2V communication between vehicles where the RSU

might exploit some vehicles in its range to relay content to other vehicles. The V2V bitrate is assumed to be constant and the proximity between the communicating vehicles must not exceed a certain threshold, otherwise, the link is said to be impaired. We also assume that at a certain time slot, a vehicle can communicate only with one other vehicle. The V2V communication should not be established inside the RSU coverage due to wireless interference.

In cases when the RSU fails to fully satisfy vehicle i due to resource limitations, vehicle i will request the rest of the needed content chunks from the cellular network. For example, if a vehicle requests 20Mb of content, however, the present RSU was only able to deliver 16Mb, which is not enough to play back until the end of the next RSU coverage. Thus, the rest, 4Mb, should be downloaded from the cellular network. Indeed, if the RSU does not make wise decisions in prefetching contents to serve passing by vehicles, the vehicles will fall back on the cellular, which will deliver the remaining bits.

2.3.2 Pricing Model

Content providers offer RSU operators higher pay to deliver high cost content as an incentive to prioritize content that generates more revenue. The pricing model we propose consists of two parts, revenue of the RSU and that of the relay vehicles. The latter is paid in return to utilizing vehicles' resources such as buffers and wireless connectivity to relay content to requesting vehicles via V2V. *The payments vehicles receive could be in the form of some credits or discount on using the RSU service.*

We assume each content provider pays the RSU operator a certain price for each data unit delivered to the vehicles through RSUs or relay vehicles. This price depends on the content provider; the more the content provider pays, the better QoS his customers have.

The RSU operator is assumed to scale the service prices in a number of groups, i.e., from 1 to 10 monetary unit for each specific amount of content delivered. In the previous example, if a content provider decides to pay 10 units, then his content(s) will have the highest priority. However, if he chooses to pay only 1 unit, then his customers will experience the lowest QoS among all others. The price list is not fixed, it could be shaped in other ways to provide larger gaps between the different groups of prices, for example, 1, 10, 20 ... etc. In this way, the priority difference between two

consecutive groups is 10 units.

2.3.3 System Operations

We assume that at a specific time slot t, there is a number of vehicles that are located at different positions within the coverage of a given RSU and each one has its own velocity, requested content, and a buffer that may have enough content to play back until the end of the present RSU. Since there is no caching, we assume each vehicle requests a unique content consisting of a number of bits. The RSU operator periodically receives control information from the cellular network on the upcoming vehicles in time to schedule its resources and prefetch the requested content. Moreover, the RSU operator schedules the V2V communications of the current vehicles within the coverage but cannot be rescheduled at a later time. Resources allocated to vehicles within the RSU's coverage may be rescheduled based on newly arriving vehicles and their requests.

The operator schedules the resources every epoch that is defined to be $N \bigtriangleup$ (N:number of time slots, $\bigtriangleup$:time slot length) by taking a snapshot of the current status, e.g., which vehicles are still within the RSU's range and those that have just arrived. Then, the operator gives each $i \in V_n$ a portion of the resources, where V_n is a set of vehicles taken into account in the nth epoch. Before i arrives, the operator calculates how much resources the RSU will have during i's residence time (T_i) and what is the best scheduling for the resources since the transmission rate is variable according to the distance from the RSU. The operator takes into account the value/price of the content.

Finally, since most of the content providers, nowadays, are broadcasting advertisements and common data such as navigation menus, movie list, icons, ...etc. embedded or within their contents, this data is considered as shared/overlap among all subscribers. Therefore, if one vehicle has managed to download this shared data from the RSU, it can later share it with other vehicles that belong to the same content provider. In this scenario, the RSU decides to either download all the shared data to the vehicle or allow the vehicle to get the shared data partially or fully from another vehicle. In case that neither the RSU nor other vehicles are able to transmit the shared data, the vehicle should request it from the cellular network.

2.4 Problem Definition and Mathematical Formulation

The main objective is to maximize the revenue of the RSU, therefore, we incrementally design and formulate this mathematically. First, we take into account the RSU resources then we extend our work by considering the V2V communication among vehicles. A further extension finalizes the system model by considering the overlap/similarities. Overlap and similarities represent a small potion of data that is shared among all the contents provided by one content provider.

In general, the problem can be defined as follows:

Definition 1. *Given a set of content providers, vehicles and requested contents, find the optimal resource allocation of RSU and V2V communication such that the RSU revenue is maximized. Assume a slotted time system with epochs of fixed number of time slots. Also, assume that RSU earns a certain profit if it downloads one data unit for each content provider and a percentage of the gain if the content is relayed over V2V. Also, assume there is overlap among the contents from the same provider.*

Table 2.1 lists the mathematical notations used in the problem formulation.

2.4.1 RSU Resource Scheduling

We divide the RSU coverage into several rate zones [S0 ... SN], each zone has a certain download rate r_s and it is calculated from:

$$r_s = \beta log_2(1 + SNR_s). \tag{2.1}$$

Now, to find if vehicle i resides in s at t, we define:

$$M_i^{s,t} = \begin{cases} 1 & \text{when } s_b \leq (t - \sigma_i) \bigtriangleup \vartheta_i \leq s_e, \forall i, t, \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where M is calculated offline and s_b and s_e represent the boundaries of the region s.

To find the size of content downloaded to vehicle i from the RSU during its residence time, we

Table 2.1: Table of notations for content delivery based on pricing model

Symbol	***Description***
$v(t)$	Set of Vehicles within the RSU's coverage area at t
ϑ_i	Velocity of vehicle i
S	Ranges of transmission rate in RSU's coverage area
d	Distance between two adjacent RSUs
L	Length of the coverage area of RSU
R	Transmission rates of ranges in RSU's coverage area
R_v	Transmission rate of V2V
Ti	Time slots when vehicle i is within RSU's coverage area
Γ_i	Playback rate of content requested by vehicle i
B	Size of RSU buffer
$V^{(n)}$	Set of vehicles belong to the current epoch
$V_n^{(n)}$	Set of vehicles belong to the current epoch only (new arrivals)
$V_o^{(n)}$	Set of vehicles belong to the current epoch and the previous epoch(s) as well
A_i	Size of content required for i
T_i	Last time slot for vehicle i within the range on RSU
σ_i	time slot when vehicle i enters epoch region
Ω_i	Downloaded content to vehicle i
n	present epoch
C_i	RSU benefit of one data unit from i
Q	Gain ratio of RSU from V2V
N_{ji}^t	1: if i and j within the communication range of V2V 0:otherwise
P_i^h	1: if i requests h 0:otherwise
E_h	Size of h
	Variables
$L_i^{(n)}$	Size of content downloaded to i via RSU in epoch n
$U_i^{(n)}$	Size of non-overlapped content transmitted from i via V2V in epoch n
$D_i^{(n)}$	Size of non-overlapped content received by i via V2V in epoch n
$\curlyvee_i^{(n)}$	Size of overlapped content transmitted from i via V2V in epoch n
$\curlywedge_i^{(n)}$	Size of overlapped content received by i via V2V in epoch n
x_i^t	1: if vehicle i is served at t 0: otherwise
y_{ij}^t	1: if vehicle i transmits content to j at t 0: otherwise
z_{ij}^t	1: if vehicle i transmits data belongs to h to j at t 0: otherwise
$w_i^{(n)}$	1: if i will download its h from the RSU 0:otherwise

define:

$$L_i^{(n)} = \sum_{t=1}^{T^{(n)}} x_i^t \bigtriangleup \{\sum_{s \in S} r_s M_i^{s,t}\}, \forall i \in V^{(n)}, \tag{2.3}$$

where $\bigtriangleup$ represents the length of one time slot.

Now, Eq(2.4) and Eq(2.5) guarantee that only one vehicle is served at each time slot.

$$\sum_{i \in V^{(n)}} x_i^t \leq 1, \forall t, \tag{2.4}$$

$$x_i^t \leq \sum_{s \in S} M_i^{s,t}, \forall i, t, \tag{2.5}$$

Also, we take into consideration the constraints on the RSU buffer size as in Eq(2.6).

$$\sum_{i \in V^{(n)}} (L_i^{(n)} + \Omega_i^{(n)}) \leq B. \tag{2.6}$$

Finally, if we consider only the RSU resources, the objective becomes:

$$\max \sum_{i \in V^{(n)}} L_i C_i. \tag{2.7}$$

Where the system maximizes the download size of vehicles L_i multiplied by its cost C_i.

2.4.2 V2V-assisted Content Delivery

Now, in order to address the impact of V2V, first we find the amount of data that each vehicle downloads via V2V communication; at each time slot, y_{ji}^t indicates whether vehicle i receives $R_v \bigtriangleup$ amount of content from vehicle j or not (Eq(2.8)). Eq(2.8) computes how much each vehicle downloads from other vehicle via V2V and outside RSU coverage where R_v is the data rate of V2V communication per second. This amount is summed and computed for each time slot until the vehicle enters the next RSU coverage.

$$D_i^{(n)} = \sum_{t=1}^{T^{(n)}} \sum_{j \in V_n^{(n)}: j \neq i} y_{ji}^t R_v \bigtriangleup, \forall i \in V_n^{(n)}. \tag{2.8}$$

Also, Eq(2.9) ensures both vehicle i and j must be within the V2V communication range at t before vehicle j can transmit to vehicle i.

$$y_{ji}^{t} \leq N_{ji}^{t}, \forall j, i \in V_n^{(n)}, \forall t, \tag{2.9}$$

where N is calculated offline based on the velocities of vehicles and their arrival times and it can be one only if the two vehicles left the RSU coverage and the distance between them is less than or equal a certain threshold (V2V communication range).

At time slot t, a vehicle can only serve one other vehicle and a vehicle can only receive from one other vehicle as in Eq(2.10) and Eq(2.11).

$$\sum_{j \in V_n^{(n)}: j \neq i} y_{ji}^{t} \leq 1, \forall i \in V_n^{(n)}, \forall t. \tag{2.10}$$

$$\sum_{i \in V_n^{(n)}: j \neq i} y_{ji}^{t} \leq 1, \forall j \in V_n^{(n)}, \forall t. \tag{2.11}$$

Also, a vehicle should not transmit and receive at the same time (Eq(2.12)).

$$\sum_{j \in V_n^{(n)}: j \neq i} y_{ji}^{t} + \sum_{j \in V_n^{(n)}: j \neq i} z_{ij}^{t} \leq 1, \forall i \in V_n^{(n)}, \forall t. \tag{2.12}$$

Next, for the total content size vehicle i transmits via V2V, we define:

$$U_i^{(n)} = \sum_{t=1}^{T^{(n)}} \sum_{j \in V_n^{(n)}: j \neq i} y_{ij}^{t} R_v \bigtriangleup, \forall i \in V_n^{(n)}. \tag{2.13}$$

Eq(2.14) and Eq(2.15) guarantee that previously scheduled V2V transmissions are not changed since that may lead to serious issues related to the chunk order.

$$U_i^{(n)} = U_i^{(n-1)}, \forall i \in V_o^{(n)}. \tag{2.14}$$

$$D_i^{(n)} = D_i^{(n-1)}, \forall i \in V_o^{(n)}. \tag{2.15}$$

The total amount to be sent via V2V should be less than or equal to the total download as in Eq(2.16).

$$L_i^{(n)} + \Omega_i^{(n)} \geq U_i^{(n)}, \forall i \in V^{(n)}. \tag{2.16}$$

Moreover, Eq(2.17) states that a vehicle should not receive more than what it needs in the first place.

$$L_i^{(n)} + \Omega_i^{(n)} - U_i^{(n)} + D_i^{(n)} \leq A_i. \tag{2.17}$$

After introducing V2V communication, the objective has changed to consider the new elements; D_i and U_i:

$$\max \sum_{i \in V_n^{(n)}} D_i C_i Q + \sum_{i \in V^{(n)}} (L_i - U_i) C_i, \tag{2.18}$$

where Q represents the percentage of profit that RSU earns if V2V used and $L_i - U_i$ deducts the amount of data that vehicle i transmit via V2V communication from the RSU profit.

2.4.3 Partial Content Overlap

For the overlaps, first, we have to calculate the overlap size will be received ($\curlywedge_i^{(n)}$) or sent ($\curlyvee_i^{(n)}$) for each vehicle as depicted in Eq(2.19) and Eq(2.20).

$$\curlywedge_i^{(n)} = \sum_{t=1}^{T^{(n)}} \sum_{j \in V_n^{(n)}: j \neq i} z_{ji}^t R_v \bigwedge, \forall i \in V_n^{(n)}. \tag{2.19}$$

$$\curlyvee_i^{(n)} = \sum_{t=1}^{T^{(n)}} \sum_{j \in V_n^{(n)}: j \neq i} z_{ij}^t R_v \bigwedge, \forall i \in V_n^{(n)}. \tag{2.20}$$

Eq(2.21) and Eq(2.22) guarantee that vehicles which transmit have all the overlap part while the receiver ones do not have it.

$$z_{ij}^t \leq 1 - w_j^{(n)}, \forall j, i \in V_n^{(n)}, \forall t. \tag{2.21}$$

$$z_{ij}^{t} \leq w_{i}^{(n)}, \forall j, i \in V_{n}^{(n)}, \forall t. \tag{2.22}$$

Also, both the transmitter and receiver of the overlap part must have requested the same overlapped content in the first place as in Eq(2.23).

$$z_{ij}^{t} \leq \sum_{h=1}^{H} P_{i}^{h} P_{j}^{h}, \forall j, i \in V_{n}^{(n)}, \forall t. \tag{2.23}$$

Since y and z deal with V2V communication, z should maintain similar constrains that ensure its work such as the proximity between vehicles (Eq(2.24)), unicast transmission (Eq(2.25), Eq(2.26) and Eq(2.27)).

$$z_{ji}^{t} \leq N_{ji}^{t}, \forall j, i \in V_{n}^{(n)}, \forall t. \tag{2.24}$$

$$\sum_{j \in V_{n}^{(n)}: j \neq i} z_{ji}^{t} \leq 1, \forall i \in V_{n}^{(n)}, \forall t. \tag{2.25}$$

$$\sum_{i \in V_{n}^{(n)}: j \neq i} z_{ji}^{t} \leq 1, \forall j \in V_{n}^{(n)}, \forall t. \tag{2.26}$$

$$\sum_{j \in V_{n}^{(n)}: j \neq i} z_{ji}^{t} + \sum_{j \in V_{n}^{(n)}: j \neq i} z_{ij}^{t} \leq 1, \forall i \in V_{n}^{(n)}, \forall t. \tag{2.27}$$

In addition, when a vehicle receives content related to the overlap part, this content should not be larger than the overlap part size as in Eq(2.28).

$$\lambda_{i}^{(n)} \leq \sum_{h=1}^{H} P_{i}^{h} E_{h}, \forall i \in V_{n}^{(n)}. \tag{2.28}$$

Eq(2.29), Eq(2.30) and Eq(2.31) make sure that no reschedule is allowed for V2V communication.

$$\Upsilon_{i}^{(n)} = \Upsilon_{i}^{(n-1)}, \forall i \in V_{o}^{(n)}. \tag{2.29}$$

$$\lambda_i^{(n)} = \lambda_i^{(n-1)}, \forall i \in V_o^{(n)}. \tag{2.30}$$

$$w_i^{(n)} = w_i^{(n-1)}, \forall i \in V_o^{(n)}. \tag{2.31}$$

Also, since we assume that V2V can establish only one connection at time, therefore, y and z should not equal 1 at the same time slot (Eq(2.32) and Eq(2.33)).

$$\sum_{j \in V_n^{(n)}} z_{ij}^t \leq 1 - y_{ij}^t, \forall i \in V_n^{(n)}, \forall t. \tag{2.32}$$

$$\sum_{j \in V_n^{(n)}} z_{ji}^t \leq 1 - y_{ji}^t, \forall i \in V_n^{(n)}, \forall t. \tag{2.33}$$

The total amount to be sent via V2V plus the overlap part, if downloaded through the RSU, should be less than or equal to the total download that a vehicle receive from the RSU as in Eq(2.34).

$$L_i^{(n)} + \Omega_i^{(n)} \geq U_i^{(n)} + w_i^{(n)} \sum_{h=1}^{H} P_i^h E_h, \forall i \in V^{(n)}. \tag{2.34}$$

Moreover, a vehicle should not receive more than what it needs in the first place and that includes RSU and V2V communication (Eq(2.35)).

$$L_i^{(n)} + \Omega_i^{(n)} - U_i^{(n)} + D_i^{(n)} \leq A_i - (1 - w_i) \sum_{h=1}^{H} P_i^h E_h, \forall i \in V^{(n)}, \tag{2.35}$$

where Ω_i is the content that has be already downloaded to vehicle i in the previous epoch(s).

Now, with the overlap parts, we can rewrite our objective as follows:

$$\max \sum_{i \in V_n^{(n)}} (D_i^{(n)} + \lambda_i^{(n)}) C_i Q + \sum_{i \in V^{(n)}} (L_i^{(n)} - U_i^{(n)}) C_i, \tag{2.36}$$

where Q is the gain ratio of the RSU from V2V (e.g 90%). $\gamma_i^{(n)}$ is not deducted from the gain since both the transmitter and receiver need it.

Algorithm 1: Heuristic - RSU Resource Scheduling

1 **Inputs:** $C_i, M_i^{s,t}, R, A_i$.
2 **for** t *in* T **do**
3 **for** i *in* $V_n^{(n)}$ **do**
4 $x_i^t = 1$ where $\{C_i * \sum_{s \in S} r_s M_i^{s,t}\}$ gives the largest profit among $V^{(n)}$ and $\bigtriangleup * \sum_{s \in S} r_s * M_i^{s,t} \leq A_i - \Omega_i^{(n)}$.

2.5 Alternative Solution

Our system model is difficult to solve as it contains binary variables and resource scheduling [26]. Therefore, we propose alternative heuristic solutions which can be solved in polynomial time. The solution is based on a greedy approach where the objective is to maximize the profit at each time slot individually taking into account other constraints. Meanwhile, for further analysis, we develop three heuristic algorithms; first one is for scheduling the resources of the RSU only, second one considers V2V communication as well, and third one takes into account the overlaps.

Algorithm 1 maximizes the profit at each time slot by selecting the vehicle which gives the highest profit among a set of vehicles within the range of the RSU. The profit is calculated by multiplying the rate of the zone the vehicle resides in by the price of the requested content. Algorithm 2 has two stages, first one maximizes the profit from RSU as in Algorithm 1, however, if a vehicle already was scheduled for V2V and not yet satisfied, then it should keep the previous schedule as shown in line 3 and 4. Next, through the V2V, the algorithm checks which two vehicles are within the communication range of V2V and the difference between their prices is the largest. Also, at the same time, the high value content vehicle still needs more content while the other vehicle downloads enough data to transmit. If the algorithm finds two vehicles that satisfy the conditions above, then set $y_{i,j}^t = 1$ to enable the transmission through V2V. Finally, Algorithm 3 combines the other two algorithms, it consists of two stages as in Algorithm 2, however, the only difference is that in the second stages it checks which one gives the highest profit, the overlap or regular V2V transmission, since two V2V communications at the same time is not allowed.

Algorithm 2: Heuristic - V2V-assisted Content Delivery

Inputs: $C_i, M_i^{s,t}, R, A_i, M_{i,j}^t, U_i, Rv$
for t *in* T **do**
  for i *in* $V^{(n)}$ **do**
    if $i \in V_n^{(n)}$ *and* $x_i^{t(n-1)} = 1$ *and* $L_i < U_i$ **then**
      $x_i^t = 1$
    else
      $x_i^t = 1$ where $\{C_i * \sum_{s \in S} r_s M_i^{s,t}\}$ gives the largest profit among $V_n^{(n)}$ and $\bigtriangleup * \sum_{s \in S} r_s * M_i^{s,t} \leq A_i - \Omega_i^{(n)}$.
for t *in* T **do**
  for i *in* $V_n^{(n)}$ **do**
    for j *in* $V_n^{(n)}$ **do**
      if $N_{i,j}^t = 1$ **then**
        set $y_{i,j}^t = 1$ if the difference between their profit $(C_j - C_i)$ is the biggest among all other vehicles and $L_i \geq U_i$ and $L_j + D_j \leq A_j$

2.6 Numerical Results and Analysis

A four-lane one way highway segment is used in the simulation and the setup parameters are shown in table 2.2. The system model was solved by IBM ILOG CPLEX, while the heuristic algorithms were coded using Python. In addition, the free-flow traffic model is implemented by Python and it is adopted from [27,28] based on field observations. The relationship between vehicle speed and traffic density can be expressed as $v = v_f(1 - \rho/\rho_{jam})$ whereas v is the mean speed, ρ is the traffic density, i.e., vehicle(s) per kilo meters, ρ_{jam} is the jam density (occurs when the speed of vehicle is 0 m/s), and v_f denotes the free flow speed (occurs when the density of the road is 0 veh/m). Moreover, the arrival rate of vehicles is calculated by $\lambda = \rho v$. Also, the vehicle subscribers randomly request contents and we consider the total revenue and download generated per one RSU.

To perform a comprehensive analysis and highlight the gains achieved with enabling V2V data exchange and allowing overlaps among delivered content, we analyze our proposed optimal solution of the following three methods:-

- RSU-only: consider video delivery through RSUs and evaluate the resulting revenue.
- RSU-V2V: consider video delivery through RSUs and V2V data exchange and evaluate the

Algorithm 3: Heuristic - Partial Content Overlap

Inputs: $C_i, M_i^{s,t}, R, A_i, M_{i,j}^t, U_i, Rv$
for t *in* T **do**
  for i *in* $V^{(n)}$ **do**
    if $i \in V_n^{(n)}$ *and* $x_i^{t(n-1)} = 1$ *and* $L_i < U_i + \Upsilon_i^{(n)}$ **then**
      $x_i^t = 1$
    else
      $x_i^t = 1$ where $\{C_i * \sum_{s \in S} r_s M_i^{s,t}\}$ gives the largest profit among $V_n^{(n)}$ and $\bigtriangleup * \sum_{s \in S} r_s * M_i^{s,t} \leq A_i - \Omega_i^{(n)}$.
for t *in* T **do**
  for i *in* $V_n^{(n)}$ **do**
    for j *in* $V_n^{(n)}$ **do**
      if $N_{i,j}^t = 1$ **then**
        set $y_{i,j}^t = 1$ if the difference between their profit $(C_j - C_i)$ is the biggest and $(C_j - C_i < C_j(1+Q))$
        Or set $z_{i,j}^t = 1$ if $w_i = 1$ and $w_j = 0$ and $\lambda_j^{(n)} + D_j + L_j \leq A_i$

resulting revenue of RSUs and relay vehicles respectively.

- RSU-V2V-overlap: consider video delivery through RSUs and V2V data exchange in the existence of overlaps among delivered content. Evaluate the resulting revenue of RSUs and relay vehicles respectively.

In addition to the three previously listed methods, we consider a method that maximizes the total amount of video delivery by the RSUs regardless of their prices. Max Download can be of great help to understand the amount of traffic that our proposed system model offloads from the cellular. The results generated below are the average of 5 iterations.

2.6.1 Model Evaluation

We start our evaluation by determining the total amount of delivered content and the resulting RSU revenue generated using RSU-only method as compared to Max Download method. As shown in Figure 2.2 (a) and (b), although RSU-only downloads less content than Max Download, the difference does not exceed 8% in the worst case and it decreases when the traffic density of the road

Table 2.2: Simulation parameters in prefetching improving content delivery

Parameter	***Value***
d	1500 meter
Number of content providers	4
Required content size	[60-300] Mb
R_v	15 Mbps
V2V communication range	60 meters
Number of rate zones	5
RSU coverage	250 meter
RSU rates	11 to 27 Mbps
Q	0.9
Δ	1 second
v_i	[10-50] meter/second
E_h	[15-30] Mb
C_i	[0.1-8] per Mb

increases. In contrast, the difference between the two methods in terms of revenue is significant increases further to reach around 30% for a density equals 25 veh/km. The proposed approach can then generate notable revenues through selective offloading from the cellular network.

Next, we highlight the gains achieved through leveraging V2V data exchange and the existence of overlap content. Figure 2.3 demonstrates the gain/revenue of both RSU-V2V and RSU-V2V-overlap with respect to RSU-only. In this figure, the y-axis represents how much revenue/gain RSU-V2V and RSU-V2V-overlap are able to achieve over RSU-only in percentage where, as previously mentioned, RSU-only does not consider V2V communications. The figure also shows the comparison using two distances. 1Km and 1.5Km here indicate the distance between two adjacent RSUs coverage area. Obviously, RSU-V2V-overlap outperforms the other two methods and the relay vehicles can almost double the gain through transmitting the overlapped data via V2V communication. Moreover, the revenue produced by RSU-V2V-overlap, as compared to the other two methods, increases dramatically while the density of the road is increasing because there are more chances that more vehicles requesting the same shared data will meet after leaving the RSU coverage. In addition, from the same figure, one can see that the distances between the adjacent RSUs can

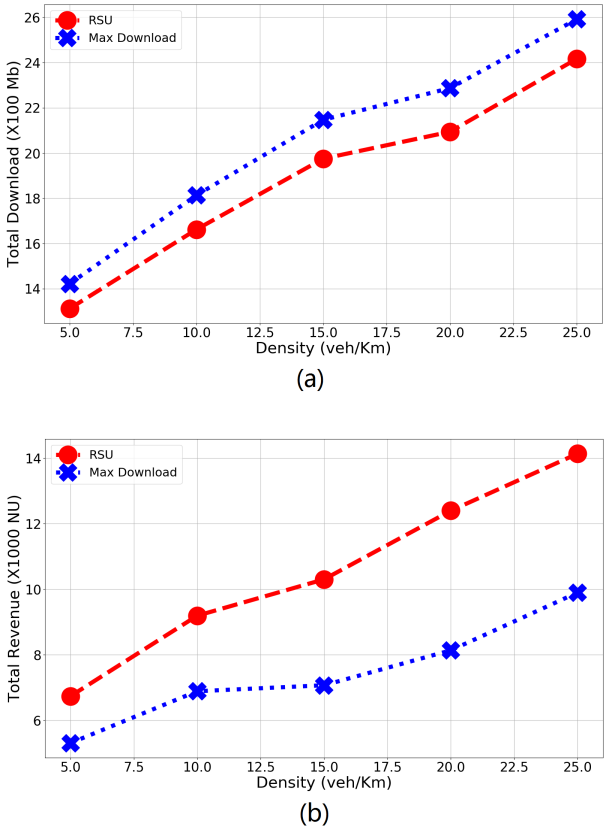


Figure 2.2: Comparison between Maximizing download and Maximizing profit in terms of (a) the total amount of data downloaded (b) the total revenue earned.

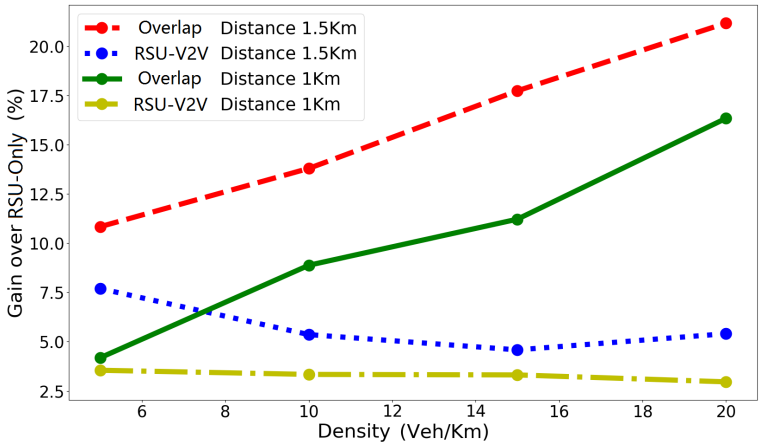


Figure 2.3: Comparison between Overlap-V2V-RSU and V2V-RSU versus RSU only

improve the performance of RSU-V2V-overlap and RSU-V2V over RSU since there will be more time slots for V2V communication as this distance extends which means that vehicles interested in same overlaps have higher chances to become connected with each other after leaving the RSU coverage.

2.6.2 Effect of Buffer Capacity

In this work, since the connection time between the vehicle and the RSU is short and WAVE protocol bandwidth is limited, then the wireless resource is the bottleneck in the network. However, as the V2X technologies advance and, also, other communication protocols can be used in the near future such as mmWAVE, C-V2X which provide much higher data rate, therefore, the buffer/cache capacity may face congestion events due to the high traffic volume needs to prefetech. The same scenario is also applicable to the backhaul, however, a dedicated fiber optic connection can always be faster than the wireless resources.

To study the impact of having low and high buffer capacity, we carried out an experiment showing this effects. As it is demonstrated in Fig. 2.4, we present two sets of results; one, Fig. 2.4 (a), for the total revenue achieved with different buffer sizes. Fig. 2.4 (b) illustrates how much can be downloaded using various buffer sizes for the three methods we propose in this contribution.

First of all, it is obvious that as the buffer size increases, the total revenue and download amount increase as well. The is due the limitation enforced by the buffer capacity which limits the ability

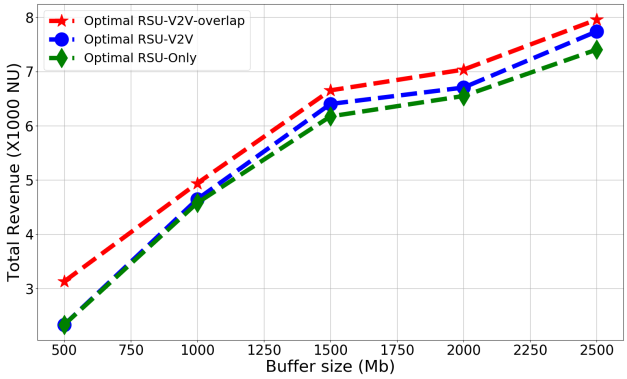
to serve contents to vehicles.

In 2.4 (a), it can be observed that optimal RSU and optimal V2V have almost similar revenue. However, as the buffer capacity increases, the gap between the two methods becomes clearer. When the buffer size increases, the system will have more choices to manage the resources for vehicles which means there are more chances to leverage V2V communication.

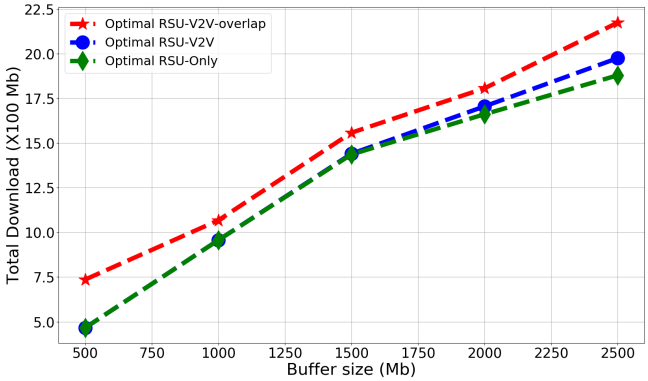
The other insight we can see in figures 2.4 (a) and (b) is that optimal overlap always outperforms the other two methods. That is because the system model can always exploit the existence of overlaps/similarities to download more contents through V2V communication. In average optimal overlap downloads 12% more data over the other two methods. Also, optimal overlap achieves around 4% revenue more than than optimal V2V where the latter does not consider overlaps.

2.6.3 Pricing Model Evaluation

In this section, we evaluate our pricing model by assigning different prices for a specific content provider and observe the impact on the QoS his customers have. Figure 2.5 depicts the results of four experiments where we fix the prices of three content providers while vary the prices of one content provider. The assigned prices for content providers 2, 3 and 4 are 0.5, 1 and 3 units respectively. For the first content provider we give different prices, 0.1, 0.7, 2 and 5. The satisfaction level of one content provider indicates the percentage of served content from that provider with respect to the total amount of requested content and its value is constrained between 0 and 1, where 0 means non of its customers received any service while 1 indicates that all its customers were fully served. In other words, it is the ratio of the amount served to the amount requested. It can be observed that the quality of service is impacted by two things, first is the price value and second is the values offered by other content providers. The price effectively represents a weight the prioritizes the content of a specific content provider among others. When the price of CP1 is the lowest, its customers get the worst satisfaction rate. However, when a higher price is paid to serve its content, the satisfaction level of its customers reached higher than any of the other three content providers. A similar behavior applies when the content provider charges different rates for its content causing customers that request the more expensive content to be better served than other customers.



(a) Total revenue of RSU



(b) Total download for vehicles

Figure 2.4: Comparison between the three methods, RSU-Only, V2V, and Overlap versus different buffer sizes

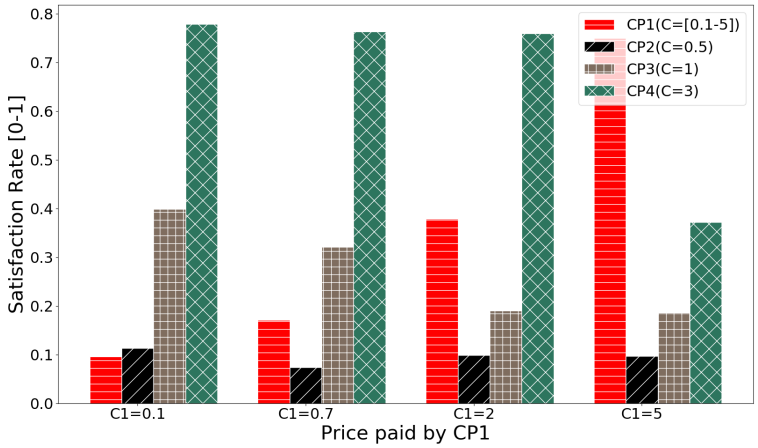


Figure 2.5: Impact of price on the QoS

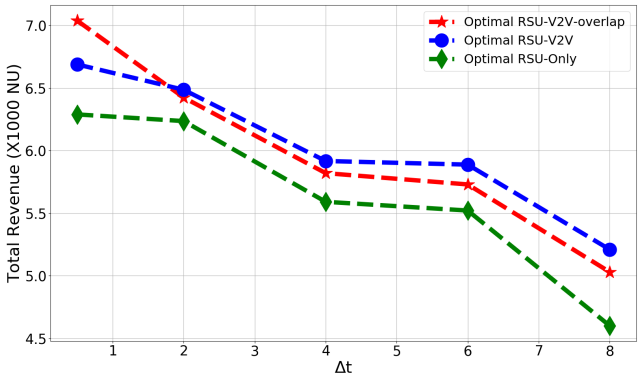


Figure 2.6: Impact of having different Δ values

2.6.4 Effect of Time Slot Size

The size of time slot is of high importance as it determines the complexity of the solution. As Δ decreases, there are more time slots that need to schedule. Thereby, a small Δ gives better control over the resource since we can better allocate the resources among the vehicles.

As we increase the time slot size, we degrade the efficiency. Indeed, as the system allocates resources for bigger time slots, it will have less freedom to manage the resources for different vehicle requirements. For example, if a vehicle needs only 0.1 sec to download the rest of content while the RSU schedules for 1 sec time slots, then the system will not be able to serve this vehicle, otherwise, it will waste 0.9 sec. Imagine now if the time slot size is 0.1 sec, then the same vehicle can receive service for 0.1 sec and the rest goes to other vehicles.

On the other size, a small time slot size has higher complexity than large one. When Δ is small, then more time slots are available which means than the RSU needs to schedule for more time slots. Therefore, since lowering the size of time slots gives better solution and increase the complexity at the same time, Δ becomes an essential parameter that highly impacts the performance of the system.

Fig. 2.6 illustrates the aforementioned effect of time slot size. It is clear in the figure that as we enlarge Δ value, the solutions of the three methods deteriorate dramatically. A time slot of size 0.5 sec can produce 25% gain over time slot of size 8 sec. In contrast, when Δ = 8 sec, the complexity of the system is 16 times less than the one of 0.5 sec.

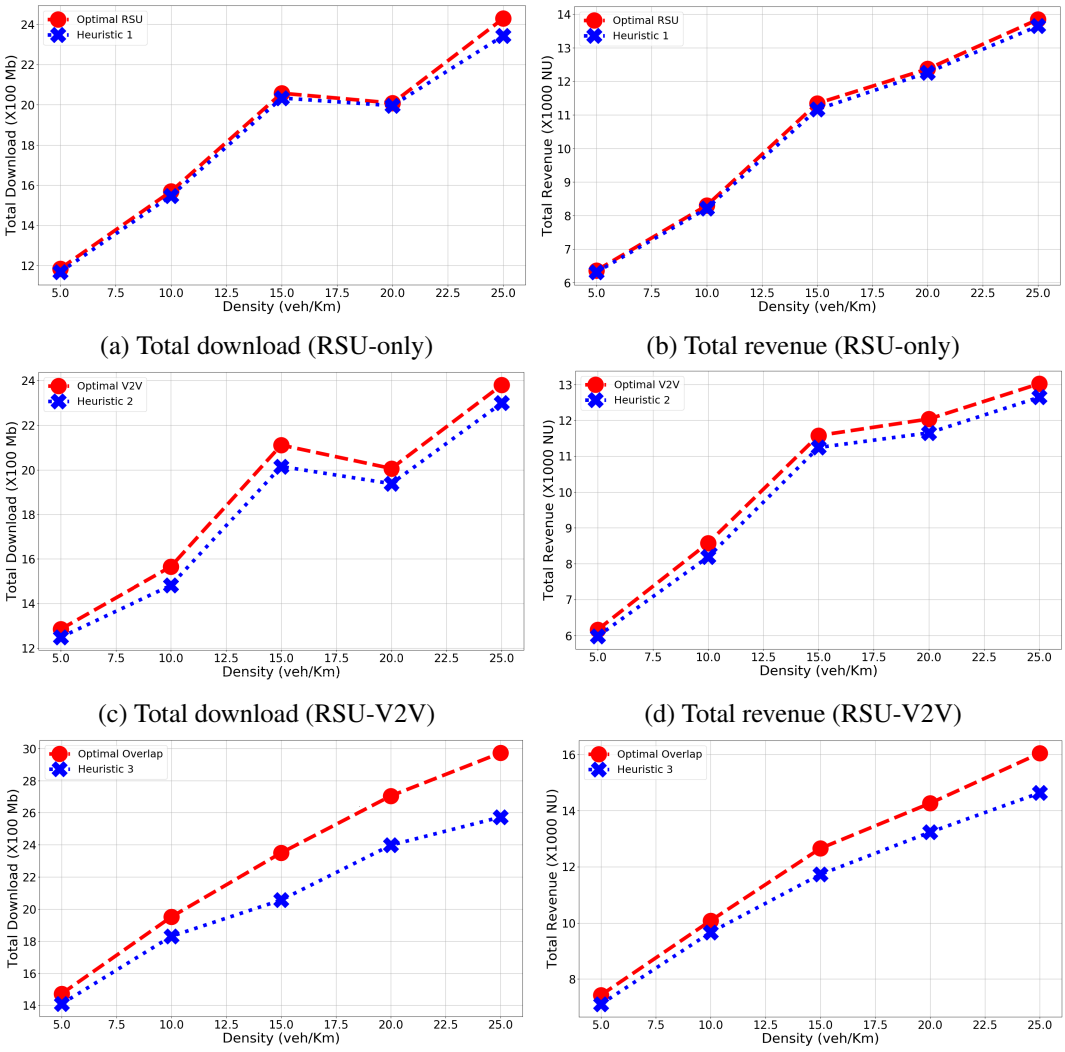
Another interesting insight in the same figure that showing the significance of using small time slot size is the difference between the revenue of RSU-V2V-overlap and RSU-V2V. As it can be seen, when Δ is small as 0.5 sec, RSU-V2V-overlap outperforms the other methods. However, as Δ becomes bigger and bigger, RSU-V2V starts performing better than RSU-V2V-overlap. RSU-V2V-overlap deals with overlaps and according to the mathematical formulation Eq(2.21) and Eq(2.22), it can either download the entire overlap part or does not download any. Therefore, as the time slot size increases, the two constraints of Eq(2.21) and Eq(2.22) will limit the solution space for RSU-V2V-overlap, while RSU-V2V does not have such constraints. Therefore, RSU-V2V can achieve better solutions.

2.6.5 Algorithms Evaluation

Now, in order to evaluate the performance of the algorithms we develop (1,2, and 3), Figure 2.7 demonstrates the gap between the optimal solutions and our algorithms in terms of download and revenue. Figure 2.7 (a) and (b) compares Algorithm 1 with the optimal solution considering RSU only. The results show that Algorithm 1 is able to provide solutions that are almost the same as that produced form the optimal.

Similarly, Figure 2.7 (c) and (d) compare Algorithm 2 with the optimal model taking into account V2V. Although the difference two solutions is now more obvious, however, Algorithm 2 is still able to produce solid solutions with gap less than 5%.

Finally, Figure 2.7 (e) and (f) compare Algorithm 3 with the optimal model that considers all the features including the overlaps. Again, Algorithm 3's performance proves that our algorithm is capable of producing solutions very close to the optimal. Even though the gap between the two solution becomes bigger as long the density of the road increases, the difference is still acceptable (within realistic densities) and it does not go larger than 10% in terms of revenue which is our main objective. The reason behind the gap between the optimal solution and the heuristic solution is as more vehicles come, there are more chances that vehicles having overlaps will meet. While the optimal solution is able to exploit every possible chance to make revenue, our heuristic method cannot. However, the results proves that with our method, it is possible to come with a very close revenue at much lower complexity.



(a) Total download (RSU-only)

(b) Total revenue (RSU-only)

(c) Total download (RSU-V2V)

(d) Total revenue (RSU-V2V)

(e) Total download (RSU-V2V-overlap)

(f) Total revenue (RSU-V2V-overlap)

Figure 2.7: Comparison between the three methods, RSU-only (Optimal RSU), RSU-V2V (Optimal V2V), and RSU-V2V-overlap (Optimal Overlap) versus different buffer sizes

From Figure 2.7, we can conclude that within polynomial time we were able to achieve enough proximity between the algorithm and the optimal values and with much less complexity.

2.7 Summary

In this chapter, we present a pricing and resource allocation model for RSU and V2V communication to encourage cooperation with the cellular operators in order to provide better video services. We formulated the problem by dividing it into several phases to analyze the impact of various factors. Furthermore, we provide alternative solutions based on heuristic algorithms and we proved that they are capable of producing impressive results with difference less than 5% in normal road circumstances.

Chapter 3

Enabling Content Delivery on Highway Segments with Disconnected RSUs

3.1 Introduction

3.1.1 Motivation

In Chapter 2, we discussed the benefit of applying pre-fetching principles in order to maximize content delivery quality. In this chapter, we further improve RSU content delivery services by considering caching techniques. As known, communication devices usually come with caching capability that is proven to have a great positive impact on reducing the costs and improving the quality of service (QoS) [29–32]. The content cached in smart vehicles for running in-vehicle infotainment services can be leveraged by RSUs to serve other passing by vehicles. This is done by having the RSU populate its own cache through fetching contents stored in vehicles crossing its coverage range. Later, these contents are delivered to other interested vehicles. Such a technique, however, comes with several limitations that challenge its foreseen benefits.

Existing literature that addresses caching in general typically focuses on predicting the content popularity and then caching the most popular content. Some works such as [2, 3, 33] address the issue of time-variant popularity by resorting to intelligent techniques such as reinforcement learning. Despite the efficiency of their methods, profiling content popularity has several shortcomings

when it comes to caching systems. Indeed, the stream of released content never stops and many contents may lose their popularity within a very short period of time. Moreover, by the time the content popularity profile is determined, the content may have already lost its popularity and become inactive. Therefore, in this work, we tend to avoid computing the popularity of contents at the RSU and, instead, rely on other features (i.e. size) that facilitate finding suitable caching policies that maximize the serving rate of incoming vehicles.

3.1.2 Challenges

To improve the QoE perceived by in-vehicle passengers, traffic congestion on backhaul links has to be alleviated through several means. One of which constitutes storing contents closer to the edge [11]. This is done by utilizing caching techniques that can selectively identify and cache desired contents at the edge to serve incoming vehicle requests. However, due to the limited resources in terms of cache size, only a small portion of the whole content library can be cached at one time. Additionally, the stream of contents generation is continues; that is, new contents are released each time and from different sources. Such contents may have different sizes and features. Meanwhile, vehicles may visit several RSUs along their route and download contents. Some of the visited RSUs may have Internet connectivity while others are disconnected or suffer from limited connectivity. In this work, we aim at actively utilizing the latter RSUs in improving the QoS by intelligently fetching proper contents from existing vehicles within their range. Eventually, these RSUs maximize the service rate for other arriving vehicles interested in similar contents. It is worth mentioning that vehicles that cannot be served by a given RSU may download their requested content from the cellular network. This downloaded content can then be cached by other RSUs along their routes.

Additional challenges are related to inciting vehicles to contribute their content as well as their wireless resources to populate RSU caches. Finally, in practice, contents come in large sizes, while the wireless radio resources are limited and vehicles' residence vary within the range of the RSU. Hence, in order to fetch a content or serve a content to a vehicle, one or a few time slots may not be sufficient to complete the process.

3.1.3 Contribution

In this chapter, we present a system that mitigates the problem of either a lack of or congested backhaul in a vehicular environment. To the best of our knowledge, it is the first work that addresses this issue by actively engaging disconnected RSUs in delivering content of dynamic libraries to upcoming vehicles. The following summarizes our contributions:

- We propose a system model that makes use of the cached contents on passing vehicles in order to fill up the RSU cache such that the RSU can later serve the requests arising form vehicles without need to access the backhaul link. The system also takes into account some kind of incentive that must be paid for the cooperating vehicles in order to motivate them to continue the collaboration with the RSU operators.

- Then, we model the proposed system taking into account the limited wireless resources, formulate the problem mathematically, and show why it is challenging to solve using traditional optimization techniques.

- Next, to solve the aforementioned problem, we resort to intelligent techniques, namely Deep Reinforcement learning besides a crafted algorithm for the downlink resource scheduling. We define our system as Markov Decision Process (MDP) and specify its items (state, action, reward). Then, we use a DQN-learning and design a heuristic algorithm to solve the caching problem. Most importantly, our DQN-based solution does not stick with certain contents, but rather, it learns the best policies which optimize the performance based on the utility of the cached contents and their size.

3.1.4 Related Work

Data dissemination and content delivery in VANETs are gaining substantial momentum in the literature owing to the great benefit they can supply to the vehicle passengers and cellular network operators too. [34] suggested cooperative caching among a set of RSUs by replicating contents. The incentive behind this work is that RSUs located at the same roads are highly correlated during the service of the traffic flow. [35] studied the stochastic delay of wireless channels alongside cache

based system in vehicular networks. Therefore, a stochastic network calculus is used to evaluate the stochastic playback delay performance of vehicular video content delivery with cache-enabled RSUs. In [36], authors proposed a caching system to deliver large contents such as videos to fast moving vehicles. They addressed the efficient content delivery problems in VANET by caching popular files in the RSUs with large storage capacity. [37] solves the problem where multiple content providers (CPs) aim to improve the data dissemination of their own contents by utilizing the storage of RSUs. They used multi-object auction-based solution to get a sub-optimal solution for the competition among the CPs. [33] proposed a Q-learning based caching strategy where the system predicts the the path trajectory of vehicles and based on that it decides which RSUs should cache the contents in order to reduce the latency. [38] suggested cooperation between RUSs and vehicles and a model determines where to obtain the content. A cross entroy based dynamic caching strategy is used to facilitate caching at the edge. [39] tried to predict the mobility of vehicles so they can prefetch the content at the edge to reduce latency. Their work is based on multi-hierarchical cache nodes where content are stored both on the RSUs and in the regional servers. Next, [40] worked on vehicular content centric networks (VCCNs) where vechicles broadcast their requests. Their work focused on caching popular content at a set of vehicle which are going to visit "hot region" and stay for a long time. Hot regions are the spots on the map which their traffic densities are higher than other regions. [3] proposed three levels of caching, namely; BS, TV White Space (TVWS), and RSU. Now, since these three levels of communication technologies provide different communication range, they suggested that big files should be cached at the BS and small ones might be stored at the RSU. Their objective is to maximize the average download rates for each transmission unit. Furthermore, the authors of [41] also worked on caching in vehicular networks. However, they considered fixed library of contents where the popularity can be easily figured out, while this is not realistic. Thus, in our work we consider different set of contents at each service time. Second, their work is based on different types of contents which may demand stringent latency such as navigation maps, while our work is for infotainment services that needs to offload large traffic volume.

Reinforcement learning has been used widely for solving caching problems. In [42], deep Q network has been employed to solve for interference alignment, an emerging interference management technique, in cache enabled system. The authors of [43] used Q-learning to solve for cache

replacement policies in order to optimize the costs of BS transmission costs. [44] proposed deep reinforcement learning as a framework to enhance and improve cache hit rate performance while it requires no information regarding the popularity of contents. In [45], federated learning is combined with reinforcement learning to optimize caching strategies for a collection of edge nodes that can exchange their learning parameters. The authors of [46] proposed liquid state machine (LSM) to solve for the joint problem of caching and resource allocation for unmanned aerial vehicle networks. The users' content request distribution is predicted by LSM beside enabling UAVs to better manage their wireless resources to serve users with stable queues. In addition, this work took into account two types of transmission; licensed and unlicensed bands.

To the best of our knowledge, our work in this chapter differs from the literature in that it is the first work which dynamically exploits common interest among vehicles to store popular contents on RSU cache. Moreover, it is the first work that solves for caching using deep reinforcement learning without assuming fixed library of contents. Instead, it builds a table to compute the utility of the cached and available contents in order to decide for better caching strategies.

3.2 System Model

We consider a road segment as shown in Fig. 3.1 where an RSU is deployed. The RSU has a cache capacity of size K and coverage area of size G. We also assume that vehicles arrive at a certain rate leading to a given traffic density ρ. In addition, without loss of generality, we assume each vehicle has one content cached and requests another content once it enters the coverage area of the RSU. All the contents are stored on a server that is not accessible by the RSU.

3.2.1 Content and Popularity Model

We consider a library of contents containing M items/contents located somewhere in the cloud with some of its contents cached at vehicles. To bring our model to reality, we assume contents have different sizes. Let S_m denote the size of m'th content. Also, let X_i and Y_i denote the content cached and content requested by the i'th vehicle respectively.

Vehicles request contents with different frequencies, e.g., content A might receive 10 requests

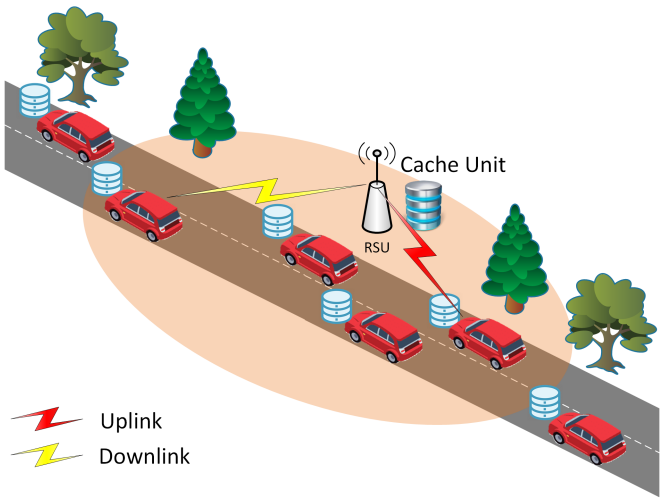


Figure 3.1: System Model

while content B receives only 1 request during the same period. The popularity of contents are modeled using Zipf distribution where the skewness parameter characterizing the distribution.

On the other hand, the lists of available contents inside the vehicles' caches also follow Zipf distribution. This is justified by the following; during their sojourns, vehicles pass through several regions at different times and days. Each region and time has its own characteristics [47, 48], thus, have various content popularities. For example, early morning, people tend to watch news while social media contents are preferred during evening times [49]. The content popularity and caching content inside the transport networks in [49] is modeled as Zipf-like distribution. Additionally, some on-vehicle caching mechanism might leverage the fact that using similar distribution can improve the performance of the network [50].

The library of contents changes from time to time. In reality, each day there are new contents generated while the existing contents start to experience much lower hit rates after some time. Therefore, in this work, we assume each service time has different library of contents. Here, we assume that the RSU operator does not know Zipf skewness parameter beforehand and it experiences new library of contents at each service time.

3.2.2 Traffic Model

In this work, we assume a free-flow model similar to [51] where the relationship between velocity and traffic can be expressed as $v = v_f(1 - \rho/\rho_{jam})$ where v_f is the expected velocity, ρ is the traffic density (i.e., vehicle(s) per kilo meters) and ρ_{jam} is the jam density (jam density is extreme traffic density when traffic flow stops completely). Moreover, the arrival rate of vehicles is calculated by $\lambda = \rho v$. The vehicle arrivals follow a Poisson distribution and velocities are generated using truncated normal distribution.

3.2.3 Caching Strategy

As mentioned above, the popularity of requested and cached contents are varying with time. Where each one is modeled using different transition and Zipf values. Therefore, the RSU should decide for each vehicle whether it takes the content cached on the vehicle or not. If the RSU decides to take a content from a vehicle, then it has to pay a certain cost in return depending on the size of content. Practically, the RSU may need to compensate the sender vehicle for leveraging its resources such as the cache and transmission equipment. Moreover, the RSU needs to offer some incentive for the vehicles to motivate them to cooperate with the RSU in order to facilitate content delivery [52]. Therefore, let $w \in (0 - \infty]$ represent the ratio of uploading one data unit of content to the RSU (uplink) with respect to serving one data unit of content to vehicles (downlink).

The size of RSU cache is limited and relatively small in comparison with the library size, hence, it can be filled up quickly. We should note that the size of the library of contents is much bigger than the cache size (i.e., $K \ll \sum_m^M (Z_m)$). Thereby, when the RSU cache becomes full and new content cannot fit in the cache, a replacement technique must be used.

3.2.4 Communication Model

We assume a time division multiplexing access as the transmission technology. We further assume separate spectrum for up and downlink. The RSU uses downlink to serve the vehicles and uplink to fetch the contents from the vehicles. In order to avoid collisions, downlink resources are allocated for vehicles and since our goal is to maximize served data amounts then allocating the

resources can be done in any order. For the uplink, however, we prevent simultaneous transmissions to avoid any possible collisions to occur. In other words, the RSU cannot decide to fetch a content from a vehicle if it is presently fetching another contents since there might not be sufficient resources to fetch both contents completely. We further assume constant data rate for up and downlink similar to [53].

3.2.5 Operation Phase

We consider a time slotted system where at each time slot there might be a vehicle arriving or already in the RSU coverage area. The RSU should decide once the vehicle arrives if RSU wants to fetch the cached content on the vehicle. If it decides to fetch the content, then the RSU receives it when enough time slots are allocated. Without loss of generality, we assume the RSU is capable to download the vehicle cached content during its sojourn time. Worth noting, this system is based on the arrival events of vehicles rather than time slots because actions are only required when a vehicle enters the RSU coverage area.

At the beginning, we assume the RSU cache is empty, therefore, the RSU cannot serve any requests. However, as the time advances, it starts to fill up its cache. It is worth-mentioning that several vehicles might arrive at the same time. In that case, the RSU decides for each vehicle individually. Moreover, the RSU cache capacity is limited, so when it is filled up, the RSU should start to replace contents that have reduced popularity with more demanded content at that time. Since the popularity changes, then the RSU should update its cache regularly.

3.3 Mathematical Formulation

To mathematically formulate the joint problem of content caching and resource allocation: For I vehicles that will pass through the coverage area of the RSU, let Y denote the request for contents and X denote the available contents. Table 3.1 lists all the mathematical notations used in our formulation.

Before starting with the mathematical formulation, we need to emphasize that this system is workable as long as the content sizes do not exceed a certain value. The largest size of content that

Table 3.1: Mathematical Notations in caching enabled content delivery

Parameters	
T	time horizon composed of several time slots $\{0,1,..,t,..,T\}$
I	set of vehicles passing during time T $\{0,1,..,i,..,I\}$
U_{rate}	Uplink data rate
D_{rate}	Downlink data rate
X_m^i	Vehicle i has content m
Y_m^i	Vehicle i requests content m
S_m	Size of content m
M	Set of contents $m = \{0,1,...,M-1\}$
V_{start}	beginning of RSU coverage
V_{end}	end of RSU coverage
P_i^t	Position of vehicle i at time t
C	Capacity of RSU cache
Variables	
k_i	1: Vehicle i is served, 0: otherwise
$u_{i,m}^t$	1: if time slot t is allocated for vehicle i to fetch content m
h_m^t	1: remove content m at time t
e_m^t	1: content m is cached at time t

is applicable in this system is equal to the maximum amount that can be fetched from one vehicle during its contact time with the RSU, Eq (3.1).

$$S_m \leq U_{rate}\Pi, \forall m \in M, \tag{3.1}$$

where Π is the least sojourn time.

In this system, a vehicle may hold several contents cached while it is in need for only one content during its contact with the RSU. Therefore, $\sum_{m=0}^{M} X_m^i > 0, \forall i \in I$ and $\sum_{m=0}^{M} Y_m^i = 1, \forall i \in I$. The RSU can decide to take one or several contents from the same vehicle as long as enough resources are allocated. In addition, the RSU can also decide not to fetch any contents.

In this work, as our interest lies mainly in finding caching decisions rather than wireless resource allocation, therefore, for simplicity, we consider fixed data rates where U_{rate} and D_{rate} represent the up and downlink data rates, We further assume a particular range for the RSU wireless coverage where vehicles can only communicate with the RSU within that range [51].

Now, let us define Γ as a parameter which indicates whether a vehicle is within the range of the RSU or not.

$$\Gamma_i^t = \begin{cases} 1 & V_{start} \leq P_i^t \leq V_{end}, \forall i \in I, t \in T, \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

This parameter is of special importance to ensure no transmission is allowed beyond the coverage range of the RSU as we will show later in this section.

Next, we present the set of constraints for our problem.

3.3.1 Cache Management

The RSU cannot cache a content unless it is fully fetched from a vehicle. That is, enough time slots have been allocated to the vehicle and the content to be uploaded. The number of time slots depends on the size of the content and uplink rate. Indeed, it is the resultant of dividing the size of content by the uplink rate.

Then, let us define e to indicate whether a content is cached or not at each time slot, where:

$$e_m^t = \begin{cases} 1 & \text{Content } m \text{ is available in the RSU cache at time } t, \forall t \in T, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

Before we show how indicator e is computed, let us introduce variable h, where h denotes whether a content is removed or not at a particular time slot. Thus, when $h_m^t = 1$, $e_m^{t+1} = 0, \forall t \in T, m \in M$. Intuitively, a content can only be removed if it is already cached, Eq(3.4).

$$h_m^t \leq e_m^t, \forall m \in M, t \in T. \tag{3.4}$$

Now, in order to find the value of indicator e, we need to find how many times a content has been fetched minus how many times it was removed. The content is considered cached if and only if the number of fetches is greater than that of deletions as in Eq (3.5).

$$e_m^t = \sum_{i=0}^{I} \mathbb{1}_{\left\{ \left\lceil \frac{S_m}{U_{rate}} \right\rceil \leq \sum_{l=0}^{t} u_{i,m}^l \right\}} - \sum_{l=0}^{t} h_m^l, \forall m \in M, t \in T, \tag{3.5}$$

where $\mathbb{1}$ is an indicator which equals 1 when its condition holds and 0 otherwise. The condition of $\mathbb{1}$ tells whether enough time slots have been already allocated to fetch the content.

Throughout the operation horizon, the RSU cache may become full owing to its limited storage capacity. Moreover, due to the dynamicty of popularity of contents, the RSU may need to maintain its cache by updating its contents from time to time. Hence, when the RSU decides to cache a new content, it should remove a content or several ones in order to free space for the recently fetched one to fit in. Eq (3.6) prevents caching amounts of contents larger than the RSU cache capacity and, in addition, it allows removing content(s) to make room for other ones.

$$\sum_{m=0}^{M} (S_m e_m^t - S_m h_m^t) \leq C, \forall t \in T. \tag{3.6}$$

3.3.2 Resource Allocation

First, in the uplink, to avoid collisions, a time slot should only be allocated for no more than one vehicle. Moreover, we assume only one content can be transmitted during a time slot.

$$\sum_{i=0}^{I}\sum_{m=0}^{M} u_{i,m}^{t} \leq 1, \forall t \in T. \tag{3.7}$$

Furthermore, the RSU can only communicate with vehicles within its range.

$$u_{i,m}^{t} \leq \Gamma_{i}^{t}, \forall t \in T, i \in I, m \in M. \tag{3.8}$$

In addition, a vehicle can solely send contents which are available on its cache.

$$u_{i,m}^{t} \leq X_{m}^{i}, \forall t \in T, i \in I, m \in M. \tag{3.9}$$

Similar to the uplink, a vehicle can be served if enough time slots are allocated to the downlink and the requested content was cached in the RSU.

$$k_i = \begin{cases} 1 & \lceil \frac{S_m}{D_{rate}} \rceil \leq \sum_{t=0}^{T} b_i^t, \forall i \in I, Y_m^i = 1, \\ 0 & \text{otherwise}, \end{cases} \tag{3.10}$$

where b is a decision variable for the downlink resources and can become 1 only if the content is cached and the requester vehicle is within the range of the RSU at that time slot as in Eq (3.11).

$$b_{i}^{t} \leq \sum_{m=0}^{M} Y_{m}^{i} e_{m}^{t} \Gamma_{i}^{t}, \forall i \in I, t \in T. \tag{3.11}$$

Again, to avoid collisions, at each time slot, only one downlink transmission is allowed.

$$\sum_{i=0}^{I} b_{i}^{t} \leq 1, \forall t \in T. \tag{3.12}$$

3.3.3 Objective

Finally, the objective of this work is to maximize the served amount taking into consideration the incentive paid for cooperative vehicles.

$$\begin{aligned} \max_{u^t_{i,m}, b^t_i, h^t_m} \quad & \sum_{i=0}^{I} \sum_{m=0}^{M} (k_i Y^i_m S_m - \sum_{t=0}^{T} w u^t_{i,m} U_{rate}), \\ \text{s.t.} \quad & Eq(3.1), Eq(3.4), Eq(3.7), (3.8), (3.9), (3.11), (3.12), \end{aligned} \tag{3.13}$$

where w is a weight for the cost of fetching contents from vehicles. In other words, it is the value measuring the loss of taking one data unit from a vehicle to the RSU with respect to the gain of serving one data unit to a vehicle. The rationale behind proposing w is to prevent frequent fetches which could cause instability in the system and waste resources and time. Also, offering rewards to the vehicles motivates them to cooperate with the RSU operator. The first part of Eq(3.13) represents the amount served through the RSU cache where k_i denotes whether a vehicle is served or not. The second component, $\sum_{t=0}^{T} w u^t_{i,m} U_{rate}$, represents the cost paid in return to fetching contents from vehicles. Here, the decision variable $u_{i,m}$ determines how many time slots are scheduled for vehicle i to fetch from. Moreover, $u_{i,m}$ is multiplied by U_{rate} where the latter is the average uplink data rate. Then, the outcome of this operation represents the amount fetched from vehicle i. If this amount is no less than the size of the content intended to be fetched, we can assume that the content is safely received and cached on the RSU as is Eq(3.5). In return, the RSU pays a cost corresponding to the total amount fetched; in this case, it is the content size.

Looking at the elements of the problem, one can see that there are a number of factors that have influence on the caching decisions. For example, content size and content popularity. These two factors highly influence the caching decisions. Popular Large-size contents generate more revenue once served, yet, take more space on the cache and have lessen probabilities of being served or fetched within the limited residual time of vehicles. Meanwhile, less popular contents with smaller sizes may provide spaces in the cache unit to store more contents. Besides, small-size contents can be served and fetched easier. Furthermore, in practice, it is not logical to assume the values of Y and X matrices are known a priori. First, the massive amount of content generated every minute and computing the popularity for each one would add an enormous overhead on the network. Second,

in highly dynamic environment such as roads, estimating content popularity for a long run may not be prudent. Thus, we put forth a deep reinforcement learning model, which is kind of approximated RL, to manage the trade off between hit rate and fetching costs [54].

It is worth noting that DRL convergence speed depends highly on the complexity of the environment and that includes mainly state space and action space. Here, in order to avoid having large action space which includes all the sup-spaces, uplink, downlink and cache replacement, we will use DQN only to solve for the uplink decisions. Uplink actions are the key in our system since it decides which content will be cached. Consequently, the two other actions will be solved via alternative methods which are laid out in the following section.

3.4 Deep Q Networks

Generally speaking, Q-learning does not account for the transition probabilities from one state to another. Rather, this method calculates the expected reward of each state-action combination:

$$Q : S \times A \rightarrow R. \tag{3.14}$$

Deep Q Networks (DQN), on the other hand, is used to overcome the limitations of Q-learning [55]. Many applications such as video games, autonomous vehicles, UAVs, and so forth, evolve thousands of states and actions in reality. Therefore, curse of dimensionality is a serious problem with many RL use cases. Thus, neural networks come in handy where deep learning is used to approximate the Q function (a function computes the expected reward of taking an action in a particular state) for indefinite number of states and actions [56]. The Q function is approximated by deep neural network (DNN) where the objective of DNN is to deal with the increased number of states and reduce the amount of time required for exploration. DQN has been proposed in the literature to solve some problems related to vehicular networks due to its ability to deal with uncertainties and dynamic nature of such environment [57–59]. Fig. 3.2 shows how DQN can be implemented to help in reducing complexity in our system. Here, each action has a Q-value (the expected return of an action in a given state) and the next action is determined by the maximum Q-value.

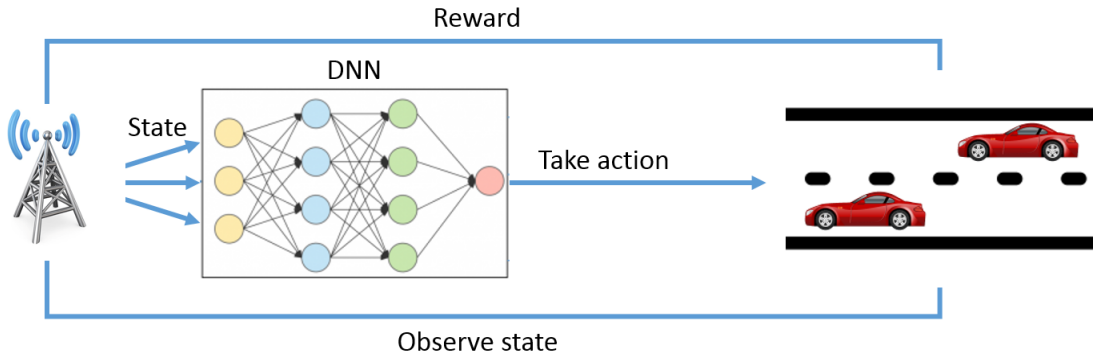


Figure 3.2: Deep Q Networks Structure

3.4.1 DQN Implementation

The Markov decision process is represented by a tuple (S, A, γ, P, R) where [60]:

- S is a set of states, also known as state space, that includes all the possible states $s_t \in S$ at any time slot t.
- A is a set of possible actions, also known as action space, that agent can take at each time slot t which is denoted by a_t.
- γ is the discount factor satisfying $0 \leq \gamma < 1$ and it specifies how much the decision maker cares about rewards in the distant future relative to those in the immediate future.
- P is the transition probability of being in next state given the current state and current action $Pr(s_{t+1} \mid s_t, a_t), \forall s_{t+1}, s_t \in S, a_t \in A$.
- $R : S \times A \to \mathbb{R}$ is a reward function where $r_t = r(s_t, a_t, s_{t+1})$ denotes the single-step reward of the system for transitioning from state s_t to state s_{t+1} due to action a_t.

Given the above mentioned MDP, we explain the action, state, and reward as follows:

- State S: The state at time slot t, $s_t \in S$, is defined as:

$$s_t = (\mathbf{X}_t, \mathbf{Y}_t, \mathbf{S}_t^{\text{available}}, \mathbf{S}_t^{\text{requested}}, \mathbf{P}_t, \mathbf{E}_t^{\text{cached}}, \mathbf{S}_t^{\text{cached}}), \tag{3.15}$$

where $\mathbf{X}_t$ is a vector that contains the indices of available contents on vehicles present at time slot t. $\mathbf{Y}_t$ is a vector that contains the indices of requested contents by vehicles present at time

slot t. $\mathbf{S}_t^{\text{available}}$ is a vector that contains the size of available contents on vehicles present at time slot t. $\mathbf{S}_t^{\text{requested}}$ is a vector that contains the size of requested contents by vehicles present at time slot t. $\mathbf{E}_t^{\text{cached}}$ is a vector that contains the indices of the cached contents at time slot t. $\mathbf{S}_t^{\text{cached}}$ is a vector that contains the size of cached contents on the RSU at time slot t.

- Action A: The action taken at time slot t, $a_t \in A$, is an integer number refers to the vehicle index ($i \in I^t$, where I^t is a set of vehicles available within the RSU coverage at time slot t) which is scheduled to upload its content to the RSU at time slot t[1].

- Reward: The immediate reward, r_t, is the sum of positive rewards (for each data unit delivered from the cache) and negative rewards w (for each data unit the RSU fetches from the coming vehicles) where w denotes the proportion of cost of fetching content from vehicles versus the reward of serving contents to vehicles.

The ultimate objective of the agent is to maximize the total revenue which is defined as the total amount served minus the total cost spent to fetch contents from vehicles as demonstrated in Eq (3.13). As shown in Algorithm 4, the agent first initializes target and value networks where the latter parameters are copied to the target network every Z (a hyperparameter) steps to overcome the problem of instability due to continuous changing with iterations. Next, the algorithm iterates until the system converges. At each iteration, a new service period is generated with different values (set of vehicles and contents). Each iteration consists of three steps, first the agent takes action according to the ε probability, either explores or exploits. We use ε-greedy as exploration method where an exploration parameter determines how much to explore versus how much to exploit. The value of ε decreases with the iterations until it reaches its minimum. Thus, we define ξ as a decay and ρ as minimum value for the ε.

After executing the action, the algorithm/agent observes the new state and the reward incurred by the action. The reward is computed based on two parts; Λ_t which is the amount served at that time slot (based on Algorithm 5) and Υ_t which indicates the amount fetched at the same time slot t multiplied by the weighting factor w. Meanwhile, if the cache is full and a new content is received,

[1] In order to avoid having varying number of actions among the time slots, we assume the number of actions is fixed and it is not less than the maximum number of available vehicles

Algorithm 4: Deep Q Networks for Content Fetching and Resource Allocation

Inputs: w, T, M, ξ, ρ, Learning Rate, Discount Factor.
Outputs: Cache and Resource Scheduling Policy.
Initialize replay memory D of C capacity.
Initialize value network with random weights θ.
Copy value network θ to create the target network θ'.
for *M iterations: Generate new service period* **do**
    for $t = 0$ ***to*** T **do**
        Set $\varepsilon = \varepsilon.\xi$.
        Set $\varepsilon = max(\rho, \varepsilon)$.
        With probability ε select an action a_t.
        Or select $a_t = max_a Q^*(s_t, a; \theta)$.
        Execute action a_t.
        if *Content m is fully fetched* **then**
            Set $e_m^{t+1} = 1$.
        if *RSU cache is full* **then**
            Delete least used item(s).
        Run Algorithm 5 and collect immediate downlink reward (Λ_t).
        Calculate r_t.
        Observe next state s_{t+1}.
        Store transition (s_t, a_t, r_t, s_{t+1}).
        Set $s_{t+1} = s_t$.
        Sample random minibatch of transition (s_t, a_t, r_t, s_{t+1}) from D.
        Perform a gradient descent.
        $\theta' \leftarrow \theta$.

the RSU removes the least used content(s) from the cache. Next, the agent gathers and stores the new sample in the buffer. Then, the agent takes random sample from the minibatch to update the value and target network. The loss function (mean squared error between the target and predicted value) is formulated and, then, a gradient descent is performed to minimize it and update the value network. Finally, every Z steps, the target network is updated by the trained value network.

Algorithm 5 is used to allocate resources for the downlink. This algorithm selects vehicle Π which should be not fully served, within the range of the RSU, and requested an available/cached content. Then, it commits to serve vehicle Π until fullness or departure. In case vehicle Π is fully served, the algorithm chooses another one which satisfies the conditions above. The advantage of

Algorithm 5 is that it does not allow sparse resource allocation which may lead to improper service experiences. That is, the RSU serves several vehicles partially with no revenue instead of focusing on some vehicles which can commit to serve them in whole and make a good revenue.

Algorithm 5: Commitment-based content service

Inputs: I^t: set of vehicles within RSU range at t.
Outputs: $\Pi = -1, \Lambda_t = 0$
if $\Pi = -1$ *or* $\Pi \notin I^t$ *or* $e^t_{Y^{\Pi}_m} = 0$ **then**
    for *each* $i \in I^t$ **do**
        if $e^t_{Y^i_m} = 1$ *and i not fully served* **then**
            Set $\Pi = i$
if $\Pi \geq 0$ **then**
    Increase vehicle Π service by D_{rate}
    if *Content is fully served to* Π **then**
        Calculate downlink reward Λ_t
        Set $\Pi = -1$

3.4.2 Complexity Analysis

This section discusses the complexity of DQN. The total computational complexity for DNN is $O(Kd_1 \sum_{g=1}^{G-1} d_g d_{g+1})$ where K is the size of the input layer. G is the number of layers and d_g denotes the number of units in g'th layer. In addition, the complexity of training one minibatch is $O(MTD)$ where M, T, and D represent number of iterations, steps, and forward/backward propagation, respectively [61]. The complexity of Algorithm 5 is $O(\mathbb{A})$ where $\mathbb{A}$ denotes the maximum number of vehicles present simultaneously in one time slot inside the range of the RSU.

3.5 Simulation and Numerical Results

3.5.1 Simulation Setup

We perform our simulation using Python and TensorFlow v2. The DQN model consists of two layers each one has 200 nodes. Moreover, we resort to Adam optimizer which is designed to train deep neural networks. In order to demonstrate the effectiveness of our DQN-proposed solution, we compare it with two baseline methods namely, random actions (Random) and minimize content

size (MIN). Noteworthy, there are no other works in the literature that study fetching and caching contents on a disconnected-RSU.

- Random: selects random actions at each time slot.
- MIN: minimizes the size of downloaded contents. For example, if the available content is smaller than the largest cached content, then the action is to take the content.

The RSU cache size is limited, hence, after it becomes full, the RSU should replace contents for any further fetching. The replacement technique we use in this work is Least Recently Used (LRU). In this technique, whenever the cache needs to free space, it starts removing the least recently used contents and stops when a sufficient space is made for the new content to be cached. The simulation is built and performed on a virtual environment, while the results are collected for 15 hours of operating time. For the content popularity, as it is aforementioned, we use Zipf distribution to generate popularities. Regarding the Zipf parameters, we set the skew parameter to 1.3 for the requests distribution and 1.1 for the available contents distribution. The rest of the experiment parameters are presented in Table 3.2.

Table 3.2: Simulation parameters in caching enabled content delivery

Parameter	Value
Service period	15h
Number of Contents	50
Size of Contents	[200-600] Mb
U_{rate}	100 Mbps
D_{rate}	100 Mbps
Size of RSU cache	1000 Mb
RSU Coverage Range	500 m
Density	[6-14] Veh/Km
w	[0.4-0.8]
Learning Rate	0.01
Discount Factor	0.99
Exploration	[0.01-0.9]

3.5.2 Numerical Evaluation

We start our study showing that our DQN model converges as presented in Fig. 3.3. It can be observed that at the very first iterations, the reward was very low. However, after iterating for several times, the reward starts increasing sharply until it saturates after almost 1000 iterations. It can be seen that after convergence, the cumulative reward keeps some fluctuations. This is due to the exploration which is essential to continue indefinitely in order to catch recently released contents. Furthermore, the set of vehicles changes each service time.

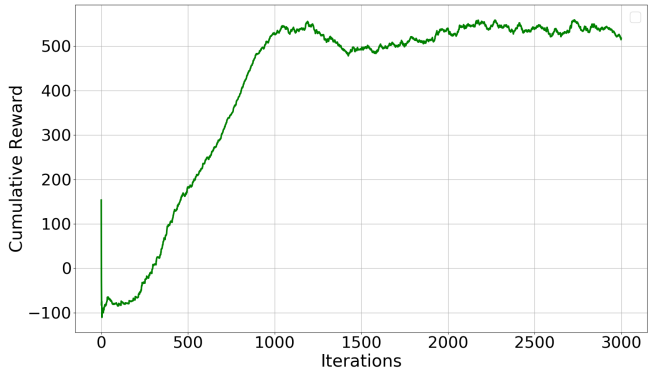
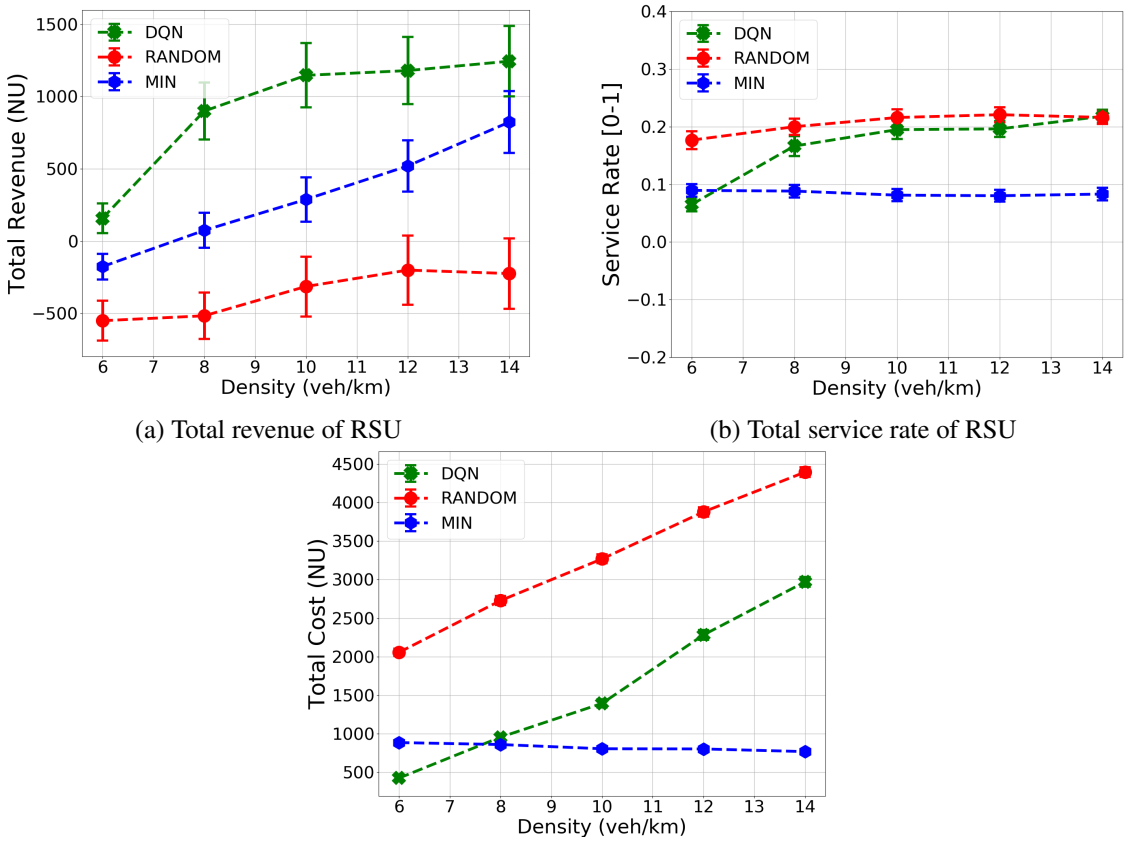


Figure 3.3: The learning curve of DQN (smoothed)

Next, the proposed system model is simulated in order to examine three essential aspects related to our DQN-based solution using various scenarios. For each scenario, we collect three performance metrics, namely the total revenue, service rate, and total cost, respectively. The total revenue is the total amount of contents served by the RSU cache minus the cost spent to fetch the new content as depicted in (3.13). In this section, we investigate the effect of different road densities, the impact of having larger RSU cache size, and finally the effect of having different weights (w). All presented results show the 95% confidence interval.

Effect of Road Density

We examine the revenue attained by our DQN-based model in comparison to the other three methods, by varying the traffic densities. As demonstrated in Fig. 3.4 (a) where the comparison is done using several densities, it can be seen that DQN outperforms the other methods in terms of



(a) Total revenue of RSU

(b) Total service rate of RSU

(c) Total cost of RSU

Figure 3.4: Comparison among the techniques using different road densities

the achieved profit. In contrast, RANDOM generates the worst reward because RANDOM does not realise the impact of assigning the resources for different vehicles within short periods which leads to the dispersion of resources. On the other hand, MIN results in lowest service rate with less than 10% service rate. This is because MIN selects small content and these contents may be unpopular. In contrast, we can also note that MIN consumes much less cost than others since the size of content it fetches is small.

At very low density (6 veh/km) the proposed method achieves around 200 NU (NU stands for normalized unit) which was the only positive revenue achieved at this density. As the density increases, the DQN model still outperforms the other methods achieving around 800 NU at 8 veh/km and almost plateaus after 10 veh/km. This is due to the constrained wireless and cache resources. It can also be observed that MIN is able to increase its revenue when the density of the road increases while the other methods do not experience a remarkable enhancement in their performance. As

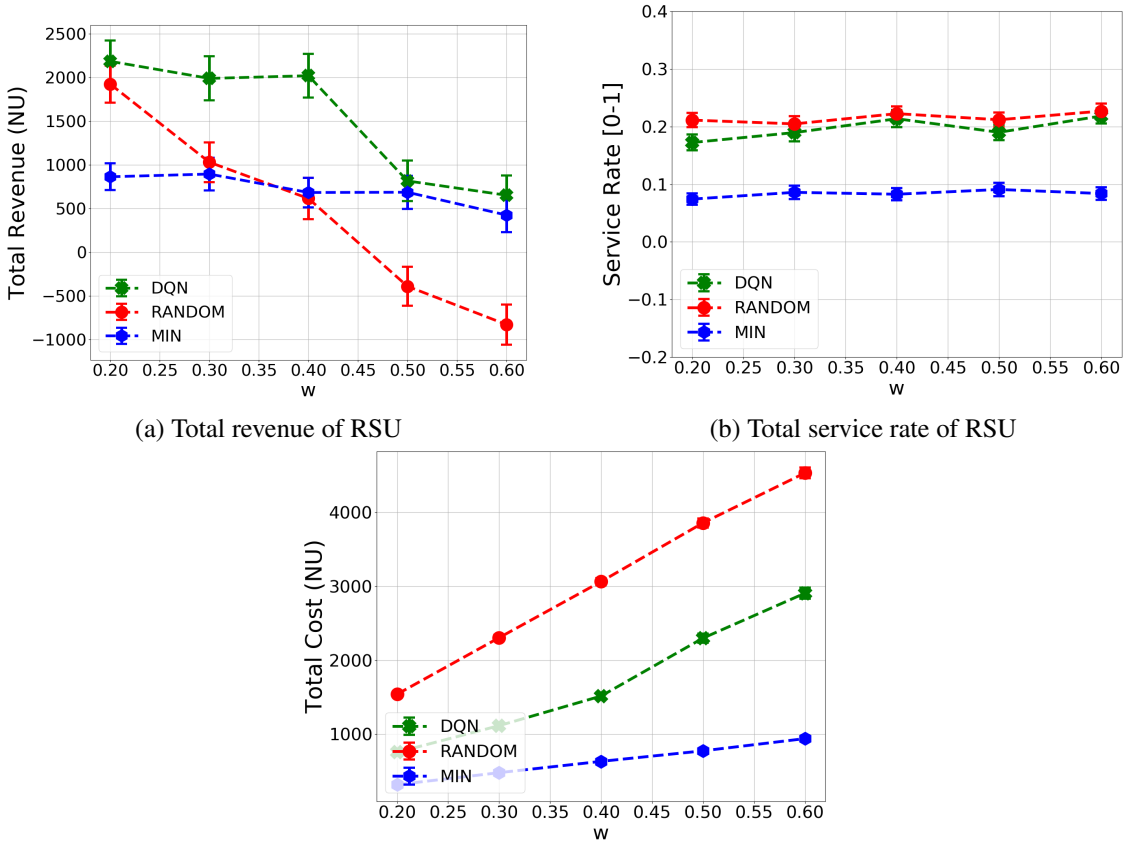
aforementioned, MIN can fit smaller contents and hence, when the number of vehicles increases, MIN can serve more vehicles.

Fig. 3.4 (b) and 3.4 (c) provide more details about how much each method is able to serve incoming requests and the total cost spent to fetch and fill the RSU cache. Fig. 3.4 (b) shows the service rate; we define the service rate as the amount of served requests over the total amount of requests. Recall that better service rate does not necessarily mean better overall performance. The overall performance depends on two factors namely, service rate and cost. Therefore, a good solution has to take care of this trade-off and try to balance for better solutions. Back to Fig. 3.4 (b), one can observe that MIN achieves less service rate than others since it picks contents of smaller size regardless of their popularity. One can also notice that at low density of 6 veh/km, DQN achieves low service rate. Indeed, as the number of vehicles is small, DQN will have less information to train on, thus, it may not find a good solution. Worth noting, in this experiment, we train our model for up to 4000 iterations for all cases. So, you can notice that as the density increases, the service rate increases as well alongside the total rewards.

Fig. 3.4 (c) demonstrates that the total cost of RANDOM increases sharply as the road density increases. It starts at around 2000 NU and end up exceeding 4000 NU. The RANDOM approach wastes cost on incomplete processes due to random fetching. MIN, however, spends less cost with around 700 NU as it fetches only small contents. Meanwhile, the cost spent by DQN exhibits an increasing behavior with respect to the road density but it remains much more cost-effective as compared to RANDOM.

Effect of Cost

Now, we study the impact of different weight values for w. As stated before, w means how much the RSU operator invests in each data unit downloaded from vehicles versus how much it gains from each data unit served to vehicles. Therefore, in this section, we study our system model using different values for w ranging from 0.2 to 0.6 with a density of 12 veh/Km. As demonstrated in Fig. 3.5 (a), the revenue generally decreases as w increases since the RSU operator spends higher cost to fetch a content from passing vehicles as w increases. The other interesting point is that DQN was able to generate more revenue even at low cost such as 0.2. Indeed, with low w, RSU



(a) Total revenue of RSU

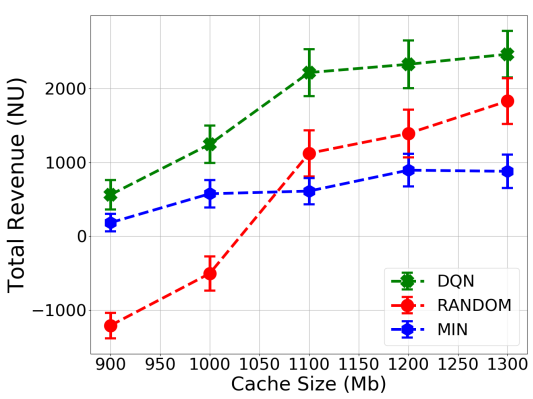
(b) Total service rate of RSU

(c) Total cost of RSU

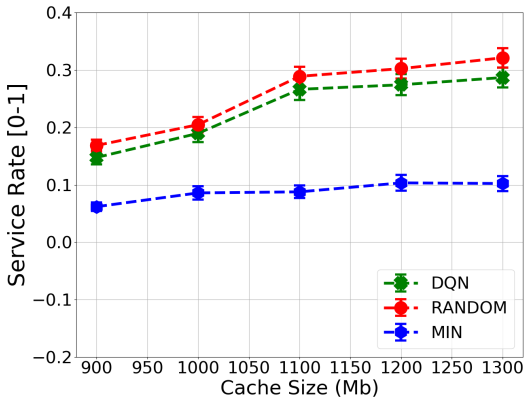
Figure 3.5: Comparison among the techniques using different w

can download contents at low prices, meaning, there is no high penalty for fetching more contents which is the major part that harms the total revenue of RANDOM. This being said, DQN can better exploit the limited cache capacity. MIN, on the other hand, does not exhibit considerable change as w increases and this is due to the fact that it fetches small-size contents that do not consume high cost.

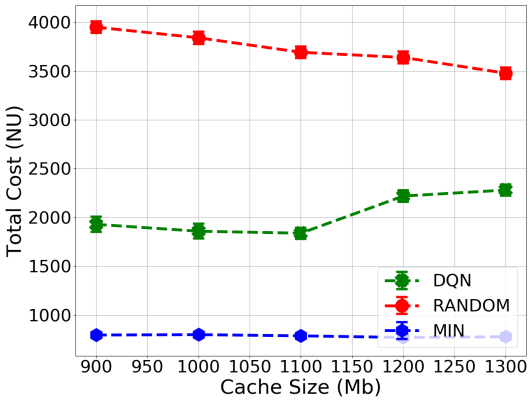
A quick look at Fig. 3.5 (b), one can see that all the methods, except MIN, provide similar service rate which is something we already justified before. Fig. 3.5 (c) shows that DQN comes in the second place with cost spent up to 3000 NU when $w = 0.6$. That means, unlike MIN, DQN is aware of the trade-off between the service rate and cost; hence, it slightly increases its costs in order to maintain good service rate.



(a) Total revenue of RSU



(b) Total service rate of RSU



(c) Total cost of RSU

Figure 3.6: Comparison among the techniques using different cache capacities

Effect of Cache Size

The capacity of RSU cache has a great impact on the system, it affects the service rate as well as cost at the same time. Larger cache can store more contents to be used to serve vehicles, while it reduces the amount of spent cost as less updates will be needed. In this experiment, to facilitate the simulation, we use a road density of 12 veh/km. In Fig. 3.6 (a), one can observe that at low cache capacity, the difference among the four methods is more obvious. Having content of different sizes, small cache size may accommodate up to one content and, hence, the different methods exhibit high differences among them depending on the cached content. However, as the cache capacity increases, more contents will fit at the same time. Therefore, not only the service rate increases, but also the performance of all methods seems to enhance as all of them use LRU to replace contents with a remarkable advantage to DQN. Hence, we can picture that the most requested contents will stay in the cache and only the least popular contents will be replaced when the size of the cache is large enough. It can also be observed that after the cache size exceeds 1100 Mb, the change on the reward becomes less obvious due to the characteristics of Zipf distribution.

Next, in Fig. 3.6 (b), it can be observed that RANDOM and DQN give almost similar service rates that increase as the cache capacity grows bigger. They all start below 20% while end up around 30% when the cache size reaches 1300 Mb. DQN seems to know how to use this extra space offered when the cache capacity increases as it improves the service rate while maintaining almost similar costs with a slight increase.

The takeaway from the results shown above is that our solution approach based on DRL principles can strike a balance between the cache hit ratio and cost spent to fetch contents. In details, our proposed solution attains the highest revenue while keeping the service rates and penalties at acceptable levels in most cases.

3.6 Summary

In this chapter, we presented a system design that maximizes the total revenue for RSU operators without accessing backhaul links. Instead, the RSU makes use of the contents already cached by vehicles to populate the RSU cache. We mathematically formulated the problem of RSU caching

and demonstrated its challenges in dynamic environments with varying contents. Consequently, we leveraged deep reinforcement learning to propose and develop a caching and resource allocation strategy at the RSUs that can adapt to a highly changing environment. The proposed solution utilizes a DQN-learning model that caches based on the utility of contents in a certain period (service period) and it is demonstrated through extensive simulations to outperform other alternative methods.

Chapter 4

Trajectory Planning and Cache Management of Aerial RSUs

4.1 Introduction

4.1.1 Preliminaries and Motivation

Building upon our previous contributions in Chapter 3, we extend that work considering more advanced aerial RSU mounted on an unmanned aerial vehicle (UAV). Owing to their agility and flexibility, UAVs can be deployed to assist a VANET infrastructure by providing vehicles' users with the same services (e.g., infotainment, road safety and assistance, content delivery, etc.) or could help vehicles when the infrastructure is not available [62]. As opposed to Base Stations (BS) and Road Side-Units (RSU), a UAV is mobile and, therefore, can follow moving objects, i.e. vehicles. A UAV can plan its trajectory to get closer to those vehicles that need to establish connections.

The network operator dispatches a UAV to help serving end users in vehicles demanding contents from the network/content provider. We can assume each UAV is augmented with a cache that enables it to store, carry and deliver contents to the vehicles. Caching is shown to have good impact on the network by reducing the access on the backhaul [63–65] through averting repeated fetching for similar contents and remarkably lowering latency down.

Now, unlike User Equipment (UEs) such as smartphones which are limited by size and capacity,

vehicles have sufficient energy and they can carry large components, i.e. cache and processing units. It is anticipated that new smart vehicles, especially those which are equipped with OnBoard Units (OBUs), would have caching capabilities. Hence, when contents are downloaded, it is preferred to keep them in their caches for a little while to prevent redundant requests. In fact, the prevailing trend of vehicle users is to visit several zones in their journey. Sometimes, they send requests for diverse contents from different entities, e.g., BS, RSU, Access Points, and others. A UAV can make profit out of this by harvesting these contents from the passing by vehicles in order to serve other vehicles which demand similar contents.

Serving contents to non-stationary entities through cache-augmented UAVs given the uncertainties of the traffic and requests remains unaddressed [66]. The existing works have either overlooked caching techniques to offload traffic or the energy consumption incurred by the UAVs movement. Therefore, in this chapter, we aim at addressing this problem and solve it efficiently leveraging recently developed machine learning tools.

4.1.2 Challenges

In contrast to stationary entities such as terrestrial base stations, a UAV can control its trajectory. However, this mobility comes with a set of pros and cons. On one hand, the UAV is able to move closer to the desired vehicles to create stronger connection with higher data rates. On the other hand, as the UAV moves towards a vehicle, it may become distant from others. Hence, the pre-scheduled transmission process may falter if the process is still incomplete. Besides, the UAV consumes different amounts of energy based on its velocity. Thereby, the UAV should plan its trajectory taking into account the mobility of vehicles and the total energy consumption in order to avoid fast battery depletion and wasting its resources. Hence, finding a good UAV trajectory to deliver contents to existing and newly arriving vehicles is not a straightforward problem. To further complicate this problem, caching decisions can be difficult to make since the cache capacity is limited while the library of contents is relatively large, and the UAV is completely oblivious to vehicles' arrivals and their demands for contents.

4.1.3 Contribution

In this work, we design a platform to assist content delivery in vehicular networks. We can summarize our novelties and contributions as follows:

- We present a UAV caching based system where the UAV helps vehicle users to download their desired contents while it makes decisions to fetch contents for its own cache from the same vehicles. The proposed system also considers the limited amount of energy on the UAV and, thus, it aims at striking a balance between the traffic offloaded and energy consumption. Then, we formulate the problem of UAV trajectory, radio resources, and caching replacement mathematically as an optimization problem to find a suitable trajectory to maximize the energy efficiency of the UAV.

- Next, owing to the complexity of the addressed problem and ambiguity of input parameters, we proposed PPO algorithm towards solving the aforementioned issue of UAV mobility in an efficient manner.

- Finally, to answer the challenges of uplink and downlink resource allocation, we develop two effective yet light designed algorithms to schedule for wireless resources. Additionally, we conduct extensive tests to prove their efficiency and compatibility with PPO model.

4.2 Related Work

UAVs have been widely studied and they attract the telecommunication community. They have been used in the literature as cache units, flying base stations, and relay nodes. Despite the numerous amount of works in this domain, only few studies exist that address the problem of caching UAV to serve highly dynamic requester objects.

The study of [67] investigates into the performance of UAVs which cache, relay, and serve multimedia contents to IoT devices. The authors of this work solve the joint problem of UAVs placement and content caching to optimize wireless throughput. First, as apposed to our work, the entertainment large contents are served to stationary IoT devices. Thus, dealing with fixed or low mobility environment is much easier than dynamic one. Second, the contents are assumed of similar

size and the UAV is assumed to cache contents proactively based on the popularity and other factors. While we address a different scenario where the UAV has no access to the backhaul and, hence, it needs to fetch the contents from the passing by vehicles. In [65], a proactive caching scheme is studied where a UAV is dispatched to provide content delivery service for vehicle users in a certain area. The objective of this work is to find UAV trajectory besides content placement and delivery mechanism to optimize the overall UAV throughput taking into account its limited battery capacity. In [64], the authors present a networking framework consisting of caching UAVs which store popular contents in advance to alleviate heavy traffic on the congested backhaul. The UAVs then serve mobile users through their cache units instead of fetching the contents from the internet. The authors of [68] suggest a new architecture for UAV to deliver content in vehicular environments. Their scenario does not consider caching or library of contents, but rather one content which is requested by multiple vehicles. In [46], the authors present a solution for the joint problem of caching and resource allocation to serve fixed users from the cache unit or via server-UAV-user link. They leverage liquid state machine (LSM) to realise the content popularity distribution. The work of [66] tries to improve the QoE of wireless devices through caching UAV. As a solution, they leverage the history and information of user in order to find out his request patterns. In [69], authors use UAV to provide coverage for vehicles in a particular area where the infrastructure becomes out of service due to disaster situations. This work aims at satisfying certain QoS to the end users. The authors of [70] utilize UAVs to help the infrastructure operation taking into account the cost incurred by deploying more UAVs. Thus, it optimizes the number of UAVs dispatched in order to provide coverage for a certain region. Moreover, in [71], the authors study the communication between a swarm of UAVs moving at fixed velocity with passing vehicles and they model the average data packet delivery delay in such scenario. In addition, [72] proposes deployment of UAVs to mitigate security problems in vehicular networks where the vehicles communication with the RSU is jammed. The UAV is leveraged to relay contents or messages in case the correspondent RSU is being attacked or there is high link interference, hence, the link is obscured.

Alternatively, a plenty of works have been devoted for static infrastructure for improving content delivery in vehicular networks. The study of [34] suggests cooperative caching among a set of RSUs by replicating contents. The incentive behind this work is that RSUs located at the same

roads are highly correlated during the service of the traffic flow. Furthermore, [35] studies the stochastic delay of content delivery in cache-based system in vehicular networks. Here, a stochastic network calculus is employed to evaluate the stochastic playback delay upper bounds of vehicular video content delivery with cache-enabled RSUs. In [36], authors proposed a caching system to deliver large contents such as videos to fast moving vehicles. They addressed the efficient content delivery problems in VANET by caching popular files in the RSUs with large storage capacity. In [37], the authors solve the problem where multiple content providers (CPs) aim to improve the data dissemination of their own contents by utilizing the storage of RSUs. They used multi-object auction-based solution to get a sub-optimal solution for the competition among the CPs. Also, [33] proposed a Q-learning based caching strategy where the system predicts the the path trajectory of vehicles and based on that it decides which RSUs should cache the contents in order to reduce the latency. Some works such as [38], suggest cooperation between RSUs and vehicles where a model is proposed to determine from where to obtain the content. Moreover, the work of [73] suggests caching using different types of infrastructure entities to minimize the total delay of content delivery.

The aforesaid works have addressed the problem of content delivery through leveraging UAVs and fixed entities, however, the existing works either suggest proactive caching mechanisms (which depends on a previous knowledge regarding the environment such as request for contents and arrival rate of requesters which, most of the time, may not be provided beforehand.) or utilize the congested and hard-to-implement backhaul link [65]. Assuming a UAV with sufficient backhaul link capacity to fetch, cache, and serve looks impractical scenario for the case of UAVs as they move and, hence, cannot maintain stable connection to the backhaul especially when they are required to follow and serve moving objects such as vehicles. Most importantly, the energy consumption of the UAV was overlooked in the previous works with caching. We list the key limitations of the existing related work on UAV servicing users in table 4.1.

Finally, none of these studies has looked into how cache-equipped UAVs can offer communication and content delivery services for dark areas, which are commonly assumed to be covered through UAVs, where no internet connection is available.

Table 4.1: Limitations of existing works.

Paper	Caching	Dynamic Environment	Energy consumption
[67]	×	✓	×
[65]	✓	✓	×
[64]	✓	×	✓
[68]	×	✓	×
[46]	✓	×	×
[66]	✓	×	✓
[69]	×	✓	×
[70]	×	✓	×
[71]	×	✓	×

4.3 System Model

Consider a one-way highway segment of a certain length (G) where a standalone UAV is utilized to provide content delivery service for the visitor vehicles as illustrated in Fig 4.1. The highway segment is presumed to entrust the UAV to respond to the content requests with a cache unit mounted on the UAV having limited space (η) to fetch, buffer, and relay contents. Meanwhile, a set of vehicles (I) travels through this particular highway where each vehicle $i \in I$ buffered a content before they approach this highway segment. The incoming vehicles are also in need of contents while they are within the highway. Assume each vehicle has one content to request and this content can be downloaded while the vehicle is within the highway segment. Therefore, we assume that each requested content can tolerate time not less than that taken by a vehicle to cross the highway.

The UAV is dispatched to carry out this operation for a certain amount of time (N). The service time consists of several time slots ($n = 1, ..., N$) of length δ. Next, the details of each aspect of the system model is given in a subsection.

Remark: In this work, for tractability, we deal only with one UAV. However, the same approach may apply for several UAVs covering a certain highway. In this context, the overall network performance could be notably improved since multiple UAV are much powerful and resource-rich to serve vehicles than a solo UAV. Given that scenario, the highway can be fragmented into smaller

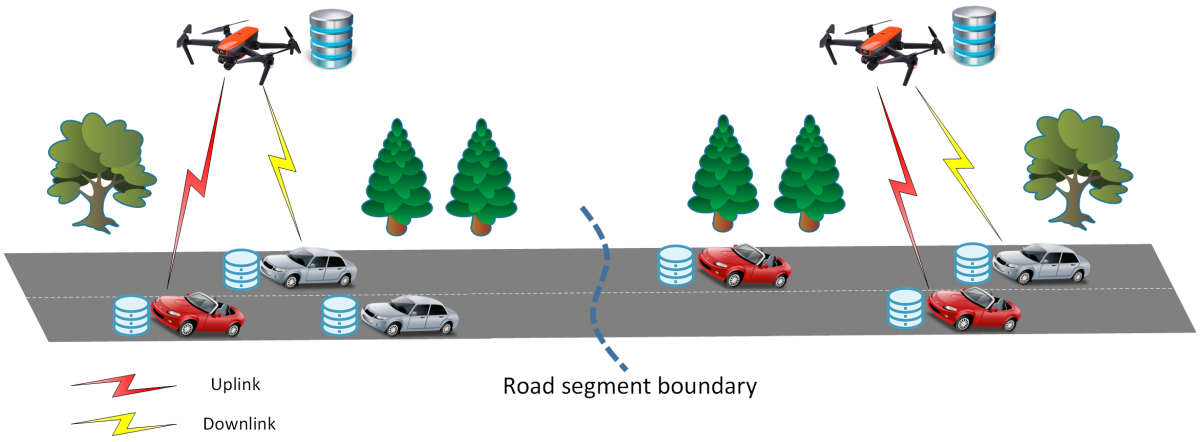


Figure 4.1: System Model

segments where there is one UAV dedicated to serve each segment separately as in Fig. 4.1. The first UAV, which is located at the beginning of the highway, may serve some of the vehicles while the remaining unserved ones will probably receive service through the consequent UAVs. The adjacent UAVs may operate on different wireless spectrum and, hence, the interference could be neglected. Interestingly, as multiple UAVs are deployed, one can leverage cooperation among them to improve cooperation among them. For instance, UAVs covering adjacent segments may get closer to each other in order to establish wireless communication and transfer contents between them. However, this is beyond the scope of this work and is kept for future research.

4.3.1 U2V and V2U Communication

We assume the communication between UAV and vehicles is established with orthogonal time division multiplexing access (TDMA). Furthermore, we assume the communication is full-duplex and over two different spectrum. For generality, we further assume the transmission power of vehicles and UAV are constant, thus, the uplink and downlink data rates (U_i^n and D_i^n respectively) vary with the distance only. At each time slot n, the UAV is either fetching a content from a vehicle, and/or forwarding a content to a vehicle, or idle.

4.3.2 Content Model

The library of contents encompasses a number of items (J) which have different sizes. The probability of content being requested is driven from Zipf distribution where the skewness parameter α_1 characterizes this distribution as in Eq(4.1).

$$\frac{\frac{1}{k^{\alpha_1}}}{\sum_{j=0}^{J} \frac{1}{k^{\alpha_1}}}, \tag{4.1}$$

where k is the item rank.

The same type of distribution models the probability of content being buffered on a vehicle, yet, with different skewness parameter α_2. Moreover, for the convenience of exposition, all contents are assumed to tolerate time larger than the travelling time of vehicles to cross the highway to avoid having varying request deadlines. Also, we assume that contents do not allow partial service. That means, in order to serve a content, it has to be fully downloaded, else, it is considered corrupted, hence, it does not count similar to [53, 74].

4.3.3 Traffic Model

To model the arrivals and velocities of passing vehicles, we assume a free-flow model similar to [51] where the relationship between velocity and traffic can be expressed as $v = v_f(1 - \rho/\rho_{jam})$ where v_f is the expected velocity, ρ is the traffic density, i.e., vehicle(s) per Km, and ρ_{jam} is the jam density (jam density is extreme traffic density when traffic flow stops completely). Moreover, the arrival rate of vehicles is calculated by $\lambda = \rho v$. The vehicle arrivals follow a Poisson distribution and velocities are generated using truncated normal distribution.

For a certain period of time (N), there is a subset of $I^n \subset I$ vehicles coming where $I = \{I^0 \cup I^1 \cup I^2 ... \cup I^N\}$. Each vehicle $i \in I^n$ has a certain velocity and arrival time (a_i) as mentioned before. Also, d_i represents the departure time of vehicle i.

4.3.4 Operation Phase and Objective

Initially, we assume the UAV is positioned at the middle of the highway segment ($w^0 = \frac{G}{2}$). When the UAV moves, it consumes energy and the amount of energy consumed with each move depends on the velocity selected at that time slot n. In practice, UAV has a maximum velocity of V_{max} which should not be violated. Also, the energy consumed for movement is modeled as non-linear function with distance traveled [75]. It must be stressed that since the energy consumed for operating the UAV is much larger than that of serving vehicle (transmission power), we only take the former into account. Besides considering energy consumption, the UAV needs to decide for the following actions:

- UAV trajectory (w^n)[1].
- Serve content to a vehicle (x_i^n)
- Fetch content from a vehicle (y_i^n).
- Caching decision or replacement policy (f_j^n).

In case that the UAV is unable to serve or fetch a content completely before the vehicle leaves, the process must be terminated[2].

Finally, the ultimate objective of utilizing the UAV is to provide service for a specific region. However, exploiting UAV comes with costs due to energy consumption. UAVs are known to be energy constrained, therefore, the UAV needs to operate smartly such that it makes the most of its energy and cache capability. In this work, we address the trade-off between content delivery and energy consumption by optimizing energy efficiency. Here, the energy efficiency is defined as the power required to transmit one data unit or the number of bits that can be sent over a unit of power consumption which is usually quantified by bits per Joule [76].

[1] Since this work concentrates mainly on improving content delivery via caching UAV and for the sake of simplicity, we assume that the altitude of the UAV is fixed.

[2] We assume once the UAV runs out of energy another one is dispatched caching similar contents of the last UAV.

4.3.5 Problem Definition

Definition 1. *Given a set of vehicles travelling through a highway covered by a UAV. Assume the UAV travelling velocity affects its energy consumption, also, the UAV is equipped with a finite cache unit. Assume the coming vehicles have one content in their buffers and raise a request for another content. Assume that the UAV has a particular service time consists of several time slots where at each time slot, the UAV can move or hover, besides, it can serve and/or fetch contents from vehicles at different data rates based on the distance. In the light of the foregoing, what is the optimal movement, serving, fetching, cache replacement actions for the UAV such that the energy efficiency is maximized.*

4.4 Problem Formulation

This section formulates the system model described above mathematically. To make it clearer and more organized, we categorize the different aspects of the system. Additionally, Table 4.2 provides a summery of the variables and parameters used in the formulation.

4.4.1 Wireless Communication

Typically, the communication channel between UAVs and end users is usually modeled as large- and small-scale fading. However, in the context of highways where there exists a clear link between transmitter and receiver, the communication link can be characterized by strong line-of-sight and, hence, small and large scale fading can be omitted [69, 70]. Thereby, the channel gain between the UAV and vehicle i at time slot n can be written as follows:

$$h_i^n = h_0 \left(\sqrt{(L_i^n - w^n)^2 + H^2} \right)^{-\tau}, \forall i \in I^n, \forall n, \tag{4.2}$$

where h_0 is the mean path gain at reference distance = 1m. H denotes the altitude of the UAV and τ denotes path loss exponent.

Let us define $P_{V \to U}$ and $P_{U \to V}$ to denote the received power of UAV and vehicles respectively.

Table 4.2: Mathematical notations in UAV assisting content delivery

Parameters	
Z_j	content size
G	highway length
J	list of contents
I	set of vehicles
N	time horizon
a_i	arrival time of vehicle i
d_i	departure time of vehicle i
L_i^n	location of vehicle i at time n
$R_{i,j}$	1: if i requests content j and 0 otherwise
$C_{i,j}$	1: if i cached content j and 0 otherwise
V_{max}	UAV maximum velocity
η	UAV cache capacity
δ	length of one time slot
τ	path loss exponent
h_0	median of the mean path gain at reference distance = 1m
σ	thermal noise power
h_i^n	channel gain between the UAV and vehicle i at time slot n
H	Fixed altitude of the UAV.
$P_{V\to U}^{i,n}$	Received power of vehicle i from the UAV at time slot n.
$P_{U\to V}$	Received power of the UAV from vehicle i at time slot n
P_V	Transmission power of vehicles.
P_U	Transmission power of the UAV.
D_i^n	Immediate data rate to vehicle i from the UAV at time slot n
U_i^n	Immediate data rate to the UAV from vehicle i at time slot n
W	Channel bandwidth available.
δ	Time slot size.
π	Air density.
v^n	Velocity of the UAV in time slot n.
Variables	
f_j^n	1: if UAV removes content j at n and 0 otherwise
x_i^n	1: if UAV serves vehicle i at n and 0 otherwise
y_i^n	1: if UAV fetches from vehicle i at n and 0 otherwise
w^n	location of UAV at n
k_j^n	1: if UAV cached content j at n and 0 otherwise
Q_i	1: if UAV served vehicle i sufficiently and 0 otherwise

Now, $P_{V\to U} = h_i^n P_V$ and $P_{U\to V} = h_i^n P_U$ where P_V and P_U are constant values representing the transmission power of vehicles and the UAV respectively.

Now, let us introduce variables D_i^n and U_i^n which are the instantaneous data rates for down and uplink scheduled to the passing vehicle i at time slot n respectively and they are calculated as follows:

$$D_i^n(x_i^n, w^n) = \begin{cases} x_i^n W \log_2\left(1 + \frac{P_{U\to V}^{i,n}}{\sigma^2}\right) & \text{if } a_i < n < d_i, \forall i \in I^n, \forall n, \\ 0 & \text{otherwise,} \end{cases} \tag{4.3}$$

$$U_i^n(y_i, w^n) = \begin{cases} y_i^n W \log_2\left(1 + \frac{P_{V\to U}^{i,n}}{\sigma^2}\right) & \text{if } a_i < n < d_i, \forall i \in I^n, \forall n, \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

where W is the available channel bandwidth (in Hz). σ is the thermal noise power which is linearly proportional to the allocated bandwidth. a_i and d_i represent the arrival and departure time of vehicle i, respectively[3]. $x_i^n \in \{0,1\}$ is a decision variable to allocate downlink resources to vehicle i. If $x_i^n = 1$ then vehicle i is receiving content from the UAV at time slot n. $y_i^n \in \{0,1\}$ is also a decision variable to schedule uplink resources to vehicle i at time slot n. Since we assume Time-Division Multiple Access (TDMA) as a channel access method for the UAV and vehicle links, then only one transmission is allowed per time slot:

$$\sum_{i=0}^{I^n} x_i^n \leq 1, \forall n. \tag{4.5}$$

$$\sum_{i=0}^{I^n} y_i^n \leq 1, \forall n. \tag{4.6}$$

4.4.2 UAV Mobility

The maximum distance the UAV can pass during one time slot should not exceed its maximum velocity.

$$|w^n - w^{n+1}| \leq V_{max}\delta, \forall n. \tag{4.7}$$

[3]The instantaneous rate depends on the instantaneous location of the vehicles. However, we are interested in the instantaneous rate of vehicles present within the highway segment, therefore, for more tractable analysis, the instantaneous rate is reduced to zero outside the given highway segment.

In addition, the movement of the UAV incurs energy consumption which depends on the velocity of the UAV similar to [70, 75].

$$P(v^n)_{total} = \underbrace{K\left(1+3\frac{M_{tip}^2}{w_b^2}\right)}_{\text{Blade profile power}} + \underbrace{\frac{1}{2}\pi(v^n)^3 F}_{\text{Parasite power}} + \underbrace{m_u g\sqrt{\left(\frac{-(v^n)^2+\sqrt{(v^n)^4+(\frac{m_U g}{\pi A})^2}}{2}\right)}}_{\text{Induced power}}. \tag{4.8}$$

where v^n denotes UAV velocity at time slot n. M_{tip} represents the blade's rotor speed, K and F are two constants which depend on the dimensions of the blade and the UAV drag coefficient, respectively, π is the air density, m_U and g respectively denote the mass of the UAV and the standard gravity, A is the area of the UAV. The total energy consumption to cover a distance d at a constant velocity UAV w can be computed as $E(v)_{total} = \int_0^{d/v} P(v)dt = P(v)\frac{d}{v}$ as in [70].

4.4.3 Cache Management

Let us define $\beta_i^n \in \{0,1\}$ as an indicator holding value 1 if the content available on vehicle i is fully fetched by the UAV before or at time slot n and 0 otherwise.

$$\beta_i^n = \begin{cases} 1 & \sum_{n'=0}^{n} \delta U_i^{n'} \geq \sum_{j=0}^{J} C_{i,j} Z_j, \forall i \in I, n \in N. \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

In addition, Eq (4.10) prevents wasting radio resources to vehicle that completely uploaded its content to the UAV.

$$y_i^{n+1} \leq 1 - \beta_i^n, \forall i \in I, n \in N. \tag{4.10}$$

Next, we introduce $k_j^n \in \{0,1\}$ as a binary variable where it is equal to 1 if content j is available on the UAV at time slot n and 0 otherwise. The value of k has three cases; it is either equal to 1 when the content is just fetched or 0 if it is removed in the previous time slot. The third case is that it remains the same if no change occurs on its value[4].

[4]For more tracebility, the initial value of $k_j^0 = 0, \forall j \in J$ since the cache is assumed empty at the beginning.

$$k_j^n(y_i^n, f_j^n) = \begin{cases} 1 & \sum_{i=0}^{I} y_i^n \beta_i^n C_{i,j} = 1, \forall n, \\ 0 & f_j^{n-1} = 1, \\ k_j^{n-1} & \text{otherwise}, \end{cases} \tag{4.11}$$

where $f_j^n \in \{0,1\}$ is a decision variable where it equals to 1 if content j is removed from the UAV cache at time slot $n+1$, 0 otherwise.

Eq(4.12) prevents violating the limited cache capacity of the UAV.

$$\sum_{j=0}^{J} k_j^n Z_j \leq \eta, \forall n. \tag{4.12}$$

A content cannot be removed from the UAV cache if it does not exist in the cache beforehand.

$$f_j^n \leq k_j^n, \forall j \in J, n. \tag{4.13}$$

4.4.4 Service Management

Before we formulate our objective function mathematically, let us define $Q_i \in \{0,1\}$ which denotes whether a vehicle has been sufficiently served or not during its sojourn time.

$$Q_i(x_i^n, y_i^n, w^n, f_j^n) = \begin{cases} 1 & \sum_{n=0}^{N} \delta D_i^n \geq \sum_{j=0}^{J} R_{i,j} Z_j, \forall i \in I^n, n, \\ 0 & \text{otherwise}, \end{cases} \tag{4.14}$$

where $R_{i,j}$ indicates whether content j is requested by vehicle i ($R_{i,j} = 1$) or not ($R_{i,j} = 0$).

In fact, the UAV cannot serve a vehicle if the requested content is not available in its cache as illustrated in Eq (4.15).

$$x_i^n \leq \sum_{j=0}^{J} R_{i,j} k_j^n, \forall i \in I^n, n. \tag{4.15}$$

4.4.5 Objective

The ultimate objective is to maximize energy efficiency (bits per Joule) and can be written as:

$$\begin{aligned} \max_{w^n, x_i^n, y_i^n, f_j^n} \quad & \sum_{j=0}^{J} \sum_{i=0}^{I} \frac{Q_i(x_i^n, y_i^n, w^n, f_j^n) Z_j R_{i,j}}{\sum_{n=0}^{N} P(w^n)}, \\ \text{s.t.} \quad & Eq(4.5), Eq(4.6), Eq(4.7), Eq(4.10), \\ & Eq(4.12), Eq(4.13), Eq(4.15), \end{aligned} \tag{4.16}$$

where Q_i is function of the four decision variables.

The presented problem is mixed integer non-linear programming (MINLP), which is known to be difficult to solve , owning to the binary variables, x_i^n, f_j^n, y_i^n and the real-value decision variable w^n, as well as, our objective function in Eq (4.16) is non-convex because of the trajectory variable in Eq (4.7) [77]. Furthermore, the solution of the problem (if exists) is dependant of the knowledge of instantaneous positions of vehicles at each time slot during their journey along the highway segment. However, in practice, the upcoming set of vehicles alongside their information (their requests and available contents in their buffers) are revealed once they approach the highway segment. Particularly, the values of parameters L_i, $R_{i,j}$, and $C_{i,j}$ remains hidden as long as vehicle i has not yet reached the highway segment. Given the high complexity of the problem and the numbers of uncertainties, definitely there is a need for alternate solution with lower complexity and high efficiency to tackle such scenario [78]. Therefore, we model our problem as MDP and, then, we propose PPO-Clip, which we will introduce next, and effective algorithms to solve for the aforementioned problem. PPO-Clip is efficient to deal with ambiguities in the environments as it is able to learn and estimate values through observations. This MDP model and PPO-Clip will be laid out in the next section.

4.5 Solution Approach

This section provides a complete solution for the predefined problem of UAV mobility and caching. Due to the intractability of the optimum problem and in order to simplify the arduous challenges addressed by this work, we divide our problem into three sub-problems. This first one

is to find trajectory of the UAV while the other sub-problems deal with wireless transmission and caching replacement.

4.5.1 PPO-Clip to Control UAV Trajectory

In this work, we resort to Deep Reinforcement Learning to solve for the trajectory decisions. Many applications such as video games, autonomous vehicles, UAVs, and so forth, encompass uncountable number of states and actions. These high-dimensional spaces create phenomena of curse of dimensionality. Precisely, we leverage Proximal Policy Optimization (PPO). PPO is a cutting-edge, benchmark, model-free, on-policy, policy gradient Reinforcement Learning algorithm designed at OpenAI [79, 80]. It is famous for its tunability besides its outstanding performance and lower complexity. In fact, the goal of PPO is to balance between implementation, batch sampling, and ease of tuning. Genuinely, PPO-Clip approximates the hard constraint applied to Trust Region Policy Optimization by using much more effortless equations. Therefore, PPO-Clip is much simpler version and demonstrates remarkable efficiency.

To go for the implementation part, the problem is formulated as MDP (similar to Section 3.4.1) where the state, action, reward, are defined as follows:

- State S: The state at time slot n, $s_n \in S$, is defined as $s_n = (w^n, D^n, U^n, L^n, C^n, R^n, Z, k^n)$. First, w^n is UAV position at time slot n. D^n is a vector that contains the amounts served to each vehicle at time slot n. U^n is a vector that contains the amounts fetched from each vehicle at time slot n. L^n is a vector that contains the amounts served to each vehicle at time slot n. C^n is a vector that contains the content requested by each vehicle at time slot n. R^n is a vector that contains the content available on each vehicle at time slot n. Z is a vector that contains the sizes of available contents. k^n is a vector that contains the cached contents on the UAV at time slot n.

- Action A: The action taken at time slot n is $a_n \in A$. In this work, in order to void the high complexity of continuous action space, we approximate the action space by descritizing the UAV velocities. Thus, our action space includes 0 speed for hovering and a certain number of velocities to step the UAV forward and backward and those velocities are predetermined.

- Reward: The immediate reward, r_n, is the sum of positive rewards due to serving vehicles sufficiently and negative reward or penalty due to consuming energy for UAV operation. It is worth mentioning that the objective function in Eq (4.16) emphasizes the total reward is a product of the division of total amount served over energy consumption. However this cannot be realised with RL. As RL calculates the immediate reward in each step, the total energy consumption value remains unknown and only reveals once the episode ends. Thereby, we modify our immediate reward to be as follows:

$$r_n = \sum_{j=0}^{J} \sum_{i=0}^{I} Q_i^n Z_j R_{i,j} - \psi P(w^n), \forall n, \tag{4.17}$$

where ψ is a weighting factor to balance the impact of the two terms. In Eq (4.17), to solve the aforesaid issue, instead of dividing the amount served over energy consumption, we subtract them. However, in order to not neglect the impact of any terms, we add ψ to tackle this issue. Now, the step-reward contains two parts; positive reward and negative reward (penalty). The positive reward is awarded to the agent when the UAV serves a vehicle sufficiently while the agent is penalized due to the energy consumed with each UAV move.

As illustrated in algorithm 6, the PPO interacts with the environment to collect the samples through several iterations (typically thousands of epochs) and realises the actual rewards.

First the PPO initializes random sampling policy and value function for the neural networks. Then, in each epoch, the agent observes the environment which consists of the set of vehicles and their availabilities, requests, cached contents, UAV position and so on. Then, an action is selected based on the policy and the action is used to move the UAV to its new position, however, the UAV should remain inside its service region. Algorithm (8) and (7) are used to realise the resource allocation among the set of vehicles present at that time slot. For the downlink, the UAV computes a positive reward if a content is served to a vehicle. In the uplink, the UAV stores a content in the cache once it is fully fetched. A replacement may be required if there is no space available to fit the new content. In each epoch, PPO computes the advantage function that helps the agent to estimate how good it performs compared to average action for a certain state. Then, the clipped objective of PPO is optimized and the policy is updated accordingly once every several epochs [80].

Algorithm 6: PPO-Clip to Find UAV Trajectory

Inputs: L_i^n, $C_{i,j}$, R_{i_j}, N, Learning Rate, γ, ε.
Outputs: The UAV velocity control policy π.
Initial policy π with random parameter θ
initial value function V with random parameters ϕ
for *each episode* $k \in \{0,1,2,...\}$ **do**
    for $n : \{0,1,2,...,N\}$ **do**
        Observe state $L_i, C_{i,j}, R_{i,j}, U_i^n, D_i^n, \forall i \in I^n$, Z_j, w^n.
        Perform action $a_n \in A$ based on policy $\pi_{\theta_{old}}$.
        Set $w^{n+1} = w^n + a_n$
        if *UAV is outside the highway segment* **then**
            Keep the UAV at the previous position.
        Set $x_i^n \in \{0,1\}$ using Algorithm (8)
        Calculate step reward as in Eq (4.17)
        Set $y_i^n \in \{0,1\}$ using Algorithm (7)
        if *vehicle i content is fully received by the UAV* **then**
            if *vehicle i content needs space* **then**
                Remove other content(s) as in 4.5-C.
            Store vehicle i content in the UAV cache.
    Compute the advantage function.
    Optimize the clipped objective.
    $\theta_{old} \leftarrow \theta$.

4.5.2 Heuristic Algorithms to Wireless Resource Allocation

The second sub-problem of UAV and vehicle radio resources scheduling is solved by two algorithms for the up and downlink.

Concerning the uplink, as laid out in Algorithm 7, the UAV sorts the contents cached based on their popularities. Next, it checks whether the content available on each present vehicle is already cached or not and whether it is downloaded earlier. If the conditions above are satisfied, then the UAV makes sure the content size will only replace content(s) which are less in size. In this step, the UAV needs to ensure that it will not require to remove contents which are more popular for those which receive less hits. The complexity of Algorithm 7 is $O(\partial J)$ where ∂ represents the maximum number of vehicles present simultaneously in the highway segment. The value of ∂ depends on the

highway density and, in reality, $\partial \ll I$.

Algorithm 7: Uplink Radio Resource Allocation

```
Inputs: n, Z_j, C_{i,j}.
Outputs: Π as the vehicle to fetch from at n.
Set Π = None.
Sort cache items by popularity
for ∀i ∈ I where a_i < n < d_i do
    if i's content does not exist in the RSU cache and the content is not fully fetched yet then
        Compute the total size of cached contents on the UAV.
        if the vehicle i content cannot fit in the cache then
            Define R = 0 as the amount to remove in order to cache the new content.
            for each content j : k_j^n = 1 do
                if vehicle i content is more popular that of Π and j then
                    Set R = R + Z_j
                    if vehicle i content size ≤ R then
                        Set Π = i
                        break
                else
                    break
        else
            if Π' content is less popular than i's then
                Set Π = i
```

For the downlink, the UAV is committed to serve a vehicle as long it resides within the highway and not fully served. The UAV also need to make sure that the distance between the UAV and the vehicle does not exceed a certain threshold (Ω) as in Algorithm 8. This condition can improve the download experiences since it prevents the UAV from sticking to only one far vehicle which will not be served after all due to far distance. In case no vehicle is being served at that time, the UAV will compare among the available vehicles based on data rates. The vehicle with higher data rate or shortest distance will be selected to download. The complexity of Algorithm 8 is $O(\partial)$.

Algorithm 8: Downlink Radio Resource Allocation

Inputs: n, Ω, Z_j, $R_{i,j}$, G (denotes vehicle currently being served).
Outputs: Λ as the best vehicle to serve.
if $G \neq None$ **then**
    if $d_G \leq n$ ***or*** $(Z_j \mid R_{i,G} = 1) \leq D_G^n$ ***or*** $|L_G - w^n| > \Omega$ **then**
        Set $G = None$
if $G \neq None$ **then**
    Set $\Lambda = G$
else
    for $i \in I$ *where* $a_i < n < d_i$ **do**
        if *i requested a content that is cached on the UAV, not completely served, has more time slots to remain on the highway segment, and the remaining time slots are enough to receive the whole content* **then**
            Set $\Lambda = i$
Set $G = \Lambda$

4.5.3 Cache Replacement

The cache replacement policy used in this work is as follows. Whenever the cache is full and a new content is fetched, the UAV starts freeing up space by removing contents which are less popular until enough room is made to store the recently fetched content.

Fig. 4.2 summarizes the solution framework and the way the different algorithms interact with each other. As PPO is built on top of actor critic, it inherits its architecture. Thus, there are two networks: actor which is responsible for generating the action, and critic which computes the advantage function. There is also a memory to store the samples of the environment to help the networks reducing the loss function.

4.6 Performance Evaluation

4.6.1 Simulation Setup

We carry out the simulation studies using Python and PyTorch. To mimic the reality, we take a highway of 2 Km where a UAV is dispatched to cover it with a 2 Gb-cache unit. For the sake of simplicity, we will do our simulation only for one UAV, however, the same can apply for several UAVs covering a large highway. The library of contents is generated having random content sizes

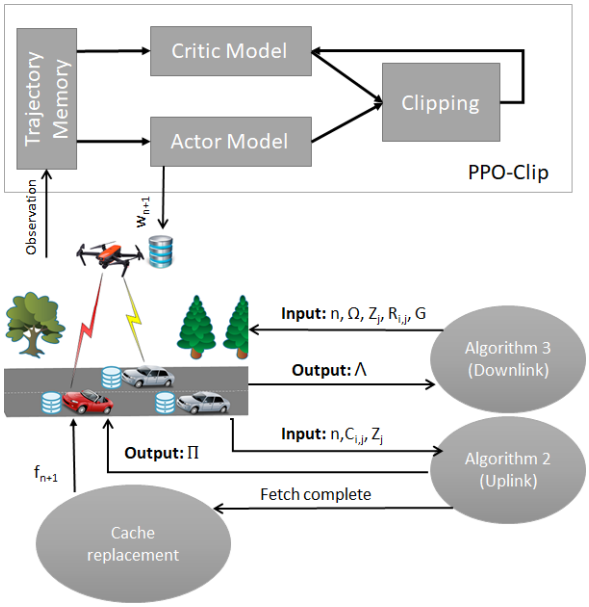


Figure 4.2: Solution approach scheme.

while the frequency of content being requested and buffered on vehicles follows Zipf distribution with skewness parameters, α_1 and α_2, equal to 1.3 and 1.6, respectively, and similar to [47]. The set of vehicles is generated at random where the arrivals follow Poisson distribution. Here, as mentioned earlier, we use free flow traffic model to general vehicle arrivals and velocities which has been used widely in the literature to simulate vehicular environment [81–83]. Meanwhile, the agent is exposed to 1,440,000 samples before results are collected where each sample represents one time slot. Then, different sets are generated for testing. For consistency, the results are averaged over 2000 iterations.

The PPO consists of 2 neural networks, actor and critic, each network has 3 layers, input, hidden, and output. The hidden layer contains 64 nodes and Adam optimizer is designed to train the DNNs. In addition, Hyperbolic Tangent is used mainly as activation function while Softmax is only used for the actor output layer. The key simulation parameters are listed in Table 4.3.

4.6.2 Baseline Methods

We study the performance of our solution approach by comparing its outputs with different methods. Since, to the best our knowledge, there is no work in the literature that addresses similar

Table 4.3: Simulation parameters in UAV assisting content delivery

Parameter	Value
Highway length	2 Km
δ	1 sec [77]
Number of UAVs	1
Number of Contents	50
Size of Contents	[700-1300] Mb
Bandwidth	10 MHz [84]
σ^2	10^{-14}
τ	3
h	10^{-15}
$P^{i,n}_{U\to V}$, $P^{i,n}_{V\to U}$	0.1 w
UAV discrete velocities	[0, 10, 20, 30, 40] m/s
Density	[2-14] Veh/Km
Learning Rate	0.002
Discount Factor	0.99
Clip	0.02

challenges of UAV mobility, resource allocation, plus cache replacement, we design 4 UAV-steering techniques to use as baselines.

- *Stationary UAV (S-UAV)*: keeps the UAV fixed at the middle of the highway segment.
- *Random UAV Mobility (RAUM)*: selects random velocity to move the UAV at each time slot.
- *Maximum Speed Selection (MASS)*: always chooses the maximum velocity for the UAV. When the UAV reaches one end of the highway segment, it goes back in the opposite direction.
- *Minimum Energy Selection (MISS)*: always chooses the velocity that infers the lowest energy consumption, for the UAV. When the UAV reaches one end of the highway segment, it goes back in the opposite direction.

It is crucial to mention that for fair comparisons, all the UAV trajectory methods suggested above are developed to work based on our proposed algorithms for the up and downlink which are explained in section 4.5.

4.6.3 Result Analysis

First of all, we examine the convergence of our model. As shown in Fig. 4.3, the PPO-Clip model converges after around 2000 iterations.

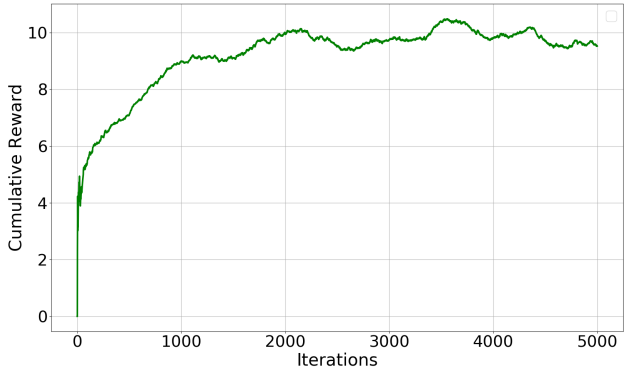


Figure 4.3: PPO-Clip convergence over iterations.

Next, to examine the efficiency of our PPO-based solution, we show in Fig. 4.4 how the performance of the PPO evolves within time until it reaches convergence state. Here, one can observe that

the solution is quite fast at the beginning where PPO reaches above 50% of the maximum performance attained in less than 5 hours and around three-quarters of the maximum performance within a day of training. In addition, we can also notice that exposing the PPO agent to more training samples may enhance the performance further until it converges after less than 6 days of training.

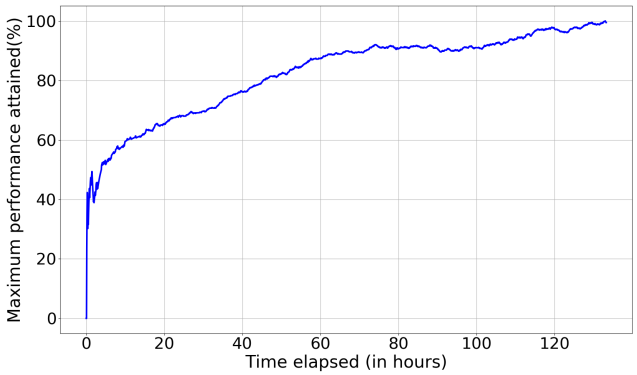


Figure 4.4: PPO-Clip performance versus time.

Before we start evaluating the PPO-based solution with other methods, it is very important to see and examine the efficiency of the up and downlink designed algorithms. In order to do that, we suggest two other algorithms to compare with, namely, Greedy and Random. Where Greedy prefers to fetch and serve vehicles that has higher immediate data rate and Random takes random actions. As illustrated in Fig. 4.5, the difference is very significant. With our proposed solution, there is around 50% difference between the proposed algorithms and Greedy. In addition, one can notice that Random does not exist in this figure. Actually, Random produces 0 gain in every attempt. This is because that it cannot complete any transmission due to the randomness in resource allocation. It can also be observed that MISS and MASS have higher amounts of contents served with Greedy method. This can be justified as follows: MASS and MISS are unable to serve or fetch large contents because they do not catch up with vehicles and their connection is not stable as in RAUM and S-UAV. However, with Greedy, it is not mandatory to fetch popular contents which might be large. Thus, MASS and MISS are able to make some gain by fetching and serving smaller and probably unpopular contents.

In terms of evaluation, we compare these four methods with our PPO-based solution approach in terms of the energy efficiency level, amount served to vehicles, and energy consumption.

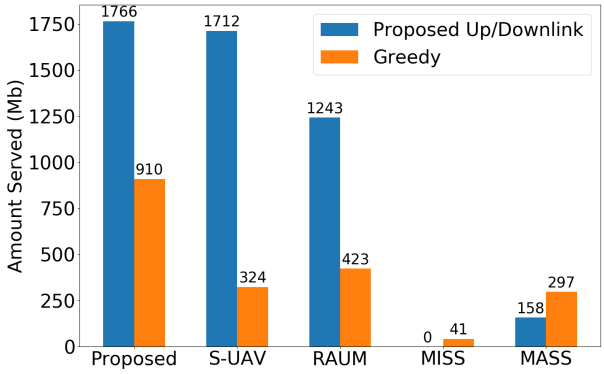


Figure 4.5: Proposed wireless allocation algorithms vs Greedy wireless allocation.

First, Fig. 4.6 demonstrates the energy efficiency levels achieved by the five methods. As it can be seen, the values of our proposed solution in addition to S-UAV and RAUM are much higher than MISS and MASS. The reason behind this is that MISS and MASS highly deviate from their regions and, thus, most of their transmission processes are interrupted before they mange to complete. In general, the energy efficiency grows as the number of vehicles increases. Such behavior is normal given that as the density of the highway increases, there will be more chances for the UAV to serve or fetch content using the same amount of energy. We can also observe that the proposed solution scores the highest levels in all cases with a remarkable differences from others. Indeed, it outperforms S-UAV and RAUM by around 20% to 30%. Meanwhile, S-UAV comes in the second place with a slight difference from RAUM. As S-UAV keeps the UAV stable in the middle of the highway, it can attain some stability to the wireless links. However, this method fails in some cases when the content is large and the UAV needs to move toward the vehicle in order to save the link. Moreover, as Eq (4.8) infers, hovering is not the best choice for power consumption. Indeed, as Eq (4.8) is non-linear, velocity of 25 m/s consumes the lowest amount of energy. Likewise, RAUM almost obtains identical performance values to S-UAV. Here, we use uniform random distribution, therefore, the UAV goes back and forth over the middle of the highway segment and does not move away from the center.

As shown in the same figure, MISS was the worst technique with very close to 0 energy efficiency regardless of the highway density. The reason behind that is MISS moves the UAV very slow while vehicles move much faster, thus, it will never catch up with vehicles to serve or fetch from

them.

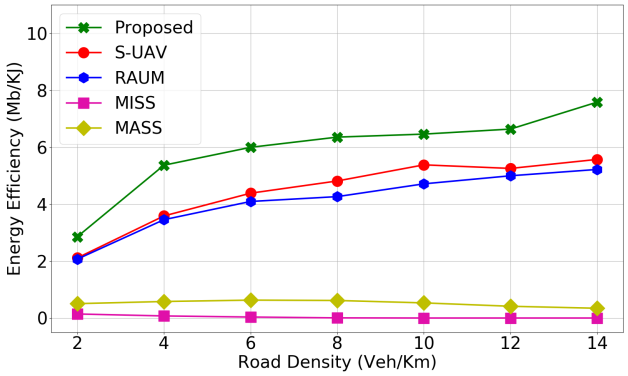


Figure 4.6: Impact of road density on energy efficiency.

Although the amount served to vehicle is not the sole goal of this work, one can note that the PPO-based solution manages to serve more amount of contents than others. In Fig. 4.7, we can see the total amount served resulted from using the proposed method and S-UAV is 15-35% higher than RAUM and 95% higher than MISS and MASS. The reason behind this good performance of S-UAV is, as mentioned above, owing to its fixed location. We can also notice that the proposed solution starts by 750 Mb at low density while ends up with more than 2000 Mb of content served. That is, the density of the road can notably increase the the service amount and the reason behind this is aforesaid. Moreover, along the y-axis, the proposed solution comes at first place.

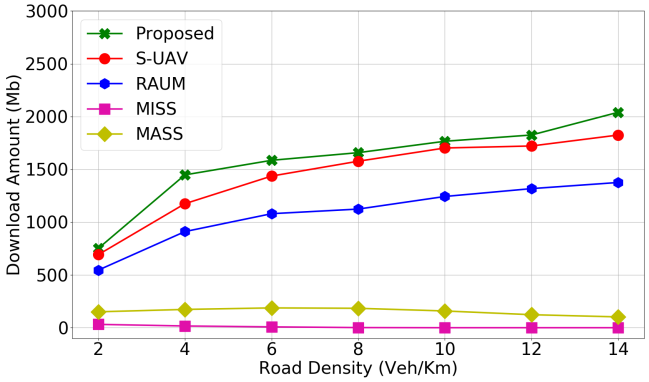


Figure 4.7: Impact of road density on amount of content served.

Next, Fig. 4.8 shows how much energy each method can incur. Based on energy consumption

function in Eq (4.8), the amount of energy consumed is correlated with the traveled distance. However, this relationship is non-linear. Hovering consumes higher energy than low velocity movement. Thus, we can see that S-UAV incurs high energy consumption at around 350 KJ. In contrast, MISS wastes less amount of energy at less than 250 KJ since it uses velocity that incurs the lowest amount of energy. Meanwhile, one can also notice that the proposed solution consumes moderate amount of energy at a bit above than 250 KJ. Taking the amount served by the proposed solution shown in Fig. 4.7 and the amount of energy consumption in this figure, it becomes very clear that the proposed solution is actually the best one.

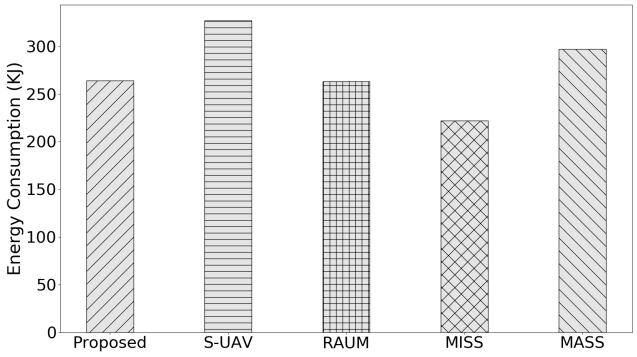


Figure 4.8: Amounts of energy consumption incurred using the five methods (per 4 minutes).

Next, we investigate into the role that cache capacity can play in improving the efficiency of the system. As it can be seen in Fig. 4.9, the energy efficiency starts at almost 0 level when the cache unit is very small to fit only one tiny content. The level of energy efficiency jumps surprisingly to 3.7 at cache capacity 1200 Mb. This increase continues as the cache unit grows. However, at some point it saturates. Actually, that depends on the shape of popularity of contents. In this work, as we use Zipf distribution to model the frequency of requests, the top popular contents receive much higher hits than other. Therefore, once these contents can fit in the cache, the extra space will have very little effect. One can also observe that the proposed solution comes in the first place at much higher performance.

Now, let us see the impact of Zipf parameter (α) on the performance. α characterizes the distribution where large values of α means the top popular contents will receive much higher hits than others and vise versa. It can be observed in Fig. 4.10 that as α increases, the performance

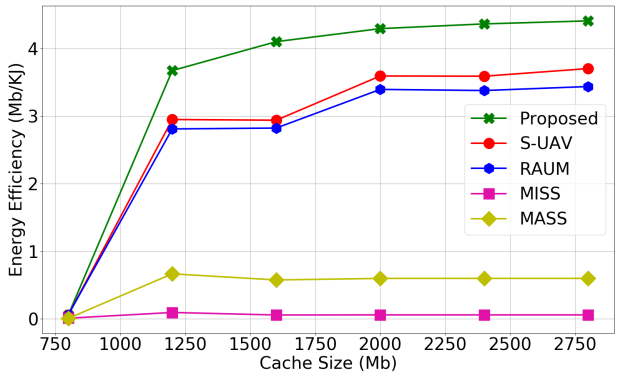


Figure 4.9: Impact of cache unit capacity on energy efficiency.

of S-UAV, RAUM and besides the proposed solution demonstrate better results. Indeed, as we mentioned earlier, when the α becomes larger, the UAV can serve more vehicles as the overlap over requests raises. We can also note that the proposed solution's performance becomes much higher than others as α increases. That means the PPO-Clip can better adapt and take advantage of the identical requests than other methods which are blind in this regard.

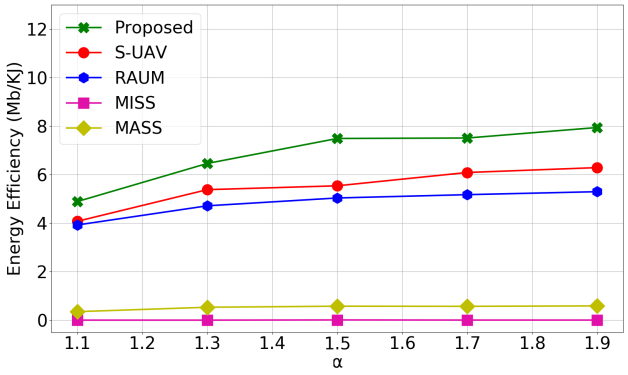


Figure 4.10: Impact of Zipf skewness parameter on energy efficiency.

4.7 Summary

In this work, we presented a novel UAV-assisted content delivery in VANETs without having to be connected to the internet. The system model, which is mathematically formulated, is solved

efficiently using Deep Reinforcement Learning technique namely, PPO-Clip with two designed algorithms. The solution method is put to test against other baseline methods to examine its adequacy. The results illustrate that even UAV without internet connectivity is able to contribute in serving contents to vehicles through taking the chance of collecting contents from the upcoming vehicles.

Chapter 5

Cooperation between Terrestrial and Aerial RSUs for Optimized Content Delivery

5.1 Introduction

5.1.1 Motivation

In Chapters 3 and 4, we presented two kinds of caching based systems to serve vehicular networks; the first one is stationary RSU and the second one is flying UAV. In this chapter, however, we attempt to mine the benefits of the two scenarios by allowing cooperation between the stationary and aerial RSUs. In general, vehicle edge caching has been widely introduced in the literature to improve content delivery services in vehicular networks [11, 85]. Caching can alleviate traffic loads and reduce latency across the network by avoiding duplicate content retrievals and transmitting contents form nearby units. Drones, or UAVs, on the other hand, are considered as appealing solutions to ameliorate vehicular network performance and overcome the key issues raised earlier, particularly with having backhaul bottlenecks. Given their agility, flexibility, and deployability, UAVs are seen as a solid candidate to help in offloading data traffic from the vehicular networks. As opposed to terrestrial infrastructure, UAVs can better communicate with vehicles as they possess the ability

to fly and follow moving objects. UAVs are also known to have much efficient links with higher probability of creating Line of Sight (LoS) since they can adjust their positions to establish better links by reducing distances and avoiding obstacles such as skyscraper and towers. Therefore, in this work, we leverage UAVs and caching principles to design a new paradigm where a UAV is deployed to cooperate and assist a RSU in serving contents.

Building upon our previous work [86], we further extend that work by considering a cooperative cache-based system. In this chapter, we address the issue of accessing popular contents for in-vehicle users. More specifically, we assume that a RSU and a UAV co-exist to serve contents to vehicles navigating on a road segment. The UAV acts as a complement infrastructure to the RSU in a sense of completing services for partially served vehicles and/or probably provide full service for un-served ones. To do so, the UAV is assumed to be equipped with finite cache capabilities to store and serve contents. In addition, due to the complexity of establishing an efficient backhaul link to the UAV, we come up with an alternative method [87]. In a nutshell, we design a relay scheme where the RSU can populate contents on the UAV over the passing vehicles in push-carry-forward fashion. In this way, the loads can be reduced on the backhaul besides maintaining the content list stored on the UAV during the operational phase.

5.1.2 Challenges

In this work, we identify three main challenges. The first challenge is due to radio resource scheduling. For example, when the RSU serves a vehicle, it has to consider the ability of the UAV to complete the remaining part. This includes answering two questions; whether the requested content is cached on the UAV and whether the UAV has enough resources (spectrum and time) to serve the rest of the contents. The latter question is based on the UAV status (e.g., number of vehicles within its coverage and still need service). More importantly, the RSU should not reject requests while wasting its resource to serve vehicles which can be served through the UAV at a later time. The RSU may also want to populate contents on the UAV cache by serving some vehicles till completion which can subsequently forward/shuttle these contents to the UAV. The second challenge is related to the UAV. The UAV needs to plan its trajectory, radio resources, and decide for contents placement (which contents and how much to cache for each one). The UAV should take into account popularity

of contents, and vehicles status (how much each vehicle has been served through the RSU). If a vehicle has been already served partially through the RSU and the UAV is capable of completing the service, the UAV has to satisfy that vehicle. Otherwise, the service is rejected and the RSU would have wasted resources on incomplete tasks. Hence, the overall performance of the network degrades. A third challenge appears as a result of the unknown parameters of the environment. There is a number of uncertainties in our context including the arrivals of upcoming vehicles, their velocities, and their requests and such information is not given in advance.

In general, the interplay between the RSU and the UAV is of utmost importance. If it is not handled properly, there will be many vehicles leaving the road segment without being served sufficiently. Thus, in order to tackle the challenges mentioned above, we propose a solution approach, Dual-Task Deep Reinforcement Learning (DTDRL), based on Deep Reinforcement Learning (DRL) to control both the RSU and UAV. DRL is an effective method to deal with dynamic environments [88].

5.1.3 Contributions

This chapter presents a scheme for serving vehicles through a cooperative system composed of an RSU and a UAV while populating the UAV cache by content delivered through the vehicles from the RSU. Our contributions are as follows:-

- We present and model a cooperative system where a deployed UAV aides an existing RSU to deliver contents for passing vehicles in a particular road segment. The UAV is used to offload some of the services provided by the RSU and hence enhance the overall network performance. To this end, an optimizer plans how to schedule the RSU and UAV radio resources besides planning the UAV trajectory and maintaining the UAV cache.

- We formulate the joint problem of UAV trajectory, RSU and UAV resource scheduling mathematically as an optimization problem to find the optimal solution that maximizes the concerned utility.

- Given the hardness of the problem and the existing uncertainties, we resort to a solution based on DRL. Thus, we formulate the problem as MDP taking into consideration two sets of

actions namely, UAV mobility and RSU service.

- Based on the action space, we develop DTDRL method where the two actions are handled. Our solution approach can handle simultaneously RSU resource scheduling and UAV mobility.
- In addition, we design three effective algorithms to assist the DTDRL agent in allocating the resources of RSU downlink and UAV uplink and downlink.
- We conduct several extensive simulation based experiments using SUMO to study the behaviors of our proposed system and compare our solution approach with other baseline methods.

5.1.4 Organization

In Section 5.2, related works are presented. Section 5.3 explains our system model in details followed by Section 5.4 which formulate the problem mathematically. In Section 5.5, we discusses our solution approach. Then, Section 5.6 presents our numerical results and finally, Section 5.7 sums up the chapter.

5.2 Related Work

A wide spectrum of studies has been devoted to content delivery in vehicular networks for multitude of infotainment services. Some works are based on stationary infrastructures while others investigate the roles of aerial base stations such as UAVs. However, there exists only limited number of papers which address the collaboration among various types of facilities. Here, as this study seeks a collaboration among stationary and non-stationary entities, this section will be more concentrated on the studies that deal with the interplay between UAVs and fixed base stations.

The authors of [89] present a framework to help planning joint UAV and RSU networks for urban areas. The proposed framework takes into account the limited budget, UAV battery capacity, wireless coverage and motivates renewable energy. However, this work does not involve a cooperative planning strategy to deliver contents. In [90], the authors propose a new scheme for cooperative RSUs in order to serve vehicles while considering content popularity for caching purposes. The

paper also considers service delay and content encoding technology. The authors in [41] propose a new scheme to jointly optimize content delivery and content caching in vehicular edge networks via deep deterministic policy gradient and formulate as double time-scale Markov decision process. In [91–93], the authors try to improve data dissemination in vehicular networks through considering different types of connections namely, vehicle to vehicle (V2V), vehicle to RSU (V2I), and vehicle to UAV (V2U). Indeed, these three papers focus more on reducing transmission latency rather than serving contents. Next, the authors of [94] provide a platform to enable UAV-assisted vehicular networks by cooperating with other transceivers, specifically, vehicles and RSUs. In spite of the several issues addressed by that work in vehicular networks which include dynamic topology, reliable connectivity and others, it does not elaborate on caching strategies to improve content delivery. Furthermore, the authors of [95] suggest software defined networks to support heterogeneous vehicular networks where RSU and BS are collaborating in multicasting contents to the users. The objective of this work is to improve the utility of each facility, RSU and BS, where each one partially contributes towards serving the demands of vehicles by offering incentives. In addition, the authors of [96] design a framework for cooperative RSU and high-altitude platforms (HAPs) where the latter broadcasts contents to users before the requests arise.

Other main limitations of the referenced works are summarized in the following two points. First, some of the works are only applicable for distributing small-size data that, in reality, does not significantly contribute to mitigate network bottlenecks. Second, the assumption of one data/file size may not represent a real scenario, besides, some of the above mentioned works provide a solution for only one static snapshot while the consecutive time slots are remain unaddressed. Unlike the works discussed above, the novelty of this current work concentrates around introducing a cooperative content service mechanism to improve the utility of RSU and UAV and provide efficient solution approach to control the different aspects of the system. Our work mainly focuses on leveraging the flexibility and mobility nature of the UAV besides providing a new method to relay contents to the UAV via passing vehicles.

5.3 System Model

We consider the service of content delivery in vehicular networks as shown in Fig. 5.1. A set of vehicles, $I = \{0, 1, 2, ..., i, ..., |I|\}$, travel through the considered network and stream contents that are stored at the network edge on caches co-located with the infrastructure, e.g., RSU[1]. Now, since vehicles have only limited residence within the range of the RSU, the content to be streamed may not be completely served by the RSU alone; hence, in this work we exploit cooperative content delivery to offload an overwhelmed RSU with large data demands where a UAV is also leveraged to assist the network in delivering contents, partially or fully to passing by vehicles. The UAV is assumed to be flying within the service area and passing by vehicles are exploited by the RSU to store, carry and forward contents to update the cache at the UAV. Here, we assume that the UAV and RSU can exchange control information over an out of band signaling channel. Accordingly, the UAV through this channel informs the RSU regarding its cache status and request contents to be delivered through passing vehicles. The optimizer can then plan the best trajectory of the UAV.

Now, upon their deployment, UAVs will generally have limited access to the network through a robust backhaul link, and accordingly, updating the cache of the UAV directly from the network may be difficult. To overcome this issue, one possible solution is to do proactive caching ahead of deployment [66]; however, since the popularity of the content may change with time, and that content may be generated and consumed spontaneously particularly with the emergence of popular social media applications, dynamically adapting the content of the cache becomes critical for improving the quality of experience of end users.

We advocate in this chapter the cooperation of vehicles with the RSU to establish a link through which the RSU can make the necessary updates on the cache of the UAV; namely, vehicles will act as content mules and the RSU opportunistically will forward through them the contents it needs to deliver to the UAV.

A time horizon of length N is considered and time is assumed to be slotted into smaller time slots $\{0, 1, 2, ..., n, ..., N\}$ where each time slot n is of length Δ sec. At each time slot n, there is a subset of vehicles $I^n \subseteq I$; some vehicles are within the range of the RSU and others could also

[1]For simplicity, we use one-way one-dimensional road segment, however, the same environment can be extended to multiple dimensions.

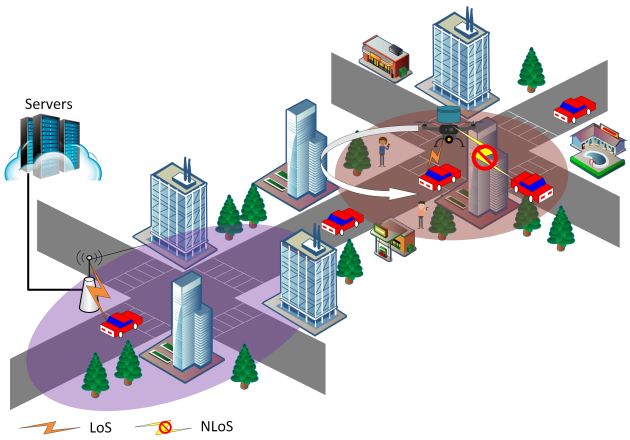


Figure 5.1: System Model

be within the coverage of the UAV. Vehicles are assumed to arrive at the road segment according to a particular arrival rate. The RSU makes decisions to serve requests initiated by each vehicle to download a content, and the RSU also makes decisions to offload content delivery to the UAV through these vehicles. We can summarize the proposed scenario as follows:

- Vehicles enter the coverage region of the RSU and each vehicle in a vehicular network broadcasts on the control channel announcement beacon messages, which contain information about the requested content and its route[2].
- The RSU schedules the transmissions to serve vehicles, taking into account the requested content size, available radio resources and vehicles' residence times.
- Given the short residence time of vehicles within the coverage of the RSU and the limited wireless resources of the RSU, the RSU may leverage a UAV to cooperate in delivering contents to the vehicles.
- It is assumed that vehicles leaving the coverage region of the RSU will pass through the coverage of the UAV.
- The UAV is assumed to be deployed with some cached contents; as explained, the RSU opportunistically selects some vehicles to transfer contents (and hence update) to the UAV

[2]For simplicity, we assume each vehicle requests one content.

by pushing the entire contents on the vehicles. The selected vehicles will *store* the contents in their local buffers and *carry* them until they become in contact with the UAV where they *forward* the buffered contents.

- To facilitate the communication between the UAV and passing by vehicles, the RSU manages the mobility of the UAV such that the UAV can fetch contents from vehicles to update its cache. Besides, the RSU tries to give sufficient contact time between the UAV and the vehicles to receive full service.

- Once the UAV cache is loaded with contents, it is ready to participate in collaborating with the RSU in delivering contents to the vehicles. Thus, the RSU will take into consideration the ability of the UAV to serve the demands.

The RSU aims at satisfying the vehicles by serving the contents in whole before they leave the considered area [86].

5.3.1 Content Model

We assume a library containing a particular number of contents (J), each of size Z_j. The popularity of contents depicted by its access frequency which is assumed to be derived from a Zipf distribution with a skewness parameter α which characterizes the distribution.

5.3.2 Communication Model

The RSU and the UAV are both assumed to be operating on the cellular spectrum and they communicate with vehicles using cellular VANETs [97]. We assume to use Time-Division Multiple Access (TDMA) where each time slot is allocated at most to one vehicle. Both infrastructures, RSU and UAV, are assumed to be capable of transmitting simultaneously. That said, each one can only communicate with vehicles within its coverage area. Typically, the communication channel between RSUs and end users is usually modeled as large- and small-scale fading. However, in the context of RSU where the RSU is situated such that there exists a clear link between transmitter and receiver [98]. Thus, the communication link can be characterized by strong line-of-sight. Consequently, the

channel gain between the RSU and vehicle i at time slot n can be written as follows:

$$h^n_{R\to i} = h_0\left(\sqrt{(L^n_i - w^n)^2}\right)^{-\tau}, \forall i \in I^n, \forall n \tag{5.1}$$

where h_0 is the mean path gain at reference distance = 1m and τ denotes path loss exponent. Also, w^n is the UAV location at time slot n and L^n_i denotes vehicle i location at time slot n.

Then, the instantaneous throughput of the RSU downlink to vehicle i can be formulated as follows:-

$$T^n_{R\to i} = W\log(1+\frac{h^n_{R\to i}P_R}{\sigma^2}), \forall i \in I^n, n, \tag{5.2}$$

where W is the available bandwidth, P_R denotes the transmission power of the RSU, and σ is the thermal noise power.

For the UAV, we use channel model for the UAVs in urban area where high-rise buildings and other objects appear which could disturb the links between the UAV and receiving vehicles. Thus, we assume that the link propagation is characterized by both Line-of-Sight (LoS) and non Line-of-Sight (NLoS). Here, $S^n_{U\to i} \in \{\text{LoS, NLoS}\}$ indicates the state of the channel between the UAV and vehicle i at time slot n. The probability of having LoS link adopted in this chapter is similar to [99]. Then, we can find the probability of channel states between the UAV and vehicle i.

$$Pr(S^n_{U\to i} = \text{LoS}) = \frac{1}{1+\eta_1 e^{(-\eta_2(\theta^n_{U\to i}-\eta_1))}}, \forall i \in I^n, n, \tag{5.3}$$

where η_1 and η_2 are constant parameters of the environment. $\theta^n_{U\to i} = \frac{180}{\pi}\arctan(\frac{z_U}{d^n_{U\to i}})$ is the angle degree between vehicle i and the UAV at n. Meanwhile, z_U denotes the height of the UAV antenna and $d^n_{U\to i}$ is the horizontal distance between vehicle i and the UAV at time slot n. Moreover, the probability of having NLoS can be found from $Pr(S^n_{U\to i} = \text{NLoS}) = 1 - Pr(S^{i,n}_U = \text{LoS})$. Next, the channel power gain for each vehicle $i \in I^n$ at time slot n is computed as:

$$h^n_{U\to i} = \begin{cases} (D^n_{U\to i})^{-\beta_1} & S^n_{U\to i} = \text{LoS}, \\ \beta_2(D^n_{U\to i})^{-\beta_1} & \text{otherwise}, \end{cases} \tag{5.4}$$

where $D^n_{U\to i}$ is the Euclidean distance between the UAV and vehicle i at time slot n; $D^n_{U\to i} = \sqrt{(d^n_{U\to i})^2 + z_U^2}$. β_1 denotes the path loss exponent and β_2 is the attenuation factor for NLoS. Thus $h^n_{U\to i}$ can also be rewritten as:

$$h^n_{U\to i} = p(S^n_{U\to i} = \text{LoS})(D^n_{U\to i})^{-\beta_1} + (1 - Pr(S^n_{U\to i} = \text{LoS}))\beta_2(D^n_{U\to i})^{-\beta_1} \quad (5.5)$$

Then, the instantaneous throughput of the UAV downlink to vehicle i can be formulated as follows:-

$$T^n_{U\to i} = W\log(1 + \frac{h^n_{U\to i}P_U}{\sigma^2}), \forall i \in I^n, n, \quad (5.6)$$

where P_U is the UAV transmission power. Likewise, we can find the uplink throughput between vehicle i and the UAV at time slot n.

$$T^n_{i\to U} = W\log(1 + \frac{h^n_{U\to i}P_V}{\sigma^2}), \forall i \in I^n, n, \quad (5.7)$$

where P_V is the vehicle i transmission power. To maintain a free interference communication (i.e., the interference below a certain threshold), a minimum distance between the RSU and the UAV is considered. Therefore, interference-free communication is considered in this work. In other words, the trajectory of the UAV maintains a distance away from the RSU to keep the interference below the threshold.

Also, in the case when they operate on the same spectrum, we assume that the UAV and RSU are equipped with directional antennas where the coverage range of each one does not overlap with that of the other or they communication quality should be greater than a certain threshold [100]. Next, we constrain the mobility of the UAV such that it does not get closer to the RSU. Also, as this work focuses on providing service to a certain road segment, the UAV is assumed to move only within the boundaries of that area.

Finally, the UAV position is determined by its trajectory where the UAV can move back, forth, or hover in its place at each time slot. Let $(w^{n+1} - w^n)$, the distance that UAV moves during one time slot, denote the UAV speed and direction at time slot n. For example, if $(w^{n+1} - w^n) > 0$, the UAV has moved forth, else, the UAV has moved back. The maximum UAV speed, V_{max}, is predetermined

and should not be violated.

5.4 Mathematical Formulation

This section formulates the system model described above mathematically. Let us define $x_{i,j}^n$ as a scheduling variable for the wireless resources of the RSU, where:

$$x_{i,j}^n = \begin{cases} 1 & \text{vehicle } i \text{ is served content } j \text{ by the RSU at time slot } n, \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

Likewise, we define y_i^n for the UAV wireless resources:

$$y_i^n = \begin{cases} 1 & \text{vehicle } i \text{ is served by the UAV at time slot } n, \\ 0 & \text{otherwise.} \end{cases} \tag{5.9}$$

Then, we define $v_{i,j}^n$ for the uplink resources from vehicle i to the UAV for content j:

$$v_{i,j}^n = \begin{cases} 1 & \text{vehicle } i \text{ is sending content } j \text{ to the UAV at time slot } n, \\ 0 & \text{otherwise.} \end{cases} \tag{5.10}$$

Now, we compute the total amount of content j served to vehicle i through the RSU at each time slot n ($U_{R\to i,j}^n$).

$$U_{R\to i,j}^n = x_{i,j}^n \Delta T_{R\to i}^n, \forall i \in I^n, n \in N, j \in J, \tag{5.11}$$

where Δ is the length of one time slot and J is the library of contents. Similarly, we compute $U_{i\to U,j}^n$ which denotes the total amount to upload content j to the UAV through vehicle j during time slot n.

$$U_{i\to U,j}^n = v_{i,j}^n \Delta T_{i\to U}^n, , \forall i \in I^n, n \in N, j \in J. \tag{5.12}$$

Then, we compute the amount served to vehicle i through the UAV during time slot n.

$$U^n_{U\to i} = y^n_i \Delta T^n_{U\to i}, \forall i \in I^n, n \in N. \tag{5.13}$$

Next, vehicle i is considered served if and only if it downloads its desired content fully from the RSU, UAV, or both. Here s_i indicates whether vehicle i is served or not as follows:

$$s_i = \begin{cases} 1 & \sum^J_{j=0} C_{i,j} Z_j \leq \sum^N_{n=0} (\sum^J_{j=0} (U^n_{R\to i,j} C_{i,j}) + U^n_{U\to i}), \forall i \in I, \\ 0 & \text{otherwise.} \end{cases} \tag{5.14}$$

5.4.1 Backhauling via Vehicles

Here, we suppose that the UAV does not have the requested contents. Therein, we assume that the UAV has a certain cache capabilities that can be filled with contents until fullness.

Let us define $\zeta^n_{i,j} \in \{0,1\}$ as an indicator holding value 1 if content j on vehicle i is fully fetched by the UAV before or at time slot n and 0 otherwise.

$$\zeta^n_{i,j} = \begin{cases} 1 & Z_j \leq \sum^n_{n'=0} U^{n'}_{i\to U,j}, \forall i \in I^n, n, j \in J, \\ 0 & \text{otherwise.} \end{cases} \tag{5.15}$$

Next, we introduce $m^n_j \in \{0,1\}$ as a binary variable where it is equal to 1 if content j is available on the UAV at time slot n and 0 otherwise. The value of m^n_j has three cases; it is either equal to 1 when the content is just fetched or 0 if it is removed in the previous time slot. The third case is that it remains the same if no change occurs on it.

$$m^n_j = \begin{cases} 1 & \sum^I_{i=0} v^n_{i,j} \zeta^n_{i,j} = 1, \forall n, j \in J, \\ 0 & f^{n-1}_j = 1, \\ m^{n-1}_j & \text{otherwise,} \end{cases} \tag{5.16}$$

where $f^n_j \in \{0,1\}$ is a decision variable that equals to 1 if content j is decided to remove at the end of time slot n and 0 otherwise. Next, we introduce $\hat{s}_{i,j} \in \{0,1\}$ which holds value of 1 if vehicle i

has downloaded the entire content j from the RSU only.

$$\hat{s}_{i,j} = \begin{cases} 1 & Z_j \leq \sum_{n=0}^{N} U^n_{R \to i,j}, \forall i \in I, j \in J, \\ 0 & \text{otherwise}, \end{cases} \tag{5.17}$$

Now, we can write the optimization problem as follows:

$$\max_{x,y,v,f,w} \quad \sum_{i=0}^{I} s_i \sum_{j=0}^{J} C_{i,j} Z_j \tag{5.18a}$$

$$\text{s.t.} \quad \sum_{n=0}^{N} y^n_i \leq 1, i \in I^n, \tag{5.18b}$$

$$\sum_{n=0}^{N} \sum_{j=0}^{J} x^n_{i,j} \leq 1, i \in I^n, \tag{5.18c}$$

$$\sum_{n=0}^{N} \sum_{j=0}^{J} v^n_{i,j} \leq 1, i \in I^n, \tag{5.18d}$$

$$v^n_{i,j} \leq \hat{s}_{i,j}, \forall n, i \in I^n, j \in J, \tag{5.18e}$$

$$y^n_i \leq \sum_{j=0}^{J} m^n_j C_{i,j}, \forall n, i \in I^n, \tag{5.18f}$$

$$\sum_{j=0}^{J} m^n_j Z_j \leq \eta, \forall n, \tag{5.18g}$$

$$f^n_j \leq m^n_j, \forall j \in J, n. \tag{5.18h}$$

$$|w^{n+1} - w^n| \leq V_{max}, \forall j \in J, n. \tag{5.18i}$$

$$|w^n - \varpi| \geq \Lambda, \forall n. \tag{5.18j}$$

Since we assume TDMA as transmission access then only one transmission is allowed at a time as in Constraint (5.18b) and (5.18c). Constraint (5.18e) limits vehicle i upload transmission for the UAV to the contents that has been downloaded fully from the RSU. Constraint (5.18f) is added to ensure vehicle i is not being served through the UAV if the latter does not possess the requested content. Constraint (5.18g) prevents violating the limited cache capacity of the UAV where η denotes UAV cache size. Also, constraint (5.18h) states that a content cannot be removed from the UAV cache if it does not exist in the cache beforehand. Constraint (5.18i) prevents violating UAV max speed. And

finally, Constraint (5.18j) restrains UAV mobility to not cross a certain point such that no overlap occurs between the coverage ranges of the RSU and UAV where ϖ and Λ denote RSU position and the minimum distance between the two entities, the RSU and UAV, respectively.

Now, if we look at the objective function with its constraints, we can see that the presented problem contains several binary variables alongside a real-value decision variable w^n. In addition, our objective function is non-convex owing to the UAV trajectory. Hence, our problem is MINLP which is known to be difficult to solve. Furthermore, there are two unknown parameters; $C_{i,j}$ and L_i^n. These two parameters belong to the vehicles and in such context, its is impractical to assume the upcoming vehicles and their demands are given beforehand. Despite that there are some heuristic methods to solve such types of problems, they are still inefficient as they cannot considers all the possible scenarios [101]. Therefore, we suggest to use DRL in order to learn the environment aspects and solve our presented problem. The complete implementation of our solution approach is explained in the next section.

5.5 Solution Approach

To solve the given problem, we design DTDRL (based on DRL) besides cooperative and full-service content delivery algorithms. The problem is formulated as MDP (similar to Section 3.4.1) where the state, action, reward, are defined as follows:

- State S: The state at time slot n, $s^n \in S$, is defined as:

$$s^n = [w^n, L^n, R^n, C^n, D^n, U^n, K^n], \tag{5.19}$$

  where w^n denotes the current position of the UAV. L^n is a vector that contains the position of each vehicle at time slot n. R^n is a vector that contains the content requested by each vehicle at time slot n. D^n is a vector indicates how much each vehicle has downloaded at time slot n. U^n is a vector indicates how much each vehicle has uploaded to the UAV at time slot n. K^n is a vector the contains the indices of cached contents on the UAV at time slot n.

- Action A: The action taken at time slot n, $a^n \in A$, is composed of two sub-actions; a_1^n and a_2^n.

The first sub-action, $a_1^n \in \{0,1\}$, is for the RSU and it specifies whether to continue or cut the service for the vehicle chosen by Algorithm 10. We design Algorithm 10 to reduce the action space where this algorithm selects a vehicle to serve each time the RSU has no vehicle scheduled to serve[3]. However, it is still the agent responsibility to decide whether to continue or stop the service for the scheduled vehicle through a_1^n. The second sub-action, a_2^n, is an integer number to select one velocity out of a list of different velocities for the UAV. Hence, The action space is of size $2Y$ where Y denotes the total discrete UAV velocities including hovering.

- Reward: The immediate reward, r^n, is the sum of positive rewards due to serving vehicles sufficiently. Here, the decision maker/agent selects vehicle i which lies in the coverage region of the RSU to be served. Vehicle i must still be in need for content at time slot n. Similarly, the solution approach also chooses vehicle within UAV coverage to transmit or receive (if the UAV has the required content cached). For the sake of simplicity, we assume that each vehicle can only download one content. Thus, the immediate reward can be written as:

$$r^n = \begin{cases} \sum_{j=0}^{J} C_{i,j} Z_j & \sum_{i=0}^{I} \sum_{j=0}^{J} C_{i,j} Z_j \leq \sum_{n'}^{n} (\sum_{j=0}^{J} (U_{R \to i,j}^{n'} C_{i,j}) + U_{U \to i}^{n'}), \\ 0 & \text{otherwise.} \end{cases} \tag{5.20}$$

Intuitively, we may design random or greedy policy. For example, the agent probably selects vehicles to serve at random or based on the strength of the communication channel between the RSU and vehicles to be served. However, such policies may lack substantial benefits owing to the complexity of the proposed system. Specifically, concerning the cooperation between the RSU and UAV is overlooked in such basic policies. Thereby, deep reinforcement learning can better explore and build knowledge about the environment and then exploit based on the enhanced policy it has developed throughout the learning phase.

As illustrated in algorithm 9, DTDRL interacts with the environment to collect the samples through several iterations and reveal the actual rewards. First, DTDRL initializes random sampling

[3]This algorithm approximates the solution and without it, the agent will need to learn for longer time. The algorithm also helps the agent to provide full content service in a prompt way rather than randomly schedule vehicles.

policy and value function for the neural networks. Then, the agent starts to interact with the environment through several iterations. In order to achieve dual task, the agent splits the the action a^n into two sub actions, a_1^n and a_2^n, using division with floor operations. By dividing the action a^n to the Y, the agent can work out the first action which corresponds to the RSU wireless scheduling. Next, the agent take the division reminder of a^n to Y which gives the second action which corresponds to the UAV velocity. The RSU decides whether to serve the selected vehicle ζ or not based on a_1^n. The vehicle ζ is chosen by Algorithm 10. This algorithm selects a vehicle which recently entered the area and thus has the longest time to remain in contact with the RSU. If the selected vehicle is fully served or the agent decides to cut service from it, by setting $a_1^n = 1$, another vehicle is selected based on the same policy. When the selected vehicle is fully served, the agent is rewarded based on that service. Likewise, the UAV adjusts its position based on a_2^n and then schedules the wireless resources based on Algorithm 11 and 12. If the selected vehicle to be served through the UAV has received full service, this is also considered as step reward.

Concerning the uplink, as laid out in Algorithm 11, the UAV checks each existing vehicle and selects the one which satisfies its criteria. The vehicle should hold a popular content which is not cached on the UAV before. The content popularity is measured based on most frequently used (MFU). If the cache has no space to cache a content, then it can replace contents based on least recently used, however, the fetched content has to be more popular. The complexity of Algorithm 11 is $O(\partial|J|)$ where ∂ represents the maximum number of vehicles present simultaneously in the road segment and $|J|$ is the number of contents in the library. The value of ∂ depends on the road density and, in reality, $\partial \ll I$.

For the downlink, the UAV is committed to serve a vehicle as long as it resides within the coverage area of the UAV and not fully served. The UAV also checks if it has already scheduled a vehicle to serve previously so it continues the service if that vehicle still satisfies the conditions

Algorithm 9: DTDRL

Inputs: L, C, N, Learning Rate, γ, ε.
Outputs: The dual policy of RSU resource allocation and UAV velocity control.
Initial policy π with random parameter θ and threshold ε
for *each episode* $k \in \{0,1,2,...\}$ **do**
    for $n : \{0,1,2,...,N\}$ **do**
        Observe state Z_j,w^n,L_i^n,$C_{i,j}$,U_i^n,D_i^n,$\forall i \in I^n$.
        Select action a^n from $\pi_{\theta_{old}}$
        Extract first action, $a_1^n = \lfloor \frac{a^n}{Y} \rfloor$
        Extract second action, $a_2^n = a^n \% Y$
        Serve vehicle that is selected by a_1^n via the RSU.
        if $a_1^n = 1$ **then**
            Set $\zeta = None$
        Call Algorithm 10
        if *the selected vehicle is fully served at n* **then**
            Add the reward to r_n based on (5.20).
        Move the UAV according to $w^{n+1} = w^n + a_2^n$.
        if *UAV is outside the road segment* **then**
            Keep the UAV at the current position.
        Set $v_{i,j}^n \in \{0,1\}$ based on Algorithm 11.
        Set $y_i^n \in \{0,1\}$ based on Algorithm 12.
        if *the selected vehicle is completely served* **then**
            Calculate step reward as in Eq (5.20).
        if *vehicle i content is fully received by the UAV* **then**
            if *vehicle i content needs space* **then**
                Remove other content(s) as in 5.5-C.
            Store vehicle i content in the UAV cache.
    Compute the advantage function.
    Optimize the clipped objective.
    $\theta_{old} \leftarrow \theta$.

Algorithm 10: RSU to Vehicle Service

Inputs: n, Z_j, $R_{i,j}$.
Outputs: ζ as the chosen vehicle to commit serving.
if $\zeta = None$ *OR Vehicle* ζ *has left the RSU coverage OR fully served* **then**
    Set $\zeta = None$
    for *each vehicle i within RSU coverage at time slot n* **do**
        if *vehicle i is not completely served and has the longest remaining time to stay within the RSU coverage* **then**
            Set $\zeta = i$

Algorithm 11: Popularity-Based Content Fetching (PBCF)

Inputs: n, Z_j, $C_{i,j}$.
Outputs: Π as the vehicle to fetch from at n.
Set $\Pi = None$.
Sort cache items by popularity
for $\forall i \in I$ *where* $a_i < n < d_i$ **do**
    if *i's content does not exist in the UAV cache and the content is not fully fetched yet* **then**
        Compute the total size of cached contents on the UAV.
        if *the vehicle i content cannot fit in the cache* **then**
            Define $R = 0$ as the amount to remove in order to cache the new content.
            for *each content* $j : k_j^n = 1$ **do**
                if *vehicle i content has higher MFU value than* Π *and* j **then**
                    Set $R = R + Z_j$
                    if *vehicle i content size* $\leq R$ **then**
                        Set $\Pi = i$
                        break
                else
                    break
        else
            if Π*' content is less popular than i's* **then**
                Set $\Pi = i$

above. However, if no vehicle has been selected previously or the service terminated, the UAV starts to look for another vehicle to serve. The UAV gives priority to vehicles which have already downloaded some of their requests. In such way, the UAV will enable the cooperation with the RSU in order to fulfill vehicles needs. The UAV also takes into consideration the availability of requested contents in its cache as it cannot serve vehicles that it does not hold along their requests in the first place. The complexity of Algorithm 12 is $O(\partial)$.

Algorithm 12: Committing Content Service (CCS)

```
Inputs: n, Z_j, R_{i,j}, G.
Outputs: G as the chosen vehicle to commit serving.
if G = None OR Vehicle G has left the UAV coverage OR fully served then
    Set G = None
    for each vehicle i within UAV coverage at time slot n do
        if vehicle i requested a content that is cached on the UAV, not completely served,
          has more content amount downloaded previously, and the remaining time slots are
          enough to receive the whole content then
            Set G = i
```

Finally, the cache replacement policy used in this work is as follows. Whenever the cache is full and a new content is fetched, the UAV starts freeing up space by removing contents which are less popular until enough room is made to store the recently fetched content.

5.6 Evaluation

5.6.1 Simulation Setup

For the Deep reinforcement learning, 3 linear layers are used with tanh as activation function for the middle layers and softmax for the output layer. Internal layers contain 64 units each and Adam optimizer is incorporated to minimize the loss function. Learning rate is set to 0.002, γ to 0.08, and clip to 0.02. On the other hand, we use SUMO to mimic the vehicular environment with a road of 1 Km. The key simulation parameters are listed in table 5.1.

Table 5.1: Simulation parameters in cooperative UAV-RSU framework

Parameter	Value
RSU and UAV coverage range	200 m
Δ	1 sec [77]
Number of Contents	20
Bandwidth	40 MHz [84]
Zipf (α)	1.5
$P^{i,n}_{U\rightarrow V}$, $P^{i,n}_{V\rightarrow U}$	10 W
UAV discrete velocities	[0, 10] m/s
Inter-arrival time of vehicles	[0.5-3] s
UAV altitude	20m [102]
η_1, η_2 (for dense urban area)	11.95, 0.136

To the best of knowledge, there is no work in the literature that targets similar problem. Thus, we use baseline methods similar to [90] to compare with our solution approach.

- NonCoop: Non-cooperative method where the RSU and UAV aim to maximize their own gain independently. That is, the RSU greedily selects vehicles to serve such that it can fully satisfy and generate revenue. Similarly, the UAV does the same and it is assumed stationary.
- No UAV: As its name suggests, there is no UAV while the RSU alone will try to maximize the revenue. This will let us understand how influential the UAV is on the overall system performance.
- Random: The RSU schedules resources to serve vehicles randomly.

5.6.2 Evaluation

We conduct five experiments to examine the various aspects of the proposed system.

DTDRL Convergence

as DTDRL encounters more observation samples and takes actions, it can learn to perform better as demonstrated in Fig. 5.2. One can observe that DTDRL converges after approximately

1000 iterations.

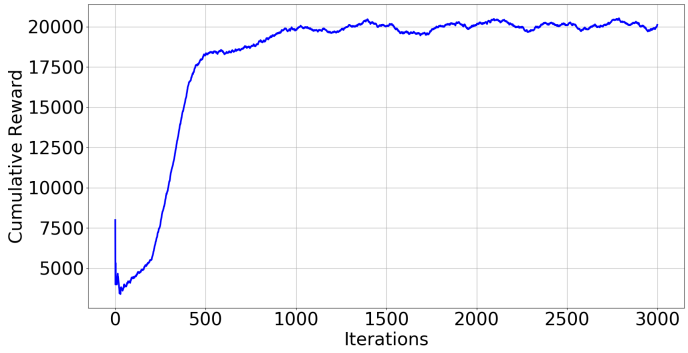


Figure 5.2: DTDRL convergence.

Vehicle Arrival Rate

In order to measure the impacts of high dense and low dense road conditions, we vary the arrival rate and compute the ratio of total vehicles served. In Fig. 5.3, we plot the results of various arrival rates ranging from 0.25 to 0.5. In this experiment, we set content size between 1.8 to 2.2Gb and cache size to 18Gb. As it can be observed, the percentage of vehicles served goes down as the arrival rate increases. In fact, as the density of the road increases, the requests will increase and the RSU and UAV will be less able to serve all the demand. One can also notice that our solution approach has significant advantage over the baselines. This advantage becomes more apparent as the arrival rate increases. That is, our solution approach can better adapt to a highly dense environment by enabling efficient cooperation between the UAV and the RSU. Besides, the mobility of UAV in DTDRL has a great impact on the system performance.

At arrival rate of 0.25 veh/sec, our proposed DTDRL attains 10% more vehicles served than the second highest which is NonCoop. This performance gap grows up to reach above 15% at arrival rate of 0.5 veh/sec. On the other hand, there is a considerable difference between NonCoop and No UAV. This difference is due to the existence of the UAV; i.e., the UAV can increase the performance by around 15%. In the meantime, Random seems to provide poor solutions as it cannot serve most of the vehicles due to the interrupted service provided.

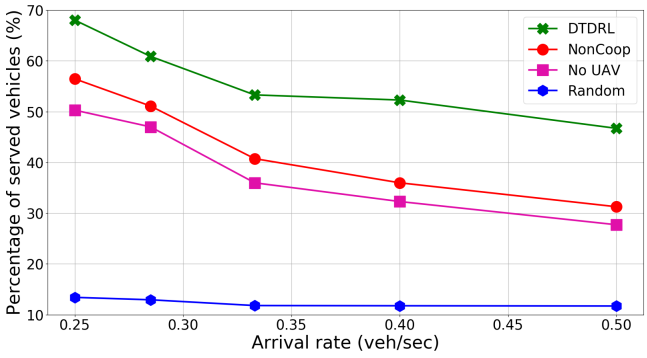


Figure 5.3: Service rates of vehicles for different vehicle arrival rates

Content Sizes

One of the key factors that impacts the performance of the system is the size of contents. When the content sizes are small, the RSU and UAV can readily serve them completely to the requesters. The cooperation becomes more necessary as the content size increases where the RSU needs assistance from the UAV to pursue the service until completion. In light of above, we conduct this study where we vary the content size from 1.4 to 2.2Gb. Note that arrival rate of vehicles is set to 0.33 veh/sec and cache capacity can store at most 10 contents.

As shown in Fig. 5.4, with small contents, DTDRL and NonCoop almost achieve similar performance since the cooperation and UAV mobility is less important. However, the superiority of DTDRL appears as the content size increases. We can see that at content size of 2.2 Gb, the difference becomes around 15%. Meanwhile, No UAV obtains less gain by around 50% to 40% vehicles served. Random, as usual, comes in the last place and the performance sharply decreases with content size.

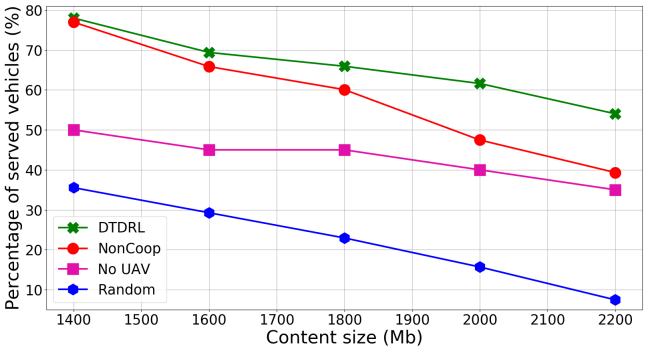


Figure 5.4: Service rates of vehicles for various content sizes.

Cache Capacity

the cache unit capacity plays key role in the proposed system as it impacts the capability of the UAV to contribute in serving vehicles. A UAV with small cache unit can store only few contents and consequently serve less number of vehicles. While a UAV with larger cache unit would more effectively cooperate with the RSU in delivering contents. In this study, we vary the cache unit size in the range of 4 to 20 Gb while content size is set to 2Gb and arrival rate of vehicles is 0.33 veh/sec. As illustrated in Fig. 5.5, larger cache unit results in higher overall performance. At 20Gb, it attains 15% gain higher than NonCoop. Meanwhile, No UAV is not impacted by the cache capacity changes since it does not include a UAV in the first place. One can also notice that even Random experiences better performance with bigger cache units.

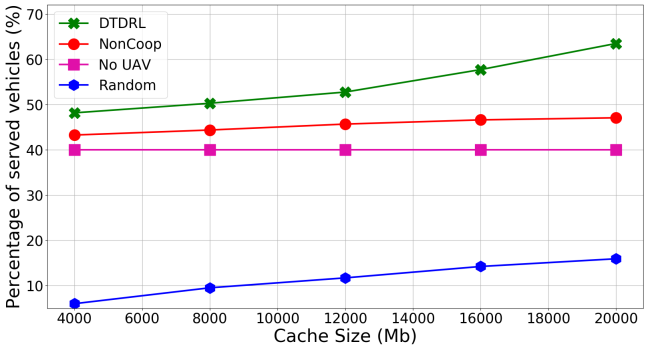


Figure 5.5: Service rates of vehicles for different cache capacities.

Content Popularity Distribution

the popularity distribution characteristics is, in general, very critical in cache based systems as it determines the average number of hits each content will receive and the shape of hits distribution. Thus, in this study we are going to vary the skewness parameter, α, of Zipf and observe the behaviors of the system. Hence, we change α between 1.1 to 1.7. At 1.1 the hits will be more fairly distributed among contents while in 1.7 the hits will be concentrated on the top popular contents.

As seen in Fig. 5.6, when α is as small as 1.1, the service provided is relatively limited. This is due to that the UAV can less contribute in serving vehicles as more requests are not cached. In this experiment we set the content size to 2 Gb and cache unit to 10Gb while arrival rate is 0.33

veh/sec. However, as α increases, one can observe that more vehicles are being served by DTDRL. As α goes higher, popular contents will receive higher requests as well. Thus, if those contents are cached, the UAV will be able to serve them. We can notice that our proposed method makes most benefit of this feature while the other methods achieve very limited gain. That is, our method can better take advantage of the cooperation and the existence of the UAV and offload more traffic volume from the RSU.

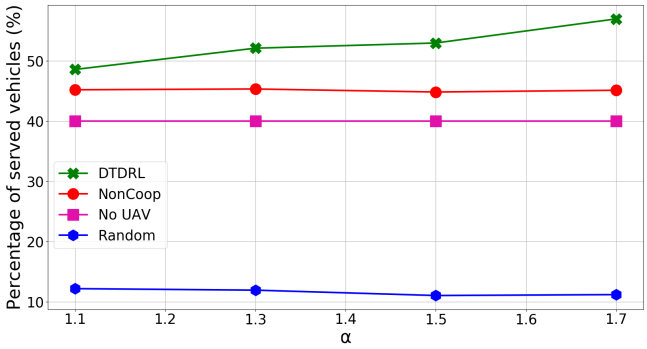


Figure 5.6: Service rates of vehicles for Zipf skewness values.

UAV mobility

needless to say, UAV trajectory has profound influences on the overall performance. Thus, in order to study the performance of our solution approach in terms of steering the UAV, we conduct a study where we compare with three different well-known counterparts.

- SUAV: Stationary UAV similar to NonCoop.
- RW: Random walking UAV.
- CW: Constant walking UAV where the UAV moves at the maximum speed. Once the UAV reaches either end of its designated coverage segment, it reverses the direction.

In this experiment, we set cache size to 10Gb, content size to 2Gb, and vehicle arrival rate to 0.3 veh/sec. Additionally, we perform two tests, one with content updates through vehicles and the second one without updates. In the latter, we assume there are random contents stored in the UAV cache and will not be updated. Through this, we can evaluate the impacts of the proposed fetching methods.

In Fig. 5.7, we present the contributions of the UAV in serving vehicles. This is computed as the total gain achieved through the UAV over the total gain. As demonstrated in the bar chart of Fig. 5.7, our proposed method generates much higher gain from the UAV than using a UAV with static cache. In general, it is very noticeable that DTDRL contributes higher than the other methods at 35% of total vehicle service and about 42% better than SUAV. SUAV, in its turn, comes in the second place with slightly better performance than RW. Meaning, that a stationary UAV can better establish stable connection with vehicles than other methods that do not take smart decisions. We can also observe that RW has slightly better performance than CW due to that we use uniform random walking where the UAV circulates around the center of its coverage. That will allow the UAV to establish opportunistically longer contact times with some vehicles. In contrary CW keeps the UAV moving far from the center of the road and thus the connection will experience several interruption events.

Moreover, one can also observe that if the UAV does not update its cache through fetching contents, there will be significant loss. Our DTDRL demonstrates the highest utility of the fetching with more than 40% increase in the UAV contribution when fetching contents from the passing vehicles. Meanwhile, the other UAV mobility methods show only limited enhancement with content updates.

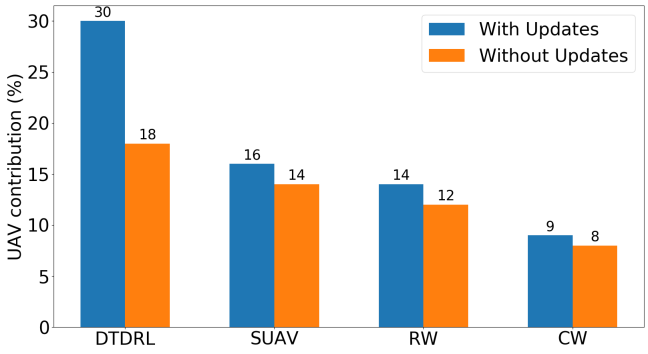


Figure 5.7: Comparison of UAV shares in serving vehicles for different trajectory techniques.

5.7 Summary

In this chapter, we presented a cooperative framework where a RSU and a UAV, equipped with a cache capability, cooperate to support content delivery service in vehicular networks. DTDRL algorithm is exploited to learn the vehicular environment and its dynamics in order to control the scheduling and caching mechanism. Simulation results show that the cooperation between the two infrastructures is very lucrative in terms of number of requests satisfied. The numerical results also demonstrate that the mobility of the UAV plays a key role in the overall performance. In addition, this work presents a new technique to populate contents on a flying UAV cache via passing vehicles and show how useful it is to leverage such technique. Finally, we demonstrate that the DTDRL approach achieves the highest performance compared to the other counterpart methods.

Chapter 6

Intelligent Surfaces to Enable Vehicular Communications in Dark Zones

6.1 Introduction

In the first four contributions discussed in Chapters 2, 3, 4, and 5, various scenarios have been studied. However, none of these works has addressed the issue of non-line-of-sight (NLoS) that appears in urban environments. Technically speaking, in many areas where large objects, *i.e.*, high-rise buildings or trucks, appear, it is very probable that wireless links between terrestrial infrastructures and vehicles face frequent disturbances. Hence, the service quality falls below the desirable levels and sometimes for extended periods of time. Moreover, certain regions, where obstacles severely block Line of Sight (LoS), are permanently out of coverage which, here, are dubbed as dark zones. Expanding wireless coverage to unserved areas translates to dramatic raise in costs. Meanwhile, recently, reconfigurable intelligent surfaces (RIS) have been recognized as a key promising technology for achieving cost- and energy-efficient communications via smartly reshaping the wireless propagation environment [103]. RIS is composed of a number of passive low-cost elements, each of which has the ability to independently tune the phase-shift of the incident radio waves. By adequately configuring the phase-shifts with the assistance of the RIS controller, the reflected signals can be constructively added [104]. Thus, the received signal strength can be improved at the point of interest.

Consequently, by leveraging RIS, an indirect LoS wireless communication link can be provided for vehicles travelling in a dark zone; *i.e.* a road where large buildings block the LoS of the Road Side Unit (RSU) [105]. It is assumed that those vehicles are requesting service and the RSU is interested in maximizing the quality-of-service (QoS) for the passing by vehicles. To this end, the RSU operator may need to jointly optimize the RSU resource scheduling and RIS element coefficients (passive beamforming) such that the minimum average bit rates of vehicles is maximized. In addition, despite that a few works have addressed RIS phase-shift configuration in vehicular networks, only non-practical phase-shift RIS case is considered where RIS elements can have continuous element tuning. However, due to limited hardware, phase-shift elements of RIS can only have limited number of values [106, 107].

Leveraging RIS in highly dynamic environments similar to vehicular communications implies a multitude of challenges. First, vehicles constantly change their position, hence, the distance between the RIS and vehicles is varying over time and that would highly affect the channel quality between them. Second, the RSU has limited resources in terms of the number of available wireless channels. Thereby, the RSU needs to optimize the radio scheduling while considering the mobility of vehicles which makes the problem more challenging especially when accounting for multi-user scenarios [108]. Third, vehicles move at different and varying speeds, that is, vehicles have various residence times. Considering the same service amounts for all the vehicles passing by the dark zones will deteriorate the performance of low speed vehicles. Subsequently, maximizing the minimum average bit rates provided to navigating vehicles regardless of their sojourn times should be considered. Fourth, the arrival times and speed of upcoming vehicles are not available, in practice, upfront to the RSU operator which makes the problem further more intricate. Finally and most importantly, discrete RIS phase-shift matrix configuration is a well-known problem which is generally hard to be solved especially in a context where the phase-shift matrix of the RIS together with wireless scheduling are jointly optimized.

To the best of our knowledge, this work is the first to consider practical/discrete RIS in vehicular networks where the mobility of vehicles together with the environment uncertainties are addressed. To this end, to tackle the aforementioned challenges, an intelligent solution approach is proposed, namely Deep Reinforcement Learning, along with effective optimization technique based on block

coordinate descent (BCD). The contributions of this work can be summarized as follows:

- A system model is presented that leverages discrete phase-shift RIS technology to extend and enhance RSU communication. Precisely, a RSU provides service for vehicles passing through a blocked zone indirectly by employing a RIS where the mobility of vehicles and future arrivals are considered.

- We investigate the joint vehicle scheduling and passive beamforming in RIS-empowered vehicular communication. This framework is formulated as an optimization problem with the goal of maximizing the minimum achievable bit rate for the vehicles passing through the dark zone. However, the formulated problem ends up to mixed integer non-convex problem, which is known to be difficult to solve.

- In order to tackle this challenge, we decouple the formulated problem into two sub-problems; wireless scheduling sub-problem and phase-shift matrix optimization sub-problem. Then, we resort to solve the first sub-problem via Deep Reinforcement Learning (DRL). To do so, the Markov Decision Process (MDP) is defined to be solved via DRL algorithm. Further, we propose BCD to solve the second sub-problem. We also demonstrate the robustness of our BCD algorithm. And, the computational complexity of the proposed algorithms are analyzed.

- Two case studies are carried out. The first one is to investigate how recent vehicular technologies can enable RIS integration with vehicular communications through obtaining precise vehicle positioning. Also, another study explores the area of RIS placement to optimize the overall network performance.

- Several extensive simulation based experiments are conducted using Simulation of Urban MObility (SUMO) to validate the effectiveness of our solution method and to compare with counterpart methods.

Notations: Vectors are denoted by bold-face italic letters. $\mathrm{diag}(x)$ denotes a diagonal matrix whose diagonal element is the corresponding element in x. $\mathbb{C}^{M\times N}$ denotes a complex matrix of $M \times N$. For any matrix M, M^H and M^T denote its conjugate transpose and transpose, respectively. $Pr(A \mid B)$ denotes the probability of event A given event B.

6.2 Related Work

Lately, high research efforts have been devoted towards investigating the introduction of the RIS to vehicular networks. In [109], the authors studied resource allocation of RIS-aided vehicular communications where they aim to maximize vehicle to base station link quality while guaranteeing vehicle to vehicle communications. The authors of [110] provided analysis for outage probability in RIS-enabled vehicular networks. This paper derives an expression of outage probability showing that RIS can reduce the outage probability for vehicles in its vicinity. The analysis also proves that higher density roads increase outage probability since passing vehicles can block the communication links. In [111], the authors proposed RIS-aided vehicular networks while considering two scenarios to estimate the channels. The first one is by assuming fixed channel estimation within a coherence time. While the second one neglects the small scale fading based on the fact that vehicular positions can be realised in advance. [112] considered constraint discrete phase-shift RIS with two challenges; channel estimation and passive beamforming.

Another body of works on RIS deals with practical considerations of discrete RIS elements. In [107], the authors introduced a finite number of phase-shift elements of RIS where the power is minimized while maintaining certain signal-to-interference-plus-noise ratio threshold. [113] proved how discrete phase-shift RIS is able to achieve high performance with minimum required number of phase quantization levels. This work shows that 3 levels are enough to attain the full diversity order. In addition, the authors of [114] also worked on practical RIS where multiple users are served in parallel. The objective of this work is to maximize the sum rate where the continuous digital beamforming and discrete RIS beamforming are done. [115] proposed RIS to assist multiple-input multiple-output (MIMO) systems with 2-bit phase-shift elements. In [116], the authors proposed utilizing RIS in cognitive radio systems yielding improved spectral efficiency and energy efficiency. [117] maximized the achievable sum rate of multi-users while the RIS sends information via controlling the reflecting modulation. [118] proposed a new location-based RIS where users' locations are not perfectly known. Hence, the angle between the users and RIS are estimated to configure the beams of the RIS and trasmitter. In addition, some other works also leverage RIS for security purposes, for example [119] suggested that RIS can help in alleviating security breaches

related to eavesdropping. In [120], the authors studied the security issues related to eavesdropping attacks under different circumstances including active and passive relays (RIS).

As opposed to the previous papers, this work accounts for vehicles mobility where vehicles constantly change their position with time. Additionally, as time progresses, new vehicles arrive to the concerned area while others depart. This process of birth-and-death vehicles brings many uncertainties to the context which are hard to cope with. Thus, we aim to find a solution approach that can handle the dynamic nature of this context besides anticipating the upcoming arrivals and other hidden information about the environment. Accounting for these two objectives will help the RSU-RIS to better decide when and how to serve vehicles during their residence time. Moreover, unlike many existing works in the literature, we propose to use practical/discrete RIS.

6.3 System Model

We consider a particular road segment with no direct connectivity via a RSU as depicted in Fig. 6.1. The line of sight (LoS) is assumed to be blocked by an obstacle, *i.e.,* a high building [121]. We also consider a predefined time horizon of length N which encompasses several smaller time slots, $[0,1,..,n,...,N]$. Meanwhile, we assume a flow of vehicles indexed by v is navigating and requesting communication services from the RSU located at (x_R, y_R, z_R) where x_R, y_r are the Cartesian coordinates and z_R is height of the infrastructure. The vehicles are moving at different and varying speeds, therefore, at each time slot $n \in N$, vehicle v location is denoted by (x_v^n, y_v^n, z_v). In order to provide uninterrupted service, the network operator leverages an RIS equipped with M elements, which is situated on a building and possesses a strong LoS with both the moving vehicles passing by the dark zone and the RSU. Here, we denote the RIS location by (x_I, y_I, z_I). The RSU operator aims to satisfy the vehicles by providing favourable quality of service.

The RSU is assumed to have a number of channels C to be scheduled for the vehicles [122][1]. In case when there are several vehicles present, the RSU has to determine how to schedule its resources and tune the RIS elements. Further, due to the mobility nature of the vehicular environment the distances between the RIS and vehicles change as time progresses. Meaning, the network operator

[1]For simplicity, we assume each vehicle can only be served via one channel.

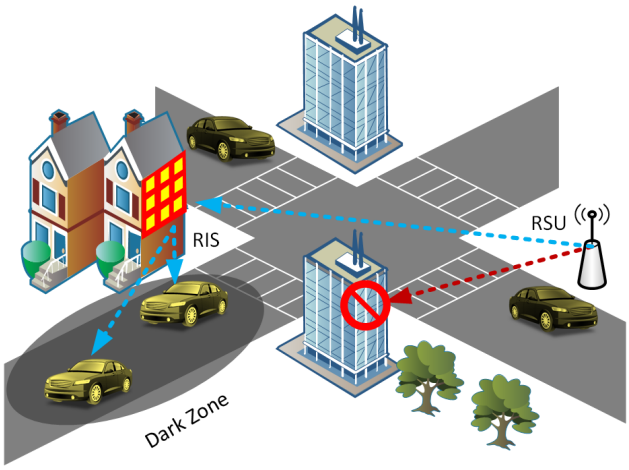


Figure 6.1: System Model

has to take into consideration that link quality degrades as vehicles moving far from the RIS.

Unlike prior work which deals with continuous phase-shift RIS, we assume a realistic scenario of a RIS where phase-shift coefficients are discrete. This scenario is more practical since it accounts for the real-world hardware limitations. However, discrete RIS is more challenging due to the additional constraint of discrete phase-shift. Moreover, the RIS consists of M elements, $[1,...,m,...,M]$, each of which is controlled via b bits. Hence, each one can be tuned to one of 2^b different angles.

6.3.1 Communication Model

In the proposed model, we consider a uniform linear array (ULA) RIS [121]. In addition, similar to the RSU, the RIS is assumed to have a certain height, z_I. The communication links between RSU and RIS and that between RIS and vehicle v are assumed to have a dominant line-of-sight (LoS). Thus, these communication links experience small-scale fading which are modeled as Rician fading with pure LoS components [123, 124]. Consequently, the channel gain between the RSU and RIS, $h_{I,R} \in \mathbb{C}^{M\times 1}$, can be formulated as follows.

$$h_{I,R} = \underbrace{\sqrt{\rho (d_{I,R})^{-\alpha}}}_{\text{path loss}} \underbrace{\sqrt{\frac{K}{1+K}} \bar{h}_{I,R}^{\text{LoS}}}_{\text{Rician fading}}, \tag{6.1}$$

where ρ is the average path loss power gain at reference distance $d_0 = 1$m. Also, K is the Rician factor and $\bar{h}_{I,R}^{\text{LoS}}$ is the deterministic LoS component which can be defined as follows

$$\bar{h}_{I,R}^{\text{LoS}} = \underbrace{\left[1, e^{-j\frac{2\pi}{\lambda}d\phi_{I,R}}, ..., e^{-j\frac{2\pi}{\lambda}(M-1)d\phi_{I,R}}\right]^T}_{\text{array response}}, \forall n \in N, \tag{6.2}$$

where $d_{I,R}$ is the Euclidean distance between the RIS and RSU. $\phi_{I,R} = \dfrac{x_I - x_R}{d_{I,R}}$ is cosine of the angle of arrival of signal from RSU to RIS. d is the separation between RIS elements and λ is the carrier wavelength.

Similarly, we can compute the channel gain between the RIS and vehicles which is denoted by $h_{I,v}^n \in \mathbb{C}^{M\times 1}$ as in Eq (6.3).

$$h_{I,v}^n = \underbrace{\sqrt{\rho (d_{I,v}^n)^{-\alpha}}}_{\text{path loss}} \underbrace{\sqrt{\frac{K}{1+K}}\bar{h}_{I,v}^{n\ \text{LoS}}}_{\text{Rician fading}}, \forall v, n \in N, \tag{6.3}$$

$$\bar{h}_{I,v}^{n\ \text{LoS}} = \underbrace{\left[1, e^{-j\frac{2\pi}{\lambda}d\phi_{I,v}^n}, ..., e^{-j\frac{2\pi}{\lambda}(M-1)d\phi_{I,v}^n}\right]^T}_{\text{array response}}, \forall v, n \in N, \tag{6.4}$$

where $d_{I,v}^n$ is the euclidean distance between the RIS and vehicle v at time slot n and $\phi_{I,v}^n = \dfrac{x_I - x_v^n}{d_{I,v}^n}$. Finally, we assume the channel is completely blocked between the RSU and vehicles in that zone similar to [121][2].

Denote the phase-shift matrix of the RIS in the nth time slot as $\theta^n = \text{diag}\{e^{j\theta_1^n}, ..., e^{j\theta_M^n}\}$, where θ_m^n is the phase-shift of the mth reflecting element $m = 1, 2, , M$. Due to the hardware limitations, the phase-shift can only be selected from a finite set of discrete values. Specifically, the set of discrete values for each reflecting RIS element can be given as $\theta_m^n \in \Omega = \{0, \frac{2\pi}{Q}, \ldots, \frac{2\pi(Q-1)}{Q}\}$, where $Q = 2^b$ and b is the number of bits that control the number of available phase-shifts for the RIS elements. Hence, the signal to noise ratio (SNR) is:

$$\lambda_v^n = \frac{P|h_{I,R}^H \theta^n h_{I,v}^n|^2}{\sigma^2}, \forall v, n \in N, \tag{6.5}$$

[2]In this work, we assume that the channel gain between RIS and vehicles is fixed within one time slot.

where P is the transmission power of the RSU and σ^2 is the thermal noise power.

Then, we can compute $h_{I,R}^H \theta^n h_{I,v}^n$ based on Eq (6.2) and Eq (6.3).

$$h_{I,R}^H \theta^n h_{I,v}^n = \frac{\rho \frac{K}{K+1}}{\sqrt{(d_{I,i}^n)^\alpha}\sqrt{(d_{I,R})^\alpha}} \times \sum_{m=1}^{M} e^{j(\theta_m^n + \frac{2\pi}{\lambda}(m-1)d\phi_{I,v}^n - \frac{2\pi}{\lambda}(m-1)d\phi_{I,R})}, \forall v, n \in N. \tag{6.6}$$

Now, instantaneous bit rate given to each vehicle is calculated as.

$$l_v^n = j_v^n \log_2(1+\lambda_v^n), \forall v, n, \tag{6.7}$$

where $j_v^n \in [0,1]$ is a decision variable to schedule the resources of RSU to vehicle v at time slot n. Hence, $j_v^n = 1$ means vehicle v is served at time slot n and 0 otherwise. Now, the average bit rate each vehicle receives throughout its sojourn time can be computed by the following.

$$z_v = \frac{1}{H_v} \sum_{n=1}^{N} l_v^n, \forall v, \tag{6.8}$$

where H_v is the residence time of vehicle v in the dark zone. Next, we formally define our problem as:

Definition 1 *Assume a flow of vehicles travelling through a dark zone. The vehicles are demanding connection to a remote RSU. Meanwhile, a RIS is deployed at specific point, i.e., on a building, where it possesses a strong LoS with the RSU and the dark zone. The RIS has a certain number of elements where the operator can tune their coefficients to provide service for vehicles in order to enhance channel gains and improve bit rates. During a certain time horizon (encompassing multiple time slots), what is the best RSU wireless scheduling and phase-shift configuration for the RIS elements such that the minimum average bit rate provided to vehicles is maximized.*

6.4 Mathematical Formulation

In this section we formulate the problem of RSU wireless scheduling and RIS element tuning mathematically. Let A_v and D_v denote the arrival time and departure time of vehicle v, respectively. The notations used in this corresponding are listed in Table 6.1.

Table 6.1: Mathematical notations in RIS enabled vehicular communications

Parameters	
x_I, y_I, z_I	RIS location
x_R, y_R, z_R	RSU location
x_v^n, y_v^n, z_v^n	Vehicle v location at time slot n
N	Time horizon consists of smaller time slots.
V^n	Set of available vehicles during time slot n
C	Number of RSU channels
H_v	Vehicle v residence time
$\phi_{I,R}$	Angle of arrival at RIS from RSU
$\phi_{I,v}^n$	Angle of arrival between the RIS and vehicle v at time slot n
α	Path loss exponent
P	Transmission power of the RSU
ρ	Median of the mean path gain at reference distance = 1m
σ	Thermal noise power
b	Number of control bits for the RIS elements
Q	Number of RIS phase-shift patterns
Variables	
j_v^n	1: if vehicle v is scheduled for service by the RSU at n and 0 otherwise
θ_m^n	RIS element m phase-shift angle at time slot n

The optimization problem alongside the objective function can be mathematically written as follows. For the sake of clarity, let $\Theta = \{\theta^1, \theta^2, \ldots, \theta^N\}$ and $J = \{j_v^n, \forall v, n \in N\}$.

$$\max_{\Theta, J} \quad \{\min z_v\} \tag{6.9a}$$

$$\text{s.t.} \quad \sum_{v=1}^{V^n} j_v^n \leq C, \forall n \in N, \tag{6.9b}$$

$$j_v^n \in [0,1], \forall n \in N, v, \tag{6.9c}$$

$$j_v^n \leq \max(n - A_v, 0), \forall n \in N, v, \tag{6.9d}$$

$$j_v^n \leq \max(D_v - n, 0), \forall n \in N, v, \tag{6.9e}$$

$$\theta_m^n \in \Omega, \forall n \in N, m \in M. \tag{6.9f}$$

Here, the objective function, Eq (6.9a), is max-min which translates to maximizing the minimum average bit rate. Constraint (6.9b) ensures that the number of channels scheduled to vehicles is no more than that available at the RSU. Constraint (6.9c) allows vehicles to be served via one channel only. Constraints (6.9d) and (6.9e) make sure that vehicle can only be served via RIS while it is within the area of the dark zone. Finally, constraint (6.9f) restrains the number of phase-shift values. Now, the problem is non-convex due to the discrete RIS element phase-shift optimization. Also, the phase-shift matrix is hard to be solved. For instance, if the phase-shift is tuned to optimally serve the first vehicle, the other ones might receive less quality and vise versa. Furthermore, in this problem, it is hard to eliminate the coupling relationship between phase-shift configuration and wireless scheduling. In addition, the information of vehicles such as their arrival, speed, and departure, are unknown in advance. Due to the dynamic nature of the environment, it is impractical to assume such information is given. Hence, a effective solution mechanism has not only to deal with the difficulties of such problem, but it has also to predict for the hidden parameters.

In order to address the above challenges, we resort to Deep Reinforcement Learning (DRL) with multi-binary action space to find a policy that maximizes the minimum average bit rate for vehicles. However, if DRL is used to solve for the two decisions of resource scheduling and phase-shift matrix, the action space will be equal to all the possible combinations of wireless scheduling and phase-shift patterns for M elements which is unbearably large. Such massive action space

would increase the DRL agent difficulty to learn. Similar to [125, 126], a more practical solution approach can be realised by delegating one decision to an optimization technique while dedicating the second one to machine learning based approach. In particular, the DRL agent first determines which vehicles are going to be served at time slot n. While, BCD algorithm is invoked to configure the phase-shift matrix such that the service offered to the scheduled vehicles is optimized. Next, the solution approach, in details, will be discussed.

6.5 Solution Approach

The solution approach for joint resource scheduling and passive beamforming is presented in this section. First, we decompose the aforementioned problem into two sub-problems, the first sub-problem is due to the resource scheduling and second one corresponds to the phase-shift matrix of the RIS. The information and mobility of the upcoming vehicles are unknown in advance. That is, solving the first sub-problem is quite challenging. Hence, we resort to DRL to observe the environment and tackle multi-user RSU scheduling. Next, the RIS elements are tuned based on Block Coordinate Descent (BCD) [127–129]. The details of our solution methodology are laid out in the next sections.

6.5.1 DRL for Wireless Scheduling

For the DRL, the problem is formulated as MDP (similar to Section 3.4.1) where the state, action, reward, are defined below. First, let us define four new notations; $f^n, \forall n \in N$ which denotes the current minimum average bit rate until time slot n, $k_v^n, \forall n \in N, v$ denotes the speed of vehicle v at time slot n, $z_v^n \forall n \in N, v$ is the current average bit rate of vehicle v until time slot n, η is the largest number of vehicles existing simultaneously, and U is the number of possible actions. Now, the state, action, reward can be explained as:

- State S: The state at time slot n, $s^n \in S$, is a vector that indicates current minimum service provided up to $n-1$ (f^n), the speeds of the existing vehicles ($k_v^n, \forall v$) in that time slots, their cumulative average bit rates up to n ($z_v^n, \forall v$), and their locations ($x_v^n, \forall v$). The state s^n can be

expressed as:

$$s^n = [f^n, \underbrace{k_1^n, z_1^n, x_1^n}_{v=1}, \underbrace{k_2^n, z_2^n, x_2^n}_{v=2}, ..., \underbrace{k_\eta^n, z_\eta^n, x_\eta^n}_{v=\eta}]. \tag{6.10}$$

- Action A: The action taken at time slot n, $a^n \in A$ is a vector of size η where η is the maximum number of vehicles available within the zone in one time slot. Also, the sum of vector a^n should be equal to C (to enforce constraint (6.9b)). For example, if $a^n[0] = 1$, then the first existing vehicle is being served at time slot n and so forth. The number of actions can be computed by $U = \eta!/(C!(\eta - C)!)$. The possible combinations of action are similar to the example below.

$$\underbrace{[\underbrace{0}_{1}, 0, 1, ..., \underbrace{1}_{\eta}]}_{1}, \underbrace{[0, 1, 1, ..., 1]}_{2}, ..., \underbrace{[1, 1, 0, ..., 0]}_{U} \tag{6.11}$$

- Reward: The immediate reward, r^n, is computed as follows. During the beginning of the operational phase, $r^n = 0$ until the first vehicle departs. For the first vehicle, the step reward is equal to its average bit rate. Henceforth, whenever any vehicle leaves the dark zone, the step reward is given as a penalty if and only if that vehicle has received less average bit rate than the other vehicles which left previously. It is worth noting that since the agent seeks to maximize the minimum average bit rate, it does not count reward if a vehicle received higher bit rate than others.

Remark *selecting an action is a non-trivial task for the problem explained above. Actually, since our objective is to maximize the minimum average bit rate for all vehicles, it is not easy to decide which vehicles to serve at each time slot. For instance, a vehicle has just entered may have plenty of time to be served later while a vehicle near the end of the road segment may have no much time to receive service. In contrary, a vehicle located at the end is way far than those vehicles near the RIS. Hence, the latter can receive much higher bit rate if selected to be served. Moreover, if the RSU postpones the service for one vehicle, other vehicles may arrive, therefore, that vehicle will have less chances to be served later. In addition, in our work, we consider multi-user communication where more than one vehicle might be scheduled by the RSU simultaneously which makes the action space more complicated. Hence, the agent needs to interact with the environment*

and try different actions and scheduling policies in order to figure out which one attains the best cumulative reward.

For DRL, we exploit PPO to develop our agent as laid out in Algorithm 13. First, the agent initializes random sampling policy and value function for the neural networks. Then, in each epoch, the agent observes the environment which consists of the set of vehicles and their information, minimum average bit rate achieved up to n. Then at each time slot n, the agent selects an action which is a binary vector that determine which set of vehicles will be served via the RSU. Based on that action, the BCD algorithm is then invoked to configure the phase-shift matrix in order to maximize the channel gain. Eventually, the time step reward is worked out which has three cases. First, if no vehicle has departed yet, $r^n = 0$. Second, if the very first vehicle departs, the reward is set to its average bit rate. Third, the consecutive vehicles leaving the area will be accounted as a penalty if and only if their average bit rate is less than f^n when they have departed.

Algorithm 13: Proposed DRL for Scheduling

Inputs: N, v, Learning Rate, γ, ε.
Outputs: RSU resource scheduling and θ^n.
Initial policy π with random parameter θ and threshold ε
Initial value function V with random parameters ϕ
for *each episode* $k \in \{0,1,2,...\}$ **do**
    for $n : \{0,1,2,...,N\}$ **do**
        Observe state $f^n, k_v^n, z_v^n, x_v^n, \forall v \in V^n$.
        Select action a^n from $\pi_{\theta_{old}}$
        Assign channel to vehicle v if it is scheduled to be served.
        Configure RIS phase-shift matrix using Algorithm 14.
        if *Vehicle v is the first one to leave* **then**
            Set $r^n = z_v$
        if *Vehicle v departed and* $z_v < f^n$ **then**
            Set $r^n = f^n - z_v$
        else
            $r^n = 0$
    Compute the advantage function.
    Optimize the clipped objective.
    $\theta_{old} \leftarrow \theta$.

6.5.2 BCD for RIS Phase-Shift Coefficients

Block coordinate descent (BCD) has been proposed in the literature to solve for RIS phase-shift matrix [128, 129]. In this correspondence, we aim to leverage BCD to maximize the sum of immediate sum of bit rates of all vehicles selected to be served at time slot n.

$$\sum_{v=1}^{V^n} j_v^n l_v^n, \forall n. \tag{6.12}$$

To do so, Algorithm 14 receives the action selected by DRL in Algorithm 13, J^n. Once the decision is taken by the agent, the BCD is then called to optimize the phase-shift matrix in iterative way. In each iteration, a sequence of block optimization procedures are performed. In each one, all elements are fixed while one is optimized by checking all its possible values, 2^b. The one that maximizes the objective will be selected. After that, the next element will be selected to optimize and so forth. This operation is iterated until Eq (6.12) has converged. In practice, Algorithm 13 needs one iteration to surpass 95% threshold of its maximum performance. Hence, this algorithm is pretty robust in dealing with the phase-shift coefficients.

Algorithm 14: BCD to Tune the RIS Phase-Shift Matrix

```
Inputs: J^n.
Outputs: θ^n
while Eq (6.12) not converged do
    for m = 1, ..., M do
        Fix m', ∀m' ≠ m, m' ∈ M
        Set θ_m^n = arg max_Ω Eq(6.12)
    Obtain Eq (6.12)
```

Concerning the complexity of Algorithm 14, it is $O(IM2^b)$ where I stands for the number of iterations until Eq (6.12) converges. In details, there are three loops in this Algorithm; first is the number of iterations, second is the number of RIS elements, and third is the number of angles available to control each element. An experiment is conducted to study the BCD performance and the results are shown in Fig. 6.2. In this experiment, we vary the number of RIS elements from

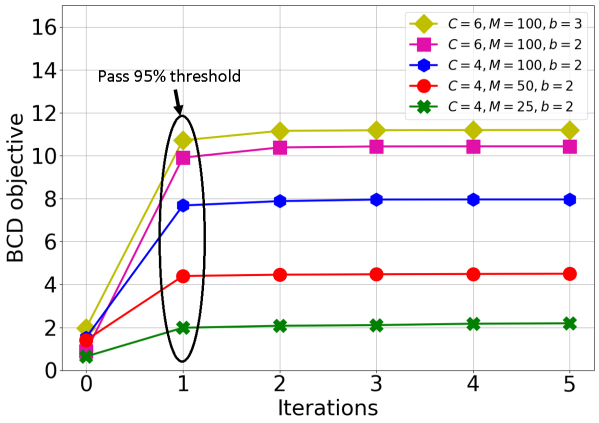


Figure 6.2: BCD convergence over iterations with different RIS elements (M), users (C), and quantization levels (b).

25 to 100 elements. Moreover, we try different number of users (C) and control bits (b). Based on the outcomes, we can approximate the complexity of Algorithm 14 to $O(M2^b)$. The complexity can further be approximated to $O(M)$ based on the fact that a b of 3 is enough [106][3].

6.6 Case Study: RIS Placement and Vehicle Positioning

In the literature, RIS is often proposed to tune static wireless environments such as user equipment or IoT devices. However, implementing RIS in dynamic medium is far more challenging owing to the highly sensitivity of RIS phase-shift alignment. Therefore, we carry out two studies to address the practical RIS placement and the impacts of vehicle positioning accuracy in the context of RIS-assist vehicular communications.

6.6.1 RIS Placement

In this section, we discuss the issue of placing RIS at different places. We statistically study how placing the RIS at different locations can actually improve or worsen the overall performance. Then, we will see what is the optimal location to situate the RIS. We start off by a hypothesis stating that the optimal RIS placement is the closest one to the RSU. This hypothesis is based on initial observations that indicate the shorter the distance between the RIS and RSU, the best channel gain

[3]Note that, based on our experiments, we found that $b = 2$ is enough to achieve high performance in our context as demonstrated in Section 6.7.

can be achieved. In order to prove our claim, we are going to derive it mathematically and then back it up with simulation experiments.

Theorem 1 Given $x_I < x_I^{'}, x_R < x_I, x_R < x_v, \forall v$, the inequality of $z_v(x_I) > z_v(x_I^{'}), \forall x_I < x_I^{'}$ always holds.

Proof: Here, for simplicity, we take a RIS and try to place it at different points to serve a single vehicle as shown in Fig. 6.3. Hence, the RIS elements will always be tuned to maximize the channel gain for that vehicle. Fortunately, Eq (6.6) can be obtained in a closed form for a single user [130].

$$\theta_m^n = \frac{2\pi}{\lambda}(m-1)d\phi_{I,v}^n + \frac{2\pi}{\lambda}(m-1)d\phi_{I,R}, \forall m \in M, n. \tag{6.13}$$

Where the phase-shifts of RIS cancel out the ones of RIS-RSU and RIS-vehicle, Eq (6.6) can be rewritten as:

$$h_{I,R}^H \theta^n h_{I,v}^n = \frac{\rho \frac{K}{1+K} M}{\sqrt{(d_{I,i}^n)^\alpha}\sqrt{(d_{I,R})^\alpha}}. \tag{6.14}$$

Since ρ, M, and α are constant, the only factors that remain variable are $d_{I,v}^n$ and $d_{I,R}$ which denote the distances of RSU-RIS-vehicle. We can also notice that Eq (6.14) is a decreasing function with respect to distances. Now, we need to prove that $z_v(x_I) > z_v(x_I^{'}), \forall x_I < x_I^{'}$. To do so, let us assume a vehicle has $H_v = N$ (we assume there is a single vehicle on the road). To this end, $z_v(x_I) > z_v(x_I^{'})$ is greater when the sum of bit rates received throughout H_v is larger.

$$l_v^1(x_I) + l_v^2(x_I) + ... + l_v^N(x_I) > l_v^1(x_I^{'}) + l_v^2(x_I^{'}) + ... + l_v^n(x_I^{'}), \forall x_I < x_I^{'}. \tag{6.15}$$

Next, for clarity, let $Y = \dfrac{P\rho^2\left(\dfrac{K}{1+K}\right)^2 M^2}{\sigma^2}$ which is a invariant value. Hence, Eq (6.15) can be rewritten as:

$$\begin{aligned} &log_2(1 + \frac{Y}{(d_{I,v}^1)^\alpha (d_{I,R})^\alpha}) + log_2(1 + \frac{Y}{(d_{I,v}^2)^\alpha (d_{I,R})^\alpha}) + ... + log_2(1 + \frac{Y}{(d_{I,v}^N)^\alpha (d_{I,R})^\alpha}) > \\ &log_2(1 + \frac{Y}{(d_{I,v}^{'1})^\alpha (d_{I,R}^{'})^\alpha}) + log_2(1 + \frac{Y}{(d_{I,v}^{'2})^\alpha (d_{I,R}^{'})^\alpha}) + ... + log_2(1 + \frac{Y}{(d_{I,v}^{'N})^\alpha (d_{I,R}^{'})^\alpha}), \forall x_I < x_I^{'}. \end{aligned} \tag{6.16}$$

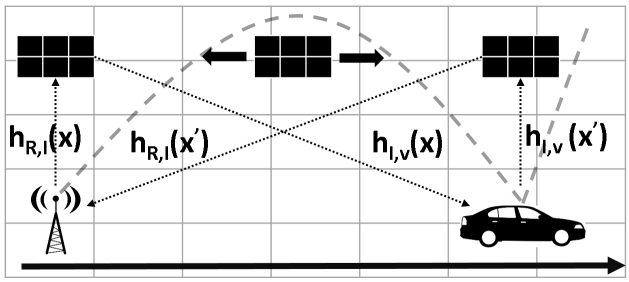


Figure 6.3: RIS placement with corresponding channel gain, in the background, in 1D space.

Now, only the distance would affect the sum of bit rates over N. To facilitate the expressions, let us further assume one dimensional environment. Hence, $d_{I,R} = |x_I - x_R|$ and $d^n_{I,v} = |x^n_v - x_I|$. We can also assume $\alpha = 1$. Therefore, as absolute values are multiplicative:

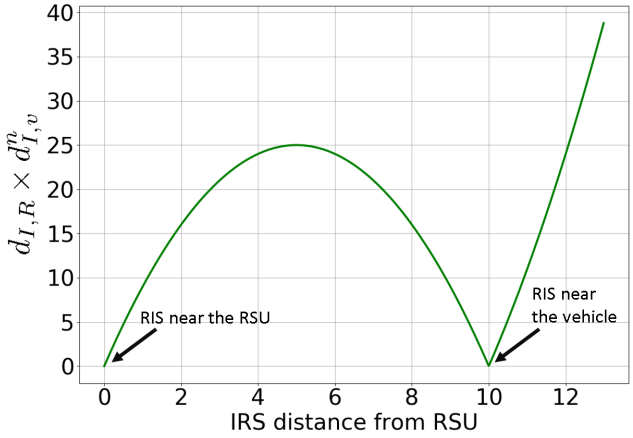
$$d_{I,R} \times d^n_{I,v} = |x_I x^n_v - x^2_I - x_R x^n_v + x_R x_I|. \tag{6.17}$$

In Fig 6.4 (a), note that Eq (6.17) has its lowest value when $x_I = x_R$ or $x_I = x^n_v$. The first term can always be achieved, during the entire time horizon ($\forall n \in N$) and for all vehicles, as long as x_I and x_R are fixed. However, the second one, $x_I = x^n_v$, as vehicles driving, only holds true for one time slot n and for a specific vehicle. Therefore, with $H_v > 1$, we can confidently say:

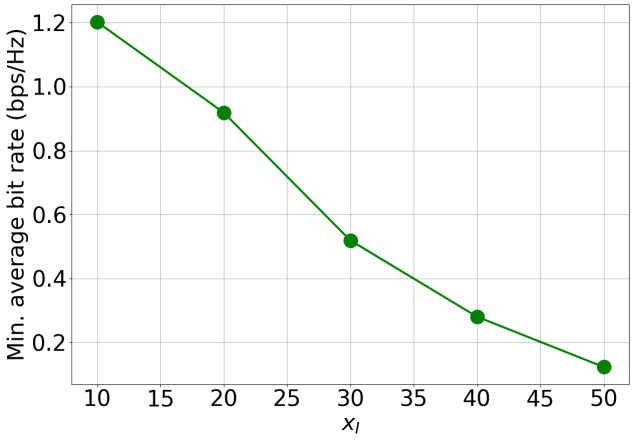
$$\sum_{n=1}^{N} log_2(1 + \frac{Y}{(d^n_{I,v})^\alpha (d_{I,R})^\alpha}) > \sum_{n=1}^{N} log_2(1 + \frac{T}{(d'^n_{I,v})^\alpha (d'^n_{I,R})^\alpha}), \forall x_I < x'_I. \tag{6.18}$$

Consequently, the right side is greater the left side which completes the proof.

In Theorem 1, we have shown that, for a single vehicle, the ideal place for the RIS is to be as closer as it can to the RSU. However, in practice, there exists several constraints that force the RIS to be distant from the RSU. For example, the LoS has to be clear between the RIS and RSU and between the RIS and the vehicles it serves. Otherwise, the wireless links would be highly disturbed and the RIS will lose its functionality. In line with our proof and discussion above, we carry out three experiments to see the achievable minimum average bit rate for multi-user scenario. We set $x_R = 0$, $x_I = [10, 20, 30, 40, 50]$. Vehicles are generated by SUMO. The outcomes are displayed in Fig. 6.4 (b). One can observe that with RIS closer to RSU, the performance was much higher. However, this performance started to degrade dramatically as the RIS moves away from the RSU.



(a) Eq (6.16) curve with varying x_I values.



(b) Multi-user scenario with varying x_I values.

Figure 6.4: Simulation results for Eq (6.16) and RIS placement.

6.6.2 Vehicle Positioning Precision

One of the challenging issues in vehicular communications is vehicle positioning in such highly dynamic environment. Hence, in the context of RIS-assist vehicle communication, inaccurate vehicles positioning might lead to severe consequences that negatively impact the channel gain. Thus, we attempt to understand whether it is possible to leverage the emerging technologies related to vehicle positioning such as 3D-LIDAR (Light Detection and Ranging), Global Positioning System, etc., for the benefits of accurately estimating real vehicle positions to enable RIS-aided vehicular communications.

First, based on the channel gain equations laid out in Eq (6.6), the impact of inaccurate vehicle positioning on the RIS-based system performance will be examined. Let Δ be the error in vehicles positioning, then, the estimated position is $x_v^n \pm \Delta$. It is worth noting that in this context, the small error (Δ) in vehicle positioning will not have significant impact on the cascaded channel (RSU-RIS-vehicle) path loss. Thus, we only study the phase-shift angle deviation (difference between accurate and estimated angles of arrival). Next, we highlight the components in Eq (6.6) that are affected by inaccurate vehicles positioning. The next equation describes the phase-shift multiplication of the two angles of arrival, $\phi_{R,I}$ and $\phi_{I,v}^n$, with the RIS elements at the real position.

$$\theta_m^n + \frac{2\pi}{\lambda}(m-1)d\frac{x_I - x_v^n}{\sqrt{(x_I - x_v^n)^2 + Y}} - \frac{2\pi}{\lambda}(m-1)d\phi_{I,R}, \forall m \in M, n \in N. \tag{6.19}$$

Next, we formulate the same equation, yet, at the estimated position.

$$\theta_m^n + \frac{2\pi}{\lambda}(m-1)d\frac{x_I - x_v^n \pm \Delta}{\sqrt{(x_I - x_v^n \pm \Delta)^2 + Y}} - \frac{2\pi}{\lambda}(m-1)d\phi_{I,R}, \forall m \in M, n \in N. \tag{6.20}$$

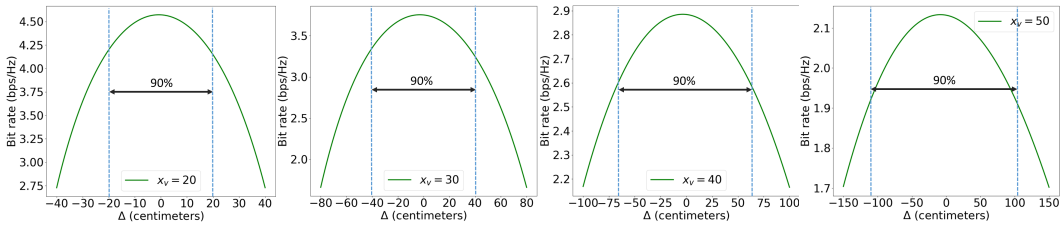
After subtracting the two equations, real and estimated one (Eq (6.19) - Eq (6.20)), we end up with the following.

$$\frac{2\pi}{\lambda}(m-1)d\left(\frac{x_I - x_v^n}{\sqrt{(x_I - x_v^n)^2 + Y}} - \frac{x_I - x_v^n \pm \Delta}{\sqrt{(x_I - x_v^n \pm \Delta)^2 + Y}}\right), \forall m \in M, n \in N. \tag{6.21}$$

Eq (6.21) clearly states that the cosine angle of arrival between RIS and vehicles is affected by

the error in vehicle positioning. It is also noted that this impact occurs for all the RIS elements. How much this deviation from the real position affects the performance is what we answer next.

In order to show the impact of that error Δ, an experiment is conducted to realise how the bit rate is changed while varying the value of Δ with various vehicle positions (the distance from the vehicle to the RIS). The results are displayed in Fig. 6.5 and indicate that the RIS can keep up to 90% of its performance with error Δ ranging form 20 to 100 centimeters depending on the vehicle position. It is observed that distant vehicles from the RIS are less impacted by Δ as $\phi_{I,v}^{n}$ is less influenced by Δ when the distance between the vehicle and RIS is larger. One can also see in this figure that as Δ grows up, the bit rate decreases. Yet, the RIS elements can be tuned with plausible Δ values to maintain most of the original performance expected from RIS deployment. According to [131], vehicular carrier-phase differential Global Navigation Satellite System (GNSS) positioning can estimate vehicle positions with accuracy of less than 17 centimeters accompanied by a success rate reaches up to 95%. As shown in Fig. 6.5, the minimum tolerance (to achieve 90% of the performance) of vehicle positioning is 20 centimeters which is compatible with GNSS precision. Note that, for simplicity, in this work, x_v^n is assumed to be accurately estimated.



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

Figure 6.5: Inaccurate vehicle positioning effects on the bit rate at different positions ((a) $x_v = 20$ (b) $x_v = 30$ (c) $x_v = 40$ (d) $x_v = 50$).

6.7 Simulation and Evaluation

6.7.1 Simulation Setup

As mentioned earlier, we use SUMO to mimic vehicular environment. Two flows of vehicles are generated; one with normal speed (max speed 50Kph) and the other with slow speed (max speed 30Kph). For the Deep Reinforcement Learning, 3 linear layers are used with tanh as activation

Table 6.2: Simulation parameters in RIS enabled vehicular communications

Parameter	Value
Road segment length	100 m
Arrival rate	0.2 Veh/sec
σ^2	-110 dBm
K	10 dB [123]
α	4
P	20 dBm
C	3
ρ	10 dBm
M	100
b	2 bits
x_I, y_I, z_I	10, 20, 10
x_R, y_R, z_R	0, 40, 10
y_v^n, z_v^n	20, 1

function for the middle layers and softmax for the output layer. Internal layers contain 64 units each and Adam optimizer is incorporated to minimize the loss function. Learning rate is set to 0.002, γ to 0.08, and clip to 0.02. The results were averaged over 500 tests. The remaining parameters used in our study are listed in Table 6.2 (unless otherwise indicated).

6.7.2 Numerical Results

First, we attempt to see the behaviour of the DRL agent. As illustrated in Fig. 6.6, the cumulative reward, here represents minimum average bit rate, is remarkably increasing as the agent is exposed to more epochs/iterations. One can note that after around 7000 iterations, the system starts to converge.

In order to validate the performance of the proposed algorithm, it is compared with three other benchmarks as follows

- **Greedy Scheduling with BCD (GS-BCD)**: In this scheme, the vehicle schedule sub-problem

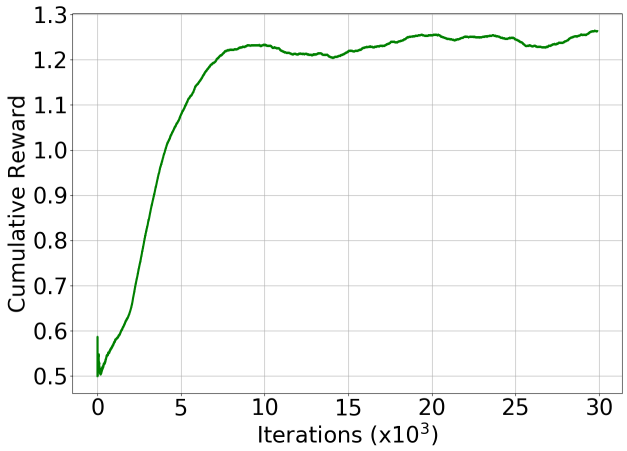


Figure 6.6: Convergence over time.

is solved with greedy algorithm; meanwhile the passive beamforming sub-problem is obtained using the proposed BCD scheme. The greedy algorithm can be explained as follows. At each time slot $n \in N$, a greedy algorithm ranks the set of vehicles V^n based on their cumulative bit rate achieved up to n (z_v^n). Then, those with the lowest average bit rates will be scheduled to be served in the following time slot.

- **Random Scheduling with BCD (RS-BCD)**: In this scheme, each time slot, the vehicles are randomly scheduled. While the proposed BCD is used for obtaining the passive beamforming at the RIS.

- **DRL with Random Phase-Shift Matrix (DRL-RPS)**:The proposed DRL algorithm is used to schedule RSU resources without any optimization over phase-shift for the RIS elements (Random values for the RIS elements' phase-shift).

It is worthwhile to compare with these baseline methods as they will show us how the performance would be if one of the two sub-problems are solved via an alternative widely-used method such as greedy or random while the second one is solved with the same method we propose.

The effect of RIS number of elements M is first studied. As demonstrated in Fig. 6.7, with small number of elements, the achievable minimum average bit rate is very limited. However, as more elements are incorporated, the gain starts to grow up gradually, especially for our proposed solution approach. Another insight one can notice is the gap between the proposed solution with

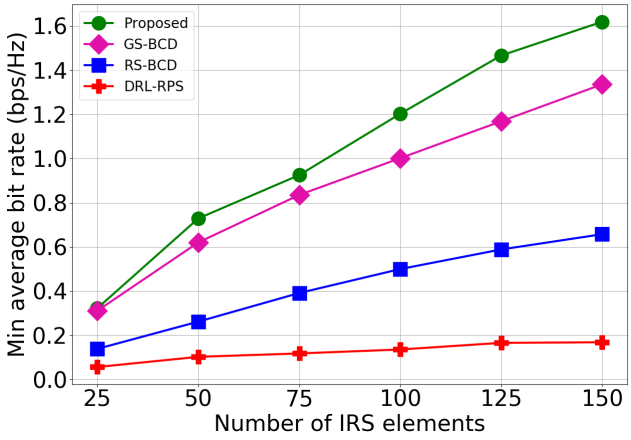


Figure 6.7: RIS number of elements M effects on network performance.

other methods over different M values. It is very apparent that this gap widens proportionally as M increases. With $M = 150$, the difference between the proposed one and the greedy algorithm is about 17%. Meanwhile, GS-BCD seems to have also a good performance compared to the other two methods. The reason behind that is GS-BCD attempts to reduce the minimum average bit rates for those vehicles with low bit rate levels. In addition, as GS-BCD leverages BCD, it also benefits from the good RIS configuration. RS-BCD, on the other hand, comes in the third place with clear gap from GS-BCD as it does not take into account low bit rate vehicles. At the end, DRL-RPS attains very poor performance that indicates that without a proper RIS configuration, the performance would be very poor even if the wireless scheduling is done carefully.

In our next experiment, we vary the value of b for practical RIS. When b is high, more phases are available for the configuration which is better for optimal RIS tuning. As it can be seen in Fig. 6.8, larger b means better performance. Yet, one can also notice that b of 2 or 3 can almost obtain the highest gain. These behaviours have also been highlighted in other works related to non-dynamic environment [105]. In the same figure, we can see that the proposed solution always achieves the highest performance regardless of b. Indeed, the difference can reach up to 19% from GS-BCD. In the meantime. RS-BCD could only achieve minimal gain with larger b and still below 0.6 of minimum average bit rate in all scenarios. In contrast to the other methods, DRL-RPS was unable to add any gain with larger b. That is due to the fact that this method does not consider RIS element tuning in the first place. Hence, higher b values may also mean higher probability of falling to align

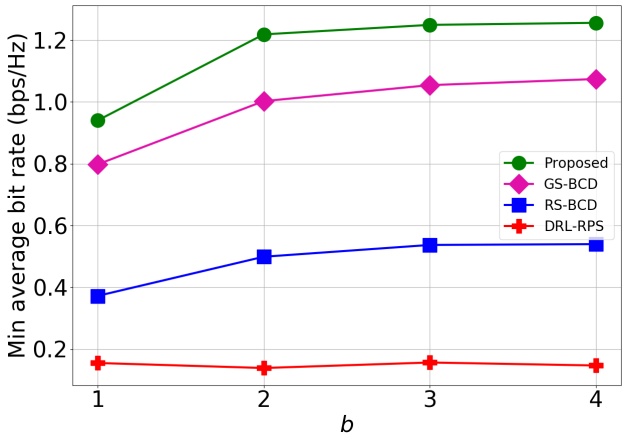


Figure 6.8: Discrete quantization levels effects.

with the vehicles as the options of angles for the RIS elements become larger.

Next, the impacts of road density on the network is studied. Here, we vary the arrival rates of vehicles which results in more or less vehicles available simultaneously within the road segment. Intuitively, with small arrival rate, the RSU and RIS can better serve the vehicles. As seen in Fig. 6.9, minimum average bit rate is slightly above 1.4 bps/Hz. However, as the road segment becomes more dense, the value degrades to approach approximately 1 bps/Hz. For the other methods, we can observe similar behaviours expect for GS-BCD where its gain seems to saturate at very low arrival rates. That is because GS-BCD does not consider the distance between the RIS and vehicles. It always assigns the resources for those with less z_v^n regardless of their location or speed. Therefore, the wireless resources might be wasted on far vehicles instead of making benefit by serving nearest ones. Also, selecting two or more vehicles with relatively large gap between them reduces the efficiency of the RIS to serve both of them since the RIS needs to maximize the sum of immediate bit rates which is undesirable in such scenario[4]. This especially appears when less numbers of vehicles exist on the road and the RSU oftentimes schedules for distant vehicles. In contrast, RS-BCD attains steep increase in minimum average bit rates with low road density since there will be less vehicles and the probability of vehicles being served is much higher. This impact is much less significance with DRL-RPS due to the miss alignment in phase-shifts with the vehicles.

[4]Note that, the immediate average bit rate, denoted by z_v^n, does not necessarily reflect the ultimate average bit rate of vehicle v. It only represents what was the average bit rate up to n which depends on the time elapsed since vehicle v arrival. Intuitively, this value decreases over time as the elapsed time increases, and this issue is not taken into consideration by GS-BCD.

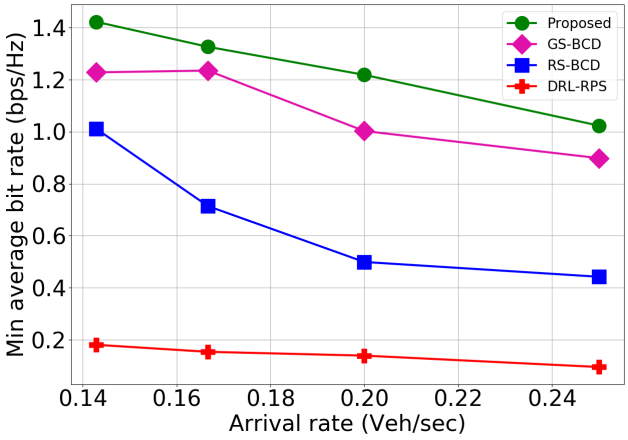


Figure 6.9: Min average bit rate values over different vehicle arrival rates.

Finally, one of the insightful indices to use in similar problem of max-min is Jain's fairness index [132]. The formula for this index is:

$$\frac{(\sum_{v=1}^{V} z_v)^2}{V \sum_{v=1}^{V} (z_v)^2} \tag{6.22}$$

Then, we conduct an experiment to see the levels of jain's fairness attained by the four algorithms. We can notice in Fig. 6.10 that our proposed solution achieved the highest level of fairness in comparison to the other methods. But, one can also see that the four methods, in fact, obtain high levels in general. For GS-BCD, it actually attempts to reduce the discrepancies in minimum average bit rates among all vehicles, hence, it enhances the fairness. RS-BCD, on the other hand, schedules the resources at random with uniform distribution which also improves the fairness. Finally, DRL-RPS leverages our DRL agent which indeed tries to maximize the original maxmin problem. That is, all the methods are able to maintain nice levels of fairness among the vehicles. Despite this fact, it is still true that only our proposed solution approach can achieve the highest fairness levels while notably maximizing the minimum average bit rates through considering the coupling effects of the two sup-problems.

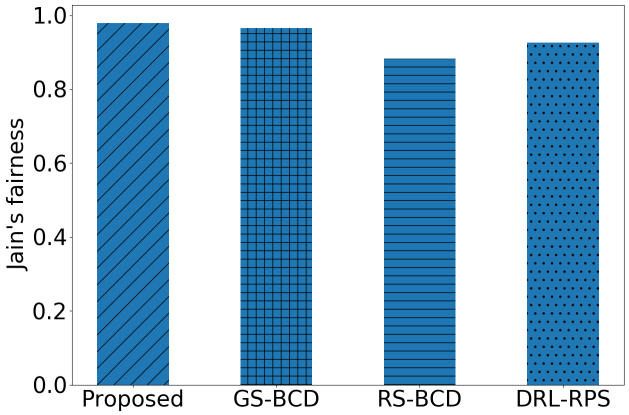


Figure 6.10: Comparison for the Jainś fairness ($M = 100$).

6.8 Summary

We have investigated the area of RIS integration with vehicular communications. That is, the core of this work evolved around a system model that employs RIS to provide favourable wireless experiences for vehicles travelling in a dark zone. The RIS has demonstrated high competence in establishing indirect links between the RSU and vehicles. Throughout this study, we have also seen that DRL is an appealing solution to cope with the highly dynamic nature of such environment and it can adapt to various road conditions and RIS options. In addition, BCD was also leveraged to provide efficient yet robust solutions to the RIS phase-shift matrix. In the numerical results, the performance of our solution method has been analyzed thoroughly by comparing it with other benchmarks. Aside from that, we have also carried out a study on RIS placement to attain optimized wireless communication with the RSU and non-static receivers.

Chapter 7

RIS-assisted Aerial RSU to Serve IoT Devices in Intelligent Transportation Systems

In this work, we build upon our previous contribution in Chapter 6 to further improve the performance of IoT-enabled vehicular networks by leveraging a UAV. In details, in this chapter, we promote the integration between UAVs and terrestrial RIS elements and study the problem of collecting data from ground IoT devices (See Fig. 7.1). A UAV is deployed and its location is adapted to collect sensory data from active devices. Devices are assumed, to conserve energy, to switch between active and passive modes. Devices, for the purpose of facilitating some critical ITS services, are scattered and when a device is active it has sampled an information from a signal that it needs to send to the edge application. As mentioned, a UAV does not have continuous LoS with all devices, and therefore, while it is in direct communication with some devices, other sensors may reach the UAV through an indirect path enabled by the RIS. We seek to concurrently optimize the RIS phase shift for its elements, the scheduling of IoT transmissions as well as the trajectory of the UAV.

Owing to the high complexity of the problem, we propose a solution based on Deep Reinforcement Learning (DRL) to cope with these challenges. DRL has been utilized in similar problems and showed to be both effective and efficient. Additionally, we suggest Block Coordinate Descent

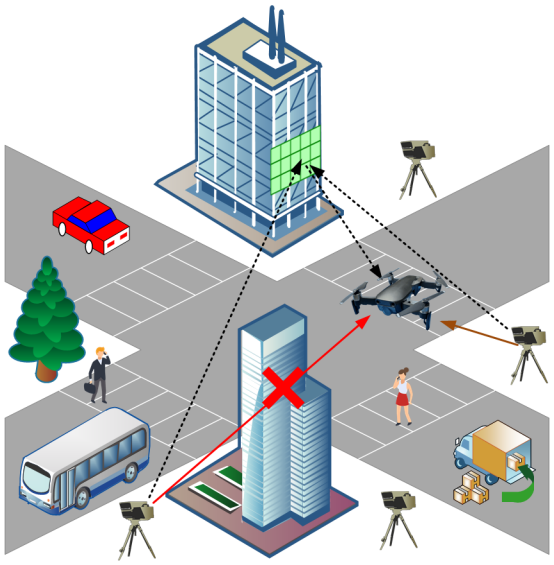


Figure 7.1: System Model.

(BCD) to solve for the RIS phase-shift configuration. Indeed, serving IoTDs using UAV assisted by RIS, to the best of our knowledge, has never been investigated before. The contributions of this work can therefore be summarized as follows:

- A UAV assisted with a RIS is leveraged to provide data gathering services from IoTDs. The UAV trajectory is optimized jointly with resource scheduling and RIS element configuration. This framework is formulated as an optimization problem aiming at maximizing the total number of served IoTDs.

- Due to the high complexity of the formulated problem, which is a mixed-integer non-convex problem as well as the presence of unknown parameters (activation time of IoTDs), the formulated optimization problem is challenging and cannot be directly solved. In order to address this issue, we decompose the original optimization problem into two sub-problems. The first one is to determine the UAV mobility as well as IoTD scheduling. The second sub-problem, and with given UAV place and scheduled indices for the IoTDs, the passive beamforming problem, phase-shift matrix of the RIS, is solved.

- The first sub-problem is converted to Markov Decision Process (MDP) while tackling the

massive action space incurred by UAV mobility in 2D space and IoTD scheduling. In addition, the state space is also defined to consider the UAV positioning and IoTDs' status while satisfying the Markov condition. After that, an agent based on proximal policy optimization (PPO) is developed to solve the MDP. Next, the second problem is solved via Block Coordinate Descent (BCD) where a low complexity method is designed to tune the RIS phase-shift to maximize the total transmission rate for all the IoTDs being served for each transmission.

- Finally, we carry out extensive simulation experiments to analyze the performance of our proposed solution in comparison with four baseline approaches. Furthermore, a study on the impact of RIS presence and size on the UAV energy efficiency is also conducted.

The organization of the rest of this chapter is as follows. Section 7.1 reviews the literature while showing the novel contributions of this work. Section 7.2 lays out the system model and presents our objective function. Section 7.3 describes the solution approach which is based on DRL and BCD. Section 7.4 studies the performance of the solution approach and signals out our observations. Finally, Section 7.5 wraps up the chapter and provides some directions for future work.

Notations: Vectors are denoted by bold-face italic letters. $\text{diag}(x)$ denotes a diagonal matrix whose diagonal element is the corresponding element in x. $\mathbb{C}^{M\times N}$ denotes a complex matrix of $M \times N$. For any matrix M, M^H and M^T denote its conjugate transpose and transpose, respectively. $Pr(A \mid B)$ denotes the probability of event A given event B.

7.1 Literature Review

Serving IoTDs has been studied widely in the literature. However, there are still gaps in terms of introducing RIS-empowered UAV communications to support IoT networks. In this section, we present some relevant publications in the area of data collection using UAVs, specifically for IoTDs, and RIS employment in wireless communication.

7.1.1 UAVs for Data Collection

In the literature, UAVs have been proposed to assist wireless communication for various purposes. The work of [133] suggests a framework for UAVs that provides UAV trajectory planning and

resource allocation while considering the security aspects in the presence of eavesdroppers. [134] presents a formation of a number of UAVs that can collaborate to enhance the the service for a machine. The authors of [135] suggest federated learning to control multiple UAVs with reduced overhead. In [77], the trajectory of UAVs and their resources are jointly optimized to serve time-constrained IoTD. In [136], the authors propose UAV assisted IoTDs by optimizing their data freshness also known as Age of Information or AoI. Using DRL, the altitude of the UAV is changed to maintain a balance between the communication links of BS-UAV and IoTD-UAV. The authors of [137] presents a UAV-based system where UAVs act as edge servers to offer computational resources for the IoTDs. The number of UAVs and their position in 3D space along with IoTD task allocation decisions are optimized jointly to provide service for the IoTD within a limited latency. In [138], an energy-constrained UAV is proposed to aid cellular communication by uploading data from base stations. To this end, the problem is formulated with the objective to maximize transmission throughput while considering resource scheduling, UAV trajectory, and energy budget. The authors in [139] develop a framework that makes the UAV able to offload computational tasks received from IoTDs and others to a nearby facility. This work also addresses the problem of highly complex communication topology in urban environment and solves the concerned problem using DRL. The authors of [140] build a system model to collect IoTDs information using UAVs where the energy consumption of the IoTDs is minimized. This work also accounts for UAV 3D placement, resource allocation, wireless interference, and UAV altitude. In [141], a UAV is deployed to supply energy for ground nodes and optimizing the minimum energy provided. To do so, the trajectory of the UAV is optimized in 1D and the optimal solution is calculated.

7.1.2 RIS-aided Wireless Communication

Reflecting surfaces have gained momentum nowadays among the research communities owing to their notable benefits they bring to wireless communication systems. In [142], the authors propose a relay UAV assisted by an RIS to receive signals from ground users. The RIS helps to improve the coverage and the overall performance of the network. The work of [143] discusses various scenarios for UAV integrated by RIS to improve wireless communications including data collection taking into account sensor nodes' transmit power. The authors of [144] develop a framework of UAV assisted

by RIS using Non-orthogonal multiple access (NOMA) technique to enhance spectrum efficiency. To achieve this, the optimization problem is formulated to do UAV trajectory planning, resource allocation as well as RIS element configuration. Then, DRL is employed to solve the problem. The authors in [145] consider frequency division multiple access for UAV assisted by RIS to optimize the max-sum rate. To do so, the UAV trajectory, RIS phase-shift, and resource allocation are jointly optimized while considering different quality-of-service for users. In [146], a relaying framework for RIS integrated with UAV is investigated where spectral and energy efficiency is sought. Different modes are studied starting with UAV only and end up with UAV-RIS. The problem is formulated where RIS number of elements and UAV altitude are optimized. In [109], the authors studied resource allocation of RIS-aided vehicular communications where they aim to maximize vehicle to base station link quality while guaranteeing vehicle to vehicle communications. The authors of [110] provide analysis for outage probability in RIS-enabled vehicular networks. This chapter derives an expression of outage probability showing that RIS can reduce the outage probability for vehicles in its vicinity. The analysis also proves that higher density roads increase outage probability since passing vehicles can block the communication links. In [111], the authors propose RIS-aided vehicular networks while considering two scenarios to estimate the channels. The first one is by assuming fixed channel estimation within a coherence time. While the second one neglects the small scale fading based on the fact that vehicular positions can be realised in advance. [112] considers constraint discrete phase-shift RIS with two challenges; channel estimation and passive beamforming. The work in [147] proposes a UAV supported by a RIS to prepare a platform for terahertz communications. To this end, UAV's trajectory, RIS phase-shift, resource allocation, and power control are jointly optimized. However, since the environment contains no uncertainties, the problem is solved using iterative algorithm, namely successive convex approximation.

The works presented above consider various applications and scenarios for UAVs and RIS deployment. However, to the best of our knowledge, no work in the literature accounted for UAV trajectory planning in 2D space with the support of a discrete RIS to collect data generated by IoTDs with a target deadline.

7.2 System Model

We consider IoTDs scattered in an urban environment to collect data essential for one or more of emerging smart cities services. The device alternates between an active and passive activation mode to conserve energy and has a period during which its information should be collected before it becomes stale and bear no value [70]. Each device location is denoted by (x_i, y_i, z_i). A UAV is dispatched to gather data from these devices and has a fixed altitude z_U. The UAV can move in two dimensions x_U^n, y_U^n. In order to enhance the communication between the IoTDs and UAV, a RIS equipped with M reflecting elements is placed at (x_R, y_R, z_R) as demonstrated in Fig. 7.1. The list of parameters and variables used is shown in Table 7.1.

7.2.1 IoT Activation

IoTDs are battery-powered, hence, they are energy constrained [148]. Consequently, IoTDs tend to be power-conservative by adopting two operational modes; sleep and active [15]. The duration and frequency of sleep and active modes depend on the applications, for instance, measuring soil humidity occurs once every hours [149] while transmitting autonomous vehicles information happens in a much faster pace to keep the traffic smooth and safe. Other applications may require more frequents activation patterns. In addition, according to [150, 151], the randomness of IoT device activation pattern can be modeled using a uniform distribution.

7.2.2 Communication Model

In this section, we present the channel model, the signal-to-noise-ratio (SNR), and the achievable data rate analysis. In our model, we assume that the IoTDs transmit their data to the UAV in the uplink using frequency division multiple access (FDMA). We denote the total number of available radio resources as C. In addition, the channel model between the IoTDs and the UAV which is referred to as "direct link" and cascaded channel model IoTDs-RIS-UAV which is referred to as "indirect link".

Table 7.1: Mathematical notations

Symbol	Description
I	A set of IoTDs existing in the area.
N	Time horizon.
(x_U^n, y_U^n, z_U)	UAV coordinates at time slots n.
(x_R, y_R, z_R)	RIS coordinates.
(x_i, y_i, z_i)	IoTD i coordinates.
Z_i	Size of data transmitted by device i.
σ^2	thermal noise power.
ρ	Median of the mean path gain at reference distance = 1m.
α	Path loss exponent
P	Transmission power.
K	Racian factor.
δ_i^n	Decision variable for wireless scheduling.
T_i	Active period starting time of IoTD i.
F_i	Active period ending time (deadline) of IoTD i.
C	Number of UAV wireless channels
b	Number of control bits for the RIS elements
Q	Number of RIS phase-shift patterns
$\phi_{R,U}^n$	Angle of arrival at RIS from UAV at time slot n.
$\phi_{R,i}$	Angle of arrival between the RIS and IoTD i.
Ω	Set of available phase shift values for the RIS elements.
ζ	Separation value between RIS elements.
λ	Carrier wave length.
ω_i	Service indicator equals to 1 is IoTD i served and 0 otherwise.
Γ_i^n	signal-to-noise-ration for IoTD i at time slot n.

Direct Link

We assume a channel model for the UAV in urban area where high-rise buildings and other objects appear which could disturb the links between the UAV and IoTDs. Thus, we assume that the link propagation is characterized by both strong Line-of-Sight (LoS) and non Line-of-Sight (NLoS). Here, $S^n_{U\to i} \in \{\text{LoS, NLoS}\}$ indicates the state of the channel between the UAV and IoTD i at time slot n. The probability of having LoS link adopted in this chapter is similar to [99]. Then, we can find the probability of channel states between the UAV and IoTD i.

$$Pr(S^n_{U\to i} = \text{LoS}) = \frac{1}{1+\eta_1 e^{(-\eta_2(\theta^n_{U\to i}-\eta_1))}}, \forall i \in I^n, n, \tag{7.1}$$

where η_1 and η_2 are constant parameters of the environment. $\theta^n_{U\to i} = \frac{180}{\pi}\arctan(\frac{z_U}{\widehat{D}^n_{U\to i}})$ is the angle degree between IoTD i and the UAV at time slot n. Meanwhile, z_U denotes the height of the UAV antenna and $\widehat{D}^n_{U\to i}$ is the horizontal distance between IoTD i and the UAV at time slot n. Moreover, the probability of having NLoS can be obtained from $Pr(S^n_{U\to i} = \text{NLoS}) = 1 - Pr(S^n_{U\to i} = \text{LoS})$. Next, the channel gain for each IoTD i at time slot n is computed as:

$$h^n_{U,i} = \begin{cases} (D^n_{U,i})^{-\beta_1} & S^n_{U\to i} = \text{LoS}, \\ \beta_2 (D^n_{U,i})^{-\beta_1} & \text{otherwise}, \end{cases} \tag{7.2}$$

where $D^n_{U,i}$ is the euclidean distance between the UAV and IoTD i at time slot n and β_1 denotes the path loss exponent and β_2 is the attenuation factor for NLoS. Thus $h^n_{U\to i}$ can also be rewritten as:

$$h^n_{U\to i} = Pr(S^n_{U\to i} = \text{LoS})(D^n_{U,i})^{-\beta_1} + (1 - Pr(S^n_{U\to i} = \text{LoS}))\beta_2 (D^n_{U,i})^{-\beta_1} \tag{7.3}$$

Indirect Link

We consider a uniform linear array (ULA) RIS [121]. In addition, similar to the UAV, the RIS is assumed to have a certain height, z_I. The communication links between the UAV and RIS and that between the RIS and IoTD i are assumed to have a dominant line-of-sight (LoS). Thus, these communication links experience small-scale fading which are modeled as Rician fading with pure

LoS components [123, 124]. Consequently, the channel gain between the UAV and RIS, $h_{R,U} \in \mathbb{C}^{M\times 1}$, can be formulated as follows.

$$h_{R,U}^{n} = \underbrace{\sqrt{\rho (D_{R,U}^{n})^{-\alpha}}}_{\text{path loss}} \underbrace{\sqrt{\frac{K}{1+K}} \bar{h}_{R,U}^{n,\text{LoS}}}_{\text{Rician fading}}, \tag{7.4}$$

where $D_{R,U}^{n}$ is the Euclidean distance between the RIS and UAV. Also, ρ is the average path loss power gain at reference distance $D_0 = 1$m. Also, K is the Rician factor and $\bar{h}_{R,U}^{\text{LoS},n}$ is the deterministic LoS component which can be defined as follows

$$\bar{h}_{R,U}^{\text{LoS},n} = \underbrace{\left[1, e^{-j\frac{2\pi}{\lambda}\zeta\phi_{R,U}^{n}}, ..., e^{-j\frac{2\pi}{\lambda}(M-1)\zeta\phi_{R,U}^{n}}\right]^{T}}_{\text{array response}}, \forall n \in N, \tag{7.5}$$

where $\phi_{R,U}^{n} = \dfrac{x_R - x_U^{n}}{D_{R,U}^{n}}$ is cosine of the angle of arrival of signal from the RIS to UAV. ζ is the separation between RIS elements and λ is the carrier wavelength.

Similarly, we can compute the channel gain between the RIS and IoTDs which is denoted by $h_{R,i}^{n} \in \mathbb{C}^{M\times 1}$.

$$h_{R,i}^{n} = \underbrace{\sqrt{\rho (D_{R,i})^{-\alpha}}}_{\text{path loss}} \underbrace{\sqrt{\frac{K}{1+K}} \bar{h}_{R,i}^{n\ \text{LoS}}}_{\text{Rician fading}}, \forall i, n \in N, \tag{7.6}$$

$$\bar{h}_{R,i}^{n\ \text{LoS}} = \underbrace{\left[1, e^{-j\frac{2\pi}{\lambda}\zeta\phi_{R,i}}, ..., e^{-j\frac{2\pi}{\lambda}(M-1)\zeta\phi_{R,i}}\right]^{T}}_{\text{array response}}, \forall i, n \in N, \tag{7.7}$$

where $D_{R,i}$ is the euclidean distance between the RIS and IoTD i and $\phi_{R,i} = \dfrac{x_R - x_i}{d_{R,i}}$.

Denote the phase-shift matrix of the RIS in the nth time slot as $\Theta^{n} = \text{diag}\{e^{j\theta_1^{n}}, ..., e^{j\theta_M^{n}}\}$, where θ_m^{n} is the phase-shift of the mth reflecting element $m = 1, 2, , M$. Due to the hardware limitations, the phase-shift can only be selected from a finite set of discrete values. Specifically, the set of discrete values for each RIS reflecting element can be given as $\theta_m^{n} \in \Omega = \{0, \frac{2\pi}{Q}, \ldots, \frac{2\pi(Q-1)}{Q}\}$, where $Q = 2^{b}$ and b is the number of bits that control the number of available phase-shifts for the RIS elements. Hence, the SNR is:

$$\Gamma_i^{n} = \frac{P|h_{U\to i}^{n} + h_{R,U}^{n,H}\Theta^{n}h_{R,i}^{n}|^2}{\sigma^2}, \forall i, n \in N, \tag{7.8}$$

where P is the IoTD transmit power and H denotes the conjugate transpose operator. Next, we can compute the channel gain component as in Eq (7.9).

$$h^n_{U\to i} + h^{n,H}_{R,U}\Theta^n h^n_{R,i} = Pr(S^n_{U\to i} = \text{LoS})(D^n_{U\to i})^{-\beta_1} + \beta_2(D^n_{U\to i})^{-\beta_1} - Pr(S^n_{U\to i} = \text{LoS})\beta_2(D^n_{U\to i})^{-\beta_1} + \frac{\rho\frac{K}{K+1}}{\sqrt{(d_{R,i})^{\alpha}}\sqrt{(d^n_{R,U})^{\alpha}}} \times \sum_{m=1}^{M} e^{j(\theta^n_m + \frac{2\pi}{\lambda}(m-1)\zeta\phi_{R,i} - \frac{2\pi}{\lambda}(m-1)\zeta\phi^n_{R,U})}, \forall i, n \in N. \tag{7.9}$$

The amount of data collected from each IoTD at time slot n can then be computed as follows.

$$l^n_i = \delta^n_i \log_2(1+\Gamma^n_i), \forall i, n, \tag{7.10}$$

where δ^n_i is a scheduling decision variable (1 if device i is scheduled at time slot n and 0 otherwise).

7.2.3 Objective Function

We seek to maximize the number of served IoTs, each during its activation period subject to several operational constraints.

$$\max_{\Theta, x_U, y_U, \delta} \quad \sum_{i=1}^{I} \omega_i \tag{7.11a}$$

$$\text{s.t.} \quad \sum_{n=1}^{N} l^n_i \geq \omega_i Z_i, \tag{7.11b}$$

$$\theta^n_m \in \Omega, \forall n \in N, m \in M, \tag{7.11c}$$

$$\sqrt{(x^{n+1}_U - x^n_U)^2 + (y^{n+1}_U - y^n_U)^2} \leq V, \tag{7.11d}$$

$$0 \leq x^n_U \leq X, \tag{7.11e}$$

$$0 \leq y^n_U \leq Y, \tag{7.11f}$$

$$\sum_{i=1}^{I} \delta^n_i \leq C, \forall n \in N. \tag{7.11g}$$

$$\delta^n_i \leq n - T_i - 1, \forall i \in I. \tag{7.11h}$$

$$\delta_i^n \leq F_i - n, \forall i \in I. \tag{7.11i}$$

In Eq (7.11b), ω_i indicates whether a IoTD data has been collected successfully ($\omega_i = 1$) or not ($\omega_i = 0$), therefore, the objective aims at maximizing the number of IoTDs admitted for service. Eq (7.11c) ensures that phase shift are set within their possible range. Eq (7.11d) does not allow the UAV to violate its maximum speed. Eq (7.11e) and (7.11f) refrain the UAV from leaving the concerned area. Eq (7.11g) emphasizes that the number of IoTD scheduled to transmit at each time slot does not violate the available resources. Eq (7.11h) and (7.11i) make sure that the IoTDs are only scheduled to transmit their data during the active period.

The above problem is a mixed-integer non-convex owing to the existence of integer and binary decision variables of wireless scheduling and RIS discrete phase shift. Besides, the UAV trajectory is itself non-convex [77]. Hence, the resulted problem is difficult to solve or obtain its optimal solution in polynomial time. Further, some environment parameters such as IoTDs' active periods are not given in real-world. In this sense, dynamic programming and linear programming based solutions are not suitable to solve such problems. Therefore, we resort to DRL to tackle the problem where DRL has been used widely to effectively solve similar problems. First, we cast the original problem into two sub-problems. The first part associates with UAV trajectory and IoTDs scheduling. This part is formulated as MDP and then solved via PPO agent. The outputs of the first part is then fed to the second sub-problem to solve the RIS phase-shift configuration via iterative BCD. The complete solution will be discussed next.

7.3 Solution Approach

7.3.1 The UAV Mobility subproblem

First, the problem defined in the previous section is formulated as MDP (similar to Section 3.4.1) where the state, action, reward, are defined as follows:

- State S: The state at time slot n, $s^n \in S$, can be expressed by:

$$s^n = [x_U^n, y_U^n, X, Y, T^n, F^n, U^n, Z^n]. \tag{7.12}$$

Where the state $s^n \in S$ includes UAV position at time slot n (x_U^n, y_U^n). Two variables X, Y denote the IoTDs' locations. A vector T^n represents active period starting time for all the IoTDs at time slot n. A vector F^n represents active period deadline for all the IoTDs at time slot n. A vector U^n represents the total amounts of data fetched from the IoTDs by time slot n. A vector Z^n represents the amount of data to be uploaded from all the IoTDs at time slot n.

- Action A: The action taken at time slot n, $a^n \in A$ contains two sup-actions. First, a_U^n, is the UAV trajectory (x_U, y_U). Second, a_i^n, is the IoTD wireless scheduling. In order to handle the first part of the action space which is related to UAV mobility that is continuous for both distance and direction of movement (angle) in 2-D space, we discretize this part into 5 directions related to the different actions that represent all kinds of movements that the UAV can take (left, right, forward, backward, stop). The second part of the action space determines the IoTDs wireless scheduling. To do so, the agent has to determine which IoTDs will be scheduled to transmit in the current time slot, meaning, the agent has to select a subset of size C from I. However, not all the IoTDs can transmit at each time slot since some of them are in sleep mode or have already uploaded their data. The average number of IoTD that are active in one time slot can be driven from the activation pattern of the IoTDs, which is chosen to be uniform in this work, and can be expressed as $I\Lambda/N$ where Λ represents the average active period length of the IoTDs[1]. Eventually, the total number of actions available can be realised through:

$$5 \times \frac{\left(\frac{I\Lambda}{N}\right)!}{C!\left(\frac{I\Lambda}{N} - C\right)!}, \tag{7.13}$$

where 5 corresponds to the UAV trajectory and the second part represents all the combinations of IoTDs that can be scheduled to serve in one time slot.

[1] For the sake of simplicity, in this work we assume all IoTDs have the same active period length.

- Reward: The immediate reward, r^n, is equal to the number of IoTDs served if the data successfully uploaded to the UAV at time slot n. Otherwise, r^n is set to 0.

For DRL, we exploit PPO to develop our agent as laid out in Algorithm 15. First, the agent initializes random sampling policy and value function for the neural networks. Then, in each epoch, the agent observes the environment which consists of the set of IoTDs and their information, active periods, data uploaded, etc. Then at each time slot n, the agent decides where the UAV should go next. If the next movement is still within the area of interest, the UAV will move, otherwise, the UAV stays at its current location. This will make sure that the UAV does not leave the area and will help the agent to learn faster by having less states and actions to explore. Next, the agent selects an action which is a binary vector that determines which set of IoTDs will be served via the UAV. Based on that action, the BCD algorithm is then invoked to configure the phase-shift matrix in order to maximize the channel gain. Eventually, the time step reward is worked out which has two cases. First, if no IoTD has uploaded its value, the step reward will be set to 0. Otherwise, the step reward will be equal to the number of IoTDs that has managed to upload their data to the UAV completely at that time slot.

Algorithm 15: Proposed DRL for Scheduling

Inputs: N, v, Learning Rate, γ, ε.
Outputs: UAV trajectory.
Initialize policy π with random parameter θ and threshold ε
Initialize value function V with random parameters ϕ
for *each episode* $k \in \{0, 1, 2, ...\}$ **do**
    while $n < N$ **do**
        Observe state s^n.
        Choose action a^n.
        Move the UAV.
        if *UAV new location is not outside the area of interest* **then**
            Keep the UAV at its last location.
        Schedule the IoT devices based on a^n.
        Compute the reward r^n.
    Compute the advantage function.
    Optimize the clipped objective.
    $\theta_{old} \leftarrow \theta$.

7.3.2 A BCD Method for RIS Configuration

After the DRL agent has decided the next UAV move as well as the set of scheduled IoTDs at time slot n, Block Coordinate Descent (BCD) is then invoked to tune the RIS phase-shift coefficients. The objective of the BCD is to maximize the amount of data collected at each time slot by maximizing the achievable data rate through all the scheduled IoTDs which can be expressed as:

$$\max_{\Theta} \quad \sum_{i=1}^{I} \delta_i^n \log_2(1+\frac{P|h_{U\to i}^n + h_{R,U}^{n,H}\Theta^n h_{R,i}^n|^2}{\sigma^2}), n. \tag{7.14a}$$

$$\text{s.t.} \quad \theta_m^n \in \Omega, \forall n \in N, m \in M. \tag{7.14b}$$

The BCD algorithm works in an iterative way where a sequence of block optimization procedures are performed. That is, in each iteration, it optimizes one RIS element by checking all the possible values in Ω while fixing the other elements. The value that maximizes the objective in Eq (7.14a) will be chosen. After that, the next element will be selected to be optimized and so forth until all the RIS elements are optimized. This procedure repeats until there is no change in the RIS phase-shift configuration. The main details of the proposed BCD approach are presented in Algorithm 16. Regarding the complexity of Algorithm 16, it is $O(VM2^b)$ where V represents the number of iterations needed until Eq (7.14a) is maximized. Indeed, there are three loops embedded in Algorithm 16; the first loop corresponds to the number of required iterations, the second one is to iterate over each RIS element, and the last one denotes the number of quantization levels to control each element. For the sake of illustration, the complete solution approach is sketched in Fig. 7.2 where the two main components, PPO and BCD, are shown. After the PPO agent observes the state of the environment, it selects an action with the help of neural networks. The action is then processed to find an action for the UAV mobility and an action for wireless resource scheduling. The two sub-actions are then go in the BCD algorithm to tune the RIS phase shift. Eventually, the environment computes the step-reward resulted from that action and sends it back to the agent.

Algorithm 16: Configure The RIS Elements

Inputs: δ_i^n.
Outputs: θ^n
while *Eq (7.14a) not maximized* **do**
  for $m = 1, ..., M$ **do**
    Fix $\theta_{m'}^n, \forall m' \neq m, m' \in M$
    Set $\theta_m^n = \arg\max_{\Omega}$ Eq(7.14a)
  Obtain θ^n

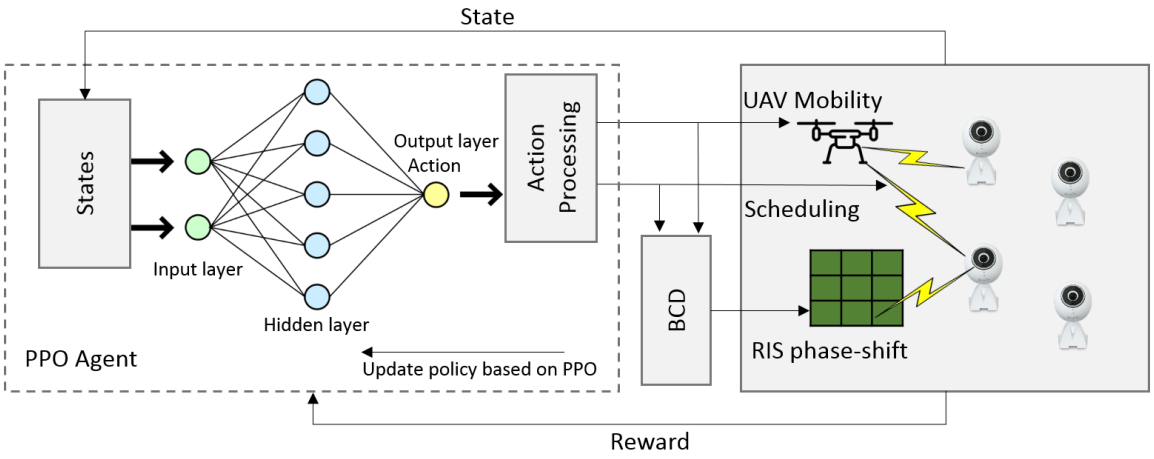


Figure 7.2: Solution architecture.

Table 7.2: Simulation Parameters

Parameter	Value
Area size	$300 \times 300\ m^2$
Number of IoTDs (I)	50 IoTDs
σ^2	-110 dBm
K	10 dB [123]
α	4
P	20 dBm
C	3
ρ	10 dBm
Z_i	50 bits
M	100
b	2 bits
z_U	50 m
z_R	1 m
z_i	1 m
η_1, η_2 (for dense urban area)	11.95, 0.136

7.4 Numerical Results

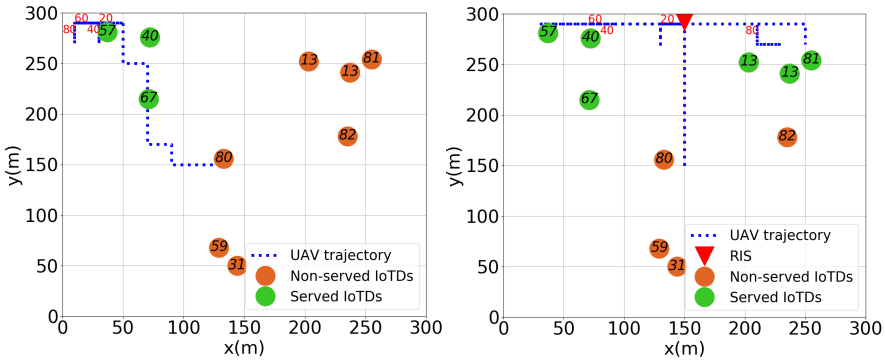
In this section, we present the simulation results to shed light on the performance of the RIS-assisted UAV in IoT wireless networks. For the simulation settings, we use PPO to build our agent, which is done using Python and TensorFlow. For the DRL, 3 linear layers are used with *tanh* as activation function for the middle layers and *softmax* for the output layer. Internal layers contain 64 units each and *Adam* optimizer is incorporated to minimize the loss function. The learning rate is set to 0.002, γ to 0.08, and clip to 0.02. For consistency and accuracy, the results were averaged over 500 data tests. The IoTDs are generated and placed based on a uniform distribution, and their active periods are uniformly distributed. The remaining parameters used in our study are listed in Table 7.2 (unless otherwise indicated).

We observe first the behaviour of the DRL agent; as illustrated in Fig. 7.3, the cumulative reward, which represents the number of IoTDs served, steadily increases as the agent is exposed to more epochs/iterations. We also observe that after around 500 iterations, the system starts to converge.

Figure 7.3: DRL agent convergence.

In addition, we show in Fig. 7.4 the UAV trajectory where two scenarios are considered; without RIS and with RIS ($M = 50$). The deadline of the activation periods are written inside the IoTD circle and the numbers in red denote the UAV location at that time slot. 10 IoTDs are considered in this network where each one generates data of size 60 bits. In both cases, DRL is used to control the UAV trajectory. The difference between the two scenarios in terms of mobility behaviour is apparent. In the scenario without RIS, Fig. 7.4 (a), the UAV flies towards a cluster of IoTDs that have their deadline in sequence and not far from each other. Recall that the UAV has limited resources and cannot collect all the generated data, hence, it is better to select a subset of IoTDs which can be served to completion. In this instance, IoTDs with deadlines 40, 57, and 67, would be the perfect choice as they locate near to each other and their deadlines do not overlap. The rest of IoTDs, such as the one near the center (deadline=80) and the two on the right side (deadline=81 and 82), will remain unserved as the UAV will not have enough time to fly and collect their data before expiration. On the other hand, in Fig. 7.4 (b), the UAV is able to serve 6 (twice more) devices with the assistance of the RIS. One can also observe that the behaviour of the agent is completely different. For example, first the UAV flies towards the RIS to improve the quality of the indirect LoS. Throughout this travel, the UAV is able to serve two IoTDs with deadline of 13 located on the right side. Then, it positions itself in the left side while balancing the channel quality of the direct and indirect LoS to serve the IoTDs with deadlines 40, 57, and 67. Finally, the UAV flies back to the other side in order to serve the IoTD of deadline 81. We can see that the IoTDs that are located far

from the RIS are not served since they have poor indirect LoS and the UAV has restricted maximum speed.



(a) (b)

Figure 7.4: UAV trajectory in two scenarios: (a) without RIS (b) with RIS ($M = 50$)

Next, in order to validate the performance of our solution approach, we develop four baseline methods to compare with. Indeed, to the best of our knowledge, no work in the literature addressed similar problem. These are explained as follows.

- Random Walk UAV: In this method, the UAV trajectory is determined based on random movement. On the other hand, the phase shift tuning is selected using the BCD method.
- Stationary UAV: the UAV is assumed to stay still in the middle of the network while the RIS phase shift is tuned using BCD.
- Random Θ^n: the RIS phase shift is configured following random distribution at each time slot while the UAV trajectory is planned using a DRL agent.
- Without RIS: In this method, the UAV has only direct link to communicate with the IoTDs. Again, in this scenario, the UAV is moved using a DRL agent.

In the rest of this section, we study the impact of various environment settings and parameters on the performance of the system and solution approach..

7.4.1 The impact of the number of RIS's elements

Fig. 7.5 shows the number of served devices (data collected within deadline) as we vary the size of the RIS (number of elements). We observe first, that indeed the size of the RIS helps improving

the performance and accordingly, more devices are served when more elements are present. One observation the figure makes is that when the phase shifts are randomly selected, not much gain can be seen from the RIS. However, as the RIS elements are optimized, the gain is apparent and varies with the way the UAV trajectory is optimized. Clearly, the best performance is attained when the two are jointly optimized, yielding close to $7\times$ gain when $M = 100$. Having a stationary UAV or a UAV with random trajectory have some gains, however the gain is much smaller than when the trajectory is determined by the developed DRL agent.

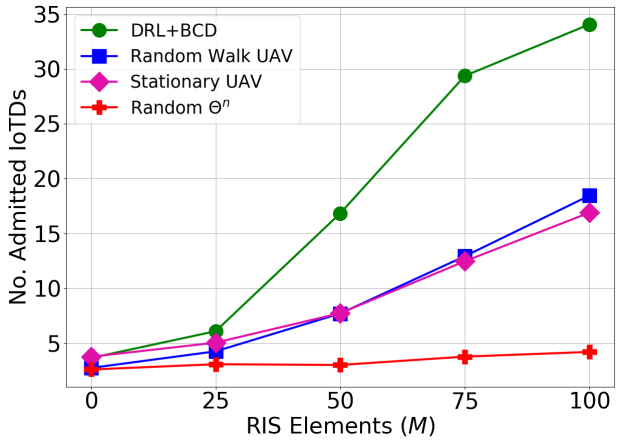


Figure 7.5: Effect of the RIS number of elements.

7.4.2 Effect of Network Size

In Fig. 7.6, we vary the number of IoTDs in the network and look at the number of served devices (percentage), under the four different methods. First, it can be seen that with smaller devices (10), the gain was relatively high, especially for our proposed solution. The difference between the proposed solution and the nearest one is greater than 50% all the way through. DRL+BCD manages to serve approximately 9 IoTDs when there are 10 IoTDs in the network. However, as the number of IoTDs increases, the service rate goes down. This is due to the limited wireless resources available for the UAV. Hence, when more IoTDs are added to the network, the ability of the UAV to fulfill the network needs will degrade. However, this issue can still be tackled by increasing the number of RIS elements[2]. In addition to our solution approach, the other baselines also show

[2]Fig. 7.5 shows that the number of RIS elements has a great impact on the number of IoTDs served. However, determining the size of the RIS is another problem and is beyond the scope of this chapter.

interesting behaviours. Intuitively, they all experience decreasing trends, however, Stationary UAV and Random Walk UAV face a sudden increase at 20 IoTDs before they start to gradually decrease. The reason behind such behaviour is that owing to the fixed wireless resources and the likelihood of IoTDs presence within the UAV vicinity, there are some resources wasted when there is little IoTDs near the UAV and RIS to serve. However, as the number of IoTDs increases, there will be more chances for the UAV to find IoTDs and serve them. Eventually, when there are more IoTDs in the network than the available resource, the percentage of served IoTDs starts to decrease and that what happens with networks of size 30, 40, and 50 IoTDs. The last two baselines, Random Θ^n and Without RIS, come in the last place with significantly poor performance, with marginal improvement for Random Θ^n, due to their weak RIS tuning or non availability of the RIS.

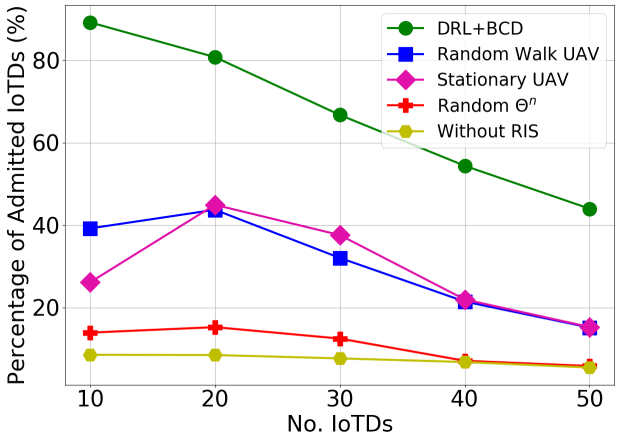


Figure 7.6: Effect of the number of IoTDs.

7.4.3 Effect of Data Size

In Fig. 7.7, we examine the impacts of having small and large size of data generated by the IoTDs. The percentage of IoTDs admitted for full service is shown versus the size of data in bits. In general, the small data generated, the better service that is provided. This is due to the fact that small data can be uploaded within shorter periods, subsequently, the UAV will have more free resources to serve other requests.

One can observe that DRL+BCD obtains the highest performance in comparison with the baselines in all scenarios. The difference between our solution approach and the other methods becomes

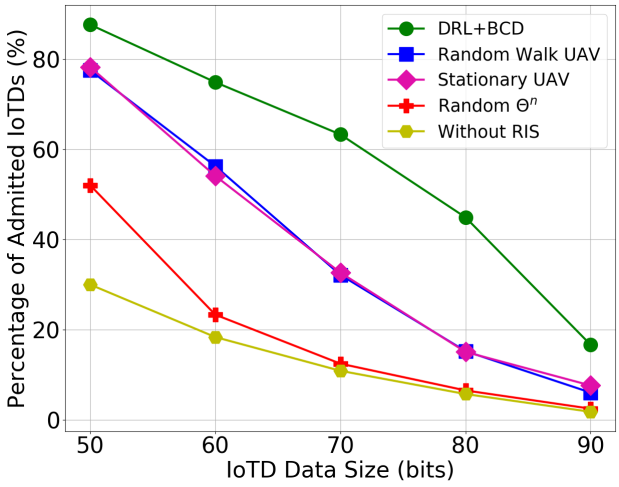


Figure 7.7: Effect of the data size generated by the IoTDs.

more apparent when the data size increases. That means, our solution approach can better adapt to more complex scenarios where the UAV and RIS resource are optimized wisely. Another insight one can look at is the impact of very large data (i.e., 90 bits). In such case, it becomes more difficult for the UAV to guarantee sufficient service for the IoTDs. Again, this issue can be tackled by investing more in the RIS size. Meanwhile, we can see a sharp decrease experienced by Random Walk UAV and Stationary UAV while both demonstrate better performance than Random Θ^n and without RIS. Interestingly, even with random phase-shift configuration, the RIS proves itself as a solid candidate to enhance the wireless link quality for the UAV when small size data is considered. At 50 bits, Random Θ^n is able to double the number of IoTDs admitted in comparison to the UAV alone scenario.

7.4.4 Effect of RIS on UAV Energy

One of the UAV limitation is the power capacity. Energy consumption of the UAV results from two sides, namely, UAV mobility and wireless communication. The first part, associated with UAV mobility, constitutes the dominant share of energy consumed from the UAV battery. Subsequently, in this section, we focus only on this part.

For this study, we use the metric of energy efficiency to quantify the benefits of RIS in aiding the UAV. In short, energy efficiency advocates for smaller energy consumption in regards to perform

higher. This can save device's battery life and improve energy consumption [152]. In the context of this work this translates to uploading more data while consuming less energy and can be expressed as.

$$\text{Energy Efficiency} = \frac{\sum_{i=1}^{I} s_i Z_i}{E(v)_{total}}, \tag{7.15}$$

where $E(v)_{total}$ represents the total energy consumption of the UAV throughout its operational time. The mobility of the UAV incurs energy and the amount consumed depends on the velocity of the UAV [70, 75] as follows.

$$P(v^n)_{total} = \underbrace{K\left(1+3\frac{M_{tip}^2}{w_b^2}\right)}_{\text{Blade profile power}} + \underbrace{\frac{1}{2}\pi(v^n)^3 F}_{\text{Parasite power}} + \underbrace{m_u g\sqrt{\left(\frac{-(v^n)^2+\sqrt{(v^n)^4+(\frac{m_U g}{\pi A})^2}}{2}\right)}}_{\text{Induced power}}. \tag{7.16}$$

where v^n denotes UAV velocity at time slot n and M_{tip} is the blade's rotor speed, K and F are two constants which depend on the dimensions of the blade and the UAV drag coefficient, respectively, π is the air density, m_U and g respectively denote the mass of the UAV and the standard gravity, A is the area of the UAV. The total energy consumption to cover a distance d at a constant velocity UAV w can be computed as $E(v)_{total} = \int_0^{d/v} P(v)dt = P(v)\frac{d}{v}$ as in [70].

In Fig. 7.8, the energy efficiency levels are obtained for various RIS sizes to understand the effect of having RIS on the UAV energy efficiency. It can be observed that without RIS, the energy consumed to upload one bit is very high. However, as the RIS size grows up, this value becomes smaller and smaller and that translates in better energy efficiency. In other words, the presence of RIS helps in two directions. First, it makes the UAV have more flexibility to plan its path such that the total amounts of energy consumed at the end is minimized as per Eq (7.16). Second, it helps to make transmission more successful by increasing the quality of the established links. This is demonstrated with RIS of 50 elements where the energy efficiency reaches above 15 bits/KJ. This significant increase keeps going up to increase by 30% with only 25 elements added to the RIS (at $M = 75$).

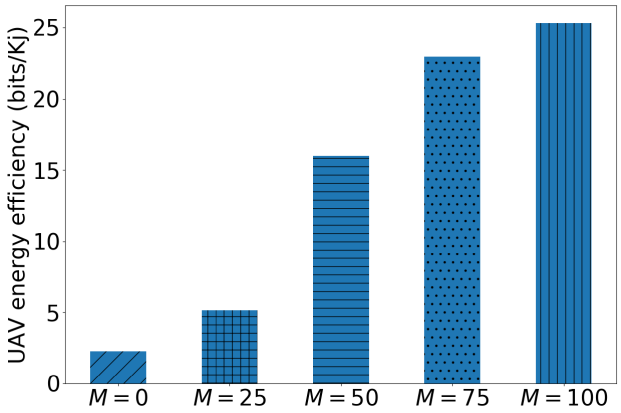


Figure 7.8: UAV energy efficiency levels with various RIS sizes.

7.5 Summary

In this work, we presented a new data gathering framework leveraging an RIS-UAV assisted network to collect data from sensing devices within their active times. The problem is formulated as mixed-integer non-convex programming and then, due to its complexity, is converted to MDP to be solved later via a DRL agent. The chapter also proposed a BCD method to handle the RIS phase shift configuration with lower complexity and high efficiency that was proved later in the numerical results by comparing it with random coefficient settings. The superiority of our solution approach is shown through simulations against four alternative methods. Indeed, the proposed solution outperformed its counterparts by more than 50% in many cases. Integrating RIS to the UAV communications has also shown another benefit of improving UAV energy efficiency remarkably especially when considering a large RIS.

Chapter 8

Conclusions and Future Research Directions

8.1 Conclusion

This thesis addressed several challenges associated with the deployment of wireless communications and digital services in vehicular networks. First, Chapter 1 of this thesis provided a comprehensive overview of the key advantages, potential applications, and communication challenges of vehicular networks to offer infotainment and safety-related applications. Then, limitations of existing studies and a summary of the research contributions of this thesis were highlighted.

In Chapter 2, a system model for RSU operators was presented aiming at offering improved QoE for the content providers' vehicular users. Three scenarios have been considered including V2I and V2V communications along with accounting for the similarities among contents and using pre-fetching techniques to reduce content delivery latency. The problem is formulated as an optimization problem and then small instances have been solved via CPLEX. In addition, alternatives, yet less complex solutions, have also been proposed. The simulation results showed that the provided solutions are significantly efficient.

Next, in Chapter 3, caching principles were utilized to further improve RSU efficiency. In this chapter, the scenario of disconnected RSU was considered. Based upon the fact that users share common interests in terms of contents, the RSU leverages the available contents on vehicles to fill

up its cache unit. Later, the RSU can start to serve vehicles which request cached contents. To this end, the popularity profiles of contents and their dynamics were taken into consideration to maximize the RSU cache hit rate. The problem was formulated mathematically and then converted to MDP to be solved via deep reinforcement learning.

The scenario presented in Chapter 3 was taken to another level in Chapter 4 by considering aerial RSU (UAV) instead of terrestrial one. In this chapter, a UAV equipped with cache capabilities to assist content delivery in vehicular environment was investigated. Based on the the restricted UAV battery capacity, the objective of this contribution was to improve the UAV energy efficiency. Then, the problem was formulated as MINLP to control UAV mobility and decide for the uplink and downlink scheduling. Next, deep reinforcement learning was employed to solve for the trajectory planning of the UAV and effective algorithms were proposed to determine wireless link establishments with the vehicles.

A more comprehensive scenario was given in Chapter 5 where the collaboration between terrestrial RSU and UAV is studied. In this regard, the RSU was assumed to be connected to the the backhaul link to fetch and serve contents to the vehicles. However, owing to the high demands and size of contents, it is sometimes difficult to serve vehicles through only one entity. Hence, the RSU can offload its demand partially to a nearby UAV. First, the RSU transfer some contents to the UAV via the vehicles. Later, the UAV starts to collaborate with the RSU by complete the content delivery of partially served vehicles. This problem was formulated as MDP and solved via dual task deep reinforcement learning to decide the UAV trajectory and amount to be served via the RSU.

Moreover, the employment of reflecting surfaces in vehicular communication was investigated in this thesis. Chapter 6 investigated the ability of RIS to establish indirect links between RSU and vehicles when the direct link is blocked due to the presence of blockage such as a high building. This chapter accounted for a more realistic scenario of RIS where practical discrete phase shifts were considered. Moreover, as opposed to the existing works, the mobility of vehicles with online phase shift tuning were addressed in this chapter. After mathematically formulating the problem, it was shown to be hard to solve, hence, BCD with deep reinforcement learning were leveraged to construct efficient and effective solutions for the wireless scheduling and RIS configuration. In addition, RIS was also proposed to aid UAV in serving IoT devices used to enable time-sensitive ITS

services in Chapter 7. The objective of this work was to maximize the number of IoTDs admitted to be served timely and completely. Again, the scenario turned out to be very complex along with some uncertainties in the environment such as IoTDs' activation times. As a result, DRL was leveraged to control UAV trajectory and wireless scheduling while the RIS phase-shift was solved via BCD. The solution approach of this work was compared with other baselines and demonstrated its effectiveness and superiority by doubling the achieved gains.

8.2 Future Work

8.2.1 Further Enhancement for Vehicular Networks

Although this dissertation covered several research challenges related to the deployment of caching, UAVs, and RIS in assisting vehicular networks, there is still room for improvement as follows.

In Chapter 2, the cost incurred by the backhaul between the RSUs and edge server can be studied along with the roles that cache can play to reduce this cost. In Chapter 3, we can extend this work by considering two kinds of sources to fetch the contents, namely, passing vehicles and backhaul links. In such scenario, the RSU may need to optimize its revenue by avoiding utilizing the backhaul link which incurred higher costs during peak hours and relay more on the passing vehicles. However, in some extreme scenarios, it may be more beneficial for the RSU operator to access the backhaul in case the content required is not available on the vehicles. A second direction can leverage V2V communication by pushing contents from the RSU to the vehicles such that vehicles can act as relay and caching nodes. In Chapter 4, the proposed idea can be further extended through considering V2V communication in order to enhance the performance of the system and improve the QoS to the vehicle users. Such scenario is of interest to network operator as the UAVs may belong to another operator and we cannot guarantee their collaboration. Here, the vehicles may need incentives to motivate them in delivering service for each others [153]. The mobility of UAV can also be improved through considering continuous action space which gives the UAVs more flexibility to planning their trajectory. In Chapter 5, the direct communication between the UAV and RSU can be studied in addition to relay contents through vehicles. In Chapter 6, we can

extend this work considering wireless resource allocation where the spectrum can be allocated to each vehicle based on its individual needs. As a result, the RIS phase-shift configuration method can also be updated to consider various link qualities for each specific vehicle depending on the allocated wireless resources. In Chapter 7 multi-UAVs can be further investigated to see how they cooperate with each other to plan their trajectory such that they avoid collisions and the ultimate objective is optimized. Moreover, the metric of Age of Information can also be applied to this context in order to ensure the freshness of the data collected from the IoTDs.

8.2.2 Federated Learning to Decentralize Training and Enforce Privacy

Over the last years, vehicular networks have been studied widely in order to set a foundation for ITS. That includes a large number of novel ITS use cases which have surfaced in the literature and the industry as well. As such, many new challenges and requirements arose that require quick and effective solutions. Among them, we recognize the issue of big data and data privacy. Such problem can be mitigated by using federated learning.

Despite that federated learning has been leveraged to improve wireless communications, the employment of this technique in vehicular networks is confined to limited attempts [154–156]. Federated learning is a method in machine learning that moves part of the training process to the entities and then collects the learning parameters only to construct a global model. Such mechanism reduces the burden on the central unit besides it contributes in saving sensitive data privacy. In the meantime, vehicular networks comprise a large set of entities including vehicles, RSU, BS, and UAVs. The data generated by such unit, especially for vehicles, is very critical and might reveal important information if compromised. Hence, federated learning can tackle such issues. In this particular section, we identify two directions of work. The first direction is of federated learning employment to calculate content popularity profiles at local and global scale. Here, each vehicle may learn and share its parameters with a central unit. Then RSU can then figure out the content popularity and cache those contents with higher demands. The second direction is related to vehicles trajectories which are private data that vehicles are reluctant to share by default with any external entity. However, using federated learning, vehicles can reveal some meaningful information to other units to make decisions without exposing their critical information.

Bibliography

[1] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, 2018.

[2] A. Ndikumana, N. H. Tran, and C. S. Hong, "Deep learning based caching for self-driving car in multi-access edge computing," *arXiv preprint arXiv:1810.01548*, 2018.

[3] J. Chen, W. Xu, N. Cheng, H. Wu, S. Zhang, and X. S. Shen, "Reinforcement learning policy for adaptive edge caching in heterogeneous vehicular network," in *Proc. IEEE Globecom*, 2018, pp. 1–6.

[4] U. Challita and W. Saad, "Network formation in the sky: Unmanned aerial vehicles for multi-hop wireless backhauling," in *Proc. IEEE Globecom*, 2017, pp. 1–6.

[5] P. V. Klaine, M. Jaber, R. D. Souza, and M. A. Imran, "Backhaul aware user-specific cell association using Q-learning," *IEEE Trans. Wireless Commun.*, 2019.

[6] E. Ahmed and H. Gharavi, "Cooperative vehicular networking: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 996–1014, 2018.

[7] W. Duan, J. Gu, M. Wen, G. Zhang, Y. Ji, and S. Mumtaz, "Emerging technologies for 5g-iov networks: Applications, trends and opportunities," *IEEE Network*, vol. 34, no. 5, pp. 283–289, 2020.

[8] R. She and Y. Ouyang, "Efficiency of uav-based last-mile delivery under congestion in low-altitude air," *Transportation Research Part C: Emerging Technologies*, vol. 122, p. 102878, 2021.

[9] M. Bansal, S. Gupta, and S. Mathur, "Comparison of ecc and rsa algorithm with dna encoding for iot security," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2021, pp. 1340–1343.

[10] R. F. Atallah, M. J. Khabbaz, and C. M. Assi, "Vehicular networking: A survey on spectrum access technologies and persisting challenges," *Vehicular Communications*, vol. 2, no. 3, pp. 125–149, 2015.

[11] A. Al-Hilo, D. Ebrahimi, S. Sharafeddine, and C. Assi, "Revenue-driven video delivery in vehicular networks with optimal resource scheduling," *Vehicular Communications*, vol. 23, p. 100215, 2020.

[12] X. Yuan, Y.-J. A. Zhang, Y. Shi, W. Yan, and H. Liu, "Reconfigurable-intelligent-surface empowered wireless communications: Challenges and opportunities," *IEEE Wireless Commun.*, 2021.

[13] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 22–28, 2017.

[14] S. Muthuramalingam, A. Bharathi, N. Gayathri, R. Sathiyaraj, B. Balamurugan *et al.*, "IoT based intelligent transportation system (IoT-ITS) for global perspective: A case study," in *Internet of Things and Big Data Analytics for Smart Generation*. Springer, 2019, pp. 279–300.

[15] T. A. Al-Janabi and H. S. Al-Raweshidy, "An energy efficient hybrid MAC protocol with dynamic sleep-based scheduling for high density IoT networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2273–2287, 2019.

[16] Y. Hui, Z. Su, T. H. Luan, and J. Cai, "Content in motion: An edge computing based relay scheme for content dissemination in urban vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3115–3128, 2018.

[17] H. T. Nguyen, H. D. Tuan, T. Q. Duong, H. V. Poor, and W.-J. Hwang, "Collaborative multi-cast beamforming for content delivery by cache-enabled ultra dense networks," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3396–3406, 2019.

[18] L. Liu, Y. Zhou, J. Yuan, W. Zhuang, and Y. Wang, "Economically optimal MS association for multimedia content delivery in cache-enabled heterogeneous cloud radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1584–1593, 2019.

[19] Z. Su, Y. Hui, and S. Guo, "D2D-based content delivery with parked vehicles in vehicular social networks," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 90–95, 2016.

[20] L. Zhang, S. Jia, Z. Liu, Y. Wang, and Y. Liu, "Bus-ads: Bus trajectory-based advertisement distribution in vanets using coalition formation games," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1259–1268, 2017.

[21] G. Einziger, C. F. Chiasserini, and F. Malandrino, "Scheduling advertisement delivery in vehicular networks," *IEEE Trans. Mobile Comput.*, 2018.

[22] V. Fux, P. Maillé, and M. Cesana, "Price competition between road side units operators in vehicular networks," in *Networking Conference, 2014 IFIP*. IEEE, 2014, pp. 1–9.

[23] Z. Su, Q. Xu, Y. Hui, M. Wen, and S. Guo, "A game theoretic approach to parked vehicle assisted content delivery in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6461–6474, 2017.

[24] M. Kadadha, H. Otrok, H. Barada, M. Al-Qutayri, and Y. Al-Hammadi, "A stackelberg game for street-centric QoS-OLSR protocol in urban vehicular Ad Hoc networks," *Vehicular Communications*, vol. 13, pp. 64–77, 2018.

[25] B. Wang, Z. Han, and K. R. Liu, "Distributed relay selection and power control for multiuser cooperative communication networks using stackelberg game," *IEEE Trans. Mobile Comput.*, vol. 8, no. 7, pp. 975–990, 2008.

[26] A. Darmann, U. Pferschy, and J. Schauer, "Resource allocation with time intervals," *Theoretical Computer Science*, vol. 411, no. 49, pp. 4217–4234, 2010.

[27] E. S. P. R. P. Roess and W. R. McShane, *Traffic Engineering 3rd ed.* Englewood Cliffs, NJ: Prentice-Hall, 2004.

[28] M. J. Khabbaz, "Modelling and delay analysis of intermittently connected roadside communication networks," Ph.D. dissertation, Concordia University, 2012.

[29] M. Naslcheraghi, M. Afshang, and H. S. Dhillon, "Modeling and performance analysis of full-duplex communications in cache-enabled D2D networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

[30] V. S. Varanasi and S. Chilukuri, "Adaptive differentiated edge caching with machine learning for V2X communication," in *Proc. 11th Int. Conf.Commun. Syst. Netw. (COMSNETS)*, 2019, pp. 481–484.

[31] D. Zhang, "Distributed cache enabled V2X networks: Proposals, research trends and challenging issues," *arXiv preprint arXiv:1803.06059*, 2018.

[32] C. Ma, M. Ding, H. Chen, Z. Lin, G. Mao, Y.-C. Liang, and B. Vucetic, "Socially aware caching strategy in device-to-device communication networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4615–4629, 2018.

[33] L. Hou, L. Lei, K. Zheng, and X. Wang, "A Q-Learning based proactive caching strategy for non-safety related services in vehicular networks," *IEEE Internet Things J.*, 2018.

[34] F. Chen, D. Zhang, J. Zhang, X. Wang, L. Chen, Y. Liu, and J. Liu, "Distribution-aware cache replication for cooperative road side units in VANETs," *Peer-to-Peer Networking and Applications*, vol. 11, no. 5, pp. 1075–1084, 2018.

[35] S. Fang, P. Fan, and Z. Khan, "Stochastic playback delay upper bounds of vehicular video content delivery networks with cache-enabled RSUs," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, 2018, pp. 1–5.

[36] R. Ding, T. Wang, L. Song, Z. Han, and J. Wu, "Roadside-unit caching in vehicular ad hoc networks for efficient popular content delivery," in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2015, pp. 1207–1212.

[37] Z. Hu, Z. Zheng, T. Wang, L. Song, and X. Li, "Roadside unit caching: Auction-based storage allocation for multiple content providers," *IEEE Trans. Commun.*, vol. 16, no. 10, pp. 6321–6334, 2017.

[38] Z. Su, Y. Hui, Q. Xu, T. Yang, J. Liu, and Y. Jia, "An edge caching scheme to distribute content in vehicular networks," *IEEE Trans. Veh. Technol.*, 2018.

[39] Z. Zhao, L. Guardalben, M. Karimzadeh, J. Silva, T. Braun, and S. Sargento, "Mobility prediction-assisted over-the-top edge prefetching for hierarchical VANETs," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1786–1801, 2018.

[40] L. Yao, A. Chen, J. Deng, J. Wang, and G. Wu, "A cooperative caching scheme based on mobility prediction in vehicular content centric networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5435–5444, 2018.

[41] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet Things J.*, 2019.

[42] 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. Technol.*, vol. 66, no. 11, pp. 10 433–10 445, 2017.

[43] J. Gu, W. Wang, A. Huang, H. Shan, and Z. Zhang, "Distributed cache replacement for caching-enable base stations in cellular networks," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 2648–2653.

[44] C. Zhong, M. C. Gursoy, and S. Velipasalar, "A deep reinforcement learning-based framework for content caching," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2018, pp. 1–6.

[45] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.

[46] M. Chen, W. Saad, and C. Yin, "Liquid state machine learning for resource and cache management in LTE-U unmanned aerial vehicle (UAV) networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1504–1517, 2019.

[47] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 180–190, 2017.

[48] L. Lu, Y. Jiang, M. Bennis, Z. Ding, F.-C. Zheng, and X. You, "Distributed edge caching via reinforcement learning in fog radio access networks," in *IEEE Veh. Tech. Conf.* IEEE, 2019, pp. 1–6.

[49] F. Jiang, K. Thilakarathna, M. A. Kaafar, F. Rosenbaum, and A. Seneviratne, "A spatio-temporal analysis of mobile internet traffic in public transportation systems: A view of web browsing from the bus," in *Proceedings of the 10th ACM MobiCom Workshop on Challenged Networks*. ACM, 2015, pp. 37–42.

[50] B. Hu, L. Fang, X. Cheng, and L. Yang, "In-vehicle caching (IV-cache) via dynamic distributed storage relay (D2 SR) in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 843–855, 2018.

[51] L. Sun, H. Shan, A. Huang, L. Cai, and H. He, "Channel allocation for adaptive video streaming in vehicular networks." *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 734–747, 2017.

[52] T. Deng, P. Fan, and D. Yuan, "Optimizing retention-aware caching in vehicular networks," *IEEE Trans. Commun.*, 2019.

[53] R. F. Atallah, C. M. Assi, and J. Y. Yu, "A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4592–4601, 2016.

[54] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, 2019.

[55] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, 2019.

[56] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.

[57] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine learning for resource management in cellular and iot networks: Potentials, current solutions, and open challenges," *IEEE Commun. Surveys Tuts.*, 2020.

[58] P. Sun, N. Aljeri, and A. Boukerche, "A queueing model-assisted traffic conditions estimation scheme for supporting vehicular edge computing," in *IEEE 30th Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2019, pp. 1–6.

[59] S. Otoum, B. Kantarci, and H. T. Mouftah, "On the feasibility of deep learning in sensor network intrusion detection," *IEEE Networking Letters*, vol. 1, no. 2, pp. 68–71, 2019.

[60] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *arXiv preprint arXiv:1811.12560*, 2018.

[61] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, 2018.

[62] H. Huang, A. V. Savkin, and C. Huang, "Reliable path planning for drone delivery using a stochastic time-dependent public transportation network," *IEEE Trans. Intell. Transp. Syst.*, 2020.

[63] H. Zhou, N. Cheng, J. Wang, J. Chen, Q. Yu, and X. Shen, "Toward dynamic link utilization for efficient vehicular edge content distribution," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8301–8313, 2019.

[64] N. Zhao, F. R. Yu, L. Fan, Y. Chen, J. Tang, A. Nallanathan, and V. C. Leung, "Caching unmanned aerial vehicle-enabled small-cell networks: Employing energy-efficient methods that store and retrieve popular content," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 71–79, 2019.

[65] H. Wu and et.al., "Joint caching and trajectory design for cache-enabled UAV in vehicular networks," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2019.

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

[67] B. Jiang, J. Yang, H. Xu, H. Song, and G. Zheng, "Multimedia data throughput maximization in Internet-of-Things system based on optimization of cache-enabled UAV," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3525–3532, 2018.

[68] S. Ortiz, C. T. Calafate, J.-C. Cano, P. Manzoni, and C. K. Toh, "A UAV-based content delivery architecture for rural areas and future smart cities," *IEEE Internet Comput.*, vol. 23, no. 1, pp. 29–36, 2018.

[69] M. Samir, M. Chraiti, C. Assi, and A. Ghrayeb, "Joint optimization of UAV trajectory and radio resource allocation for drive-thru vehicular networks," in *Proc. IEEE WCNC*, Apr. 2019, pp. 1–6.

[70] M. Samir, D. Ebrahimi, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Trajectory planning of multiple dronecells in vehicular networks: A reinforcement learning approach," *IEEE Networking Letters*, vol. 2, no. 1, pp. 14–18, 2020.

[71] M. Khabbaz, J. Antoun, S. Sharafeddine, and C. Assi, "Modelling and delay analysis of intermittent V2U communication in secluded areas," *IEEE Trans. Wireless Commun.*, 2020.

[72] L. Xiao, X. Lu, D. Xu, Y. Tang, L. Wang, and W. Zhuang, "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4087–4097, 2018.

[73] H. Wu, W. Xu, J. Chen, L. Wang, and X. Shen, "Matching-based content caching in heterogeneous vehicular networks," in *2018 IEEE Global Communications Conference*. IEEE, 2018, pp. 1–6.

[74] M. H. Cheung, F. Hou, V. W. Wong, and J. Huang, "Dora: Dynamic optimal random access for vehicle-to-roadside communications," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 4, pp. 792–803, 2012.

[75] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, 2019.

[76] Y. Cai, Z. Wei, R. Li, D. W. K. Ng, and J. Yuan, "Energy-efficient resource allocation for secure UAV communication systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr 2019, pp. 1–8.

[77] M. Samir, S. Sharafeddine, C. Assi, T. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Trans. Wireless Commun.*, 2019.

[78] M. Zhu, X.-Y. Liu, and X. Wang, "Deep reinforcement learning for unmanned aerial vehicle-assisted vehicular networks," *arXiv preprint arXiv:1906.05015*, 2019.

[79] J. Schulman and et.al., "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[80] E. Bøhn, E. M. Coates, S. Moe, and T. A. Johansen, "Deep reinforcement learning attitude control of fixed-wing UAVs using proximal policy optimization," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, 2019, pp. 523–533.

[81] L. Sun, H. Shan, A. Huang, L. Cai, and H. He, "Channel allocation for adaptive video streaming in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 734–747, 2016.

[82] M. J. Khabbaz, W. F. Fawaz, and C. M. Assi, "A simple free-flow traffic model for vehicular intermittently connected networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1312–1326, 2012.

[83] Q. Zheng, K. Zheng, H. Zhang, and V. C. Leung, "Delay-optimal virtualized radio resource scheduling in software-defined vehicular networks via stochastic learning," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7857–7867, 2016.

[84] Y. K. Tun and et.al., "Joint radio resource allocation and content caching in heterogeneous virtualized wireless networks," *IEEE Access*, 2020.

[85] Z. Ning, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, B. Sadoun, and R. Y. Kwok, "Joint computing and caching in 5G-envisioned internet of vehicles: A deep reinforcement learning-based traffic control system," *IEEE Trans. Intell. Transp. Syst.*, 2020.

[86] A. Al-Hilo *et al.*, "UAV-assisted content delivery in intelligent transportation systems-joint trajectory planning and cache management," *IEEE Trans. Intell. Transp. Syst.*, 2020.

[87] M. Najafi *et al.*, "Statistical modeling of the FSO fronthaul channel for UAV-based communications," *arXiv preprint arXiv:1905.12424*, 2019.

[88] C.-J. Hoel *et al.*, "Combining planning and deep reinforcement learning in tactical decision making for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 294–305, 2019.

[89] M. C. Lucic *et al.*, "A generalized dynamic planning framework for green UAV-assisted intelligent transportation system infrastructure," *IEEE Syst. J.*, 2020.

[90] J. Chen *et al.*, "Cooperative edge caching with location-based and popular contents for vehicular networks," *IEEE Trans. Veh. Technol.*, 2020.

[91] R. Zhang *et al.*, "UAV-aided data dissemination protocol with dynamic trajectory scheduling in vanets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.

[92] R. Lu *et al.*, "UAV-assisted data dissemination with proactive caching and file sharing in V2X networks," in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.

[93] F. Zeng *et al.*, "UAV-assisted data dissemination scheduling in vanets," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[94] W. Shi *et al.*, "Drone assisted vehicular networks: Architecture, challenges and opportunities," *IEEE Network*, vol. 32, no. 3, pp. 130–137, 2018.

[95] Y. Hui *et al.*, "Collaborative content delivery in software-defined heterogeneous vehicular networks," *IEEE/ACM Trans. Netw.*, 2020.

[96] S. Zhang *et al.*, "Air-ground integrated vehicular network slicing with content pushing and caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2114–2127, 2018.

[97] P. Wang *et al.*, "Cellular V2X communications in unlicensed spectrum: Harmonious coexistence with VANET in 5G systems," *IEEE Wireless Commun.*, vol. 17, no. 8, pp. 5212–5224, 2018.

[98] W. S. Atoui *et al.*, "Offline and online scheduling algorithms for energy harvesting RSUs in VANETs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6370–6382, 2018.

[99] M. Mozaffari *et al.*, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3949–3963, 2016.

[100] L. Yang *et al.*, "Beam tracking and optimization for UAV communications," *IEEE Wireless Commun.*, vol. 18, no. 11, pp. 5367–5379, 2019.

[101] M. Shokry *et al.*, "Age of information aware trajectory planning of UAVs in intelligent transportation systems: A deep learning approach," *IEEE Trans. Veh. Technol.*, 2020.

[102] Z. Qiu *et al.*, "Low altitude UAV air-to-ground channel measurement and modeling in semi-urban environments," *Wireless Commun. Mobile Comput.*, vol. 2017, 2017.

[103] Q. Wu and R. Zhang, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 106–112, 2019.

[104] M. Elhattab, M. A. Arfaoui, C. Assi, and A. Ghrayeb, "Reconfigurable intelligent surface assisted coordinated multipoint in downlink NOMA networks," *IEEE Commun. Lett.*, pp. 1–1, 2020.

[105] B. Di, H. Zhang, L. Song, Y. Li, Z. Han, and H. V. Poor, "Hybrid beamforming for reconfigurable intelligent surface based multi-user communications: Achievable rates with limited discrete phase shifts," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1809–1822, 2020.

[106] H. Zhang, B. Di, L. Song, and Z. Han, "Reconfigurable intelligent surfaces assisted communications with limited phase shifts: How many phase shifts are enough?" *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4498–4502, 2020.

[107] Q. Wu and R. Zhang, "Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1838–1851, 2019.

[108] H. Alwazani, A. Kammoun, A. Chaaban, M. Debbah, M.-S. Alouini *et al.*, "Intelligent reflecting surface-assisted multi-user MISO communication: Channel estimation and beamforming design," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 661–680, 2020.

[109] Y. Chen, Y. Wang, J. Zhang, and Z. Li, "Resource allocation for intelligent reflecting surface aided vehicular communications," *IEEE Trans. Veh. Technol.*, 2020.

[110] J. Wang, W. Zhang, X. Bao, T. Song, and C. Pan, "Outage analysis for intelligent reflecting surface assisted vehicular communication networks," *arXiv preprint arXiv:2004.08063*, 2020.

[111] D. Dampahalage, K. Manosha, N. Rajatheva *et al.*, "Intelligent reflecting surface aided vehicular communications," *arXiv preprint arXiv:2011.03071*, 2020.

[112] C. You, B. Zheng, and R. Zhang, "Channel estimation and passive beamforming for intelligent reflecting surface: Discrete phase shift and progressive refinement," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2604–2620, 2020.

[113] P. Xu, G. Chen, Z. Yang, and M. Di Renzo, "Reconfigurable intelligent surfaces assisted communications with discrete phase shifts: How many quantization levels are required to achieve full diversity?" *IEEE Wireless Commun. Lett.*, 2020.

[114] B. Di, H. Zhang, L. Li, L. Song, Y. Li, and Z. Han, "Practical hybrid beamforming with finite-resolution phase shifters for reconfigurable intelligent surface based multi-user communications," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4565–4570, 2020.

[115] W. Tang, J. Y. Dai, M. Z. Chen, K.-K. Wong, X. Li, X. Zhao, S. Jin, Q. Cheng, and T. J. Cui, "MIMO transmission through reconfigurable intelligent surface: System design, analysis, and implementation," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2683–2699, 2020.

[116] J. Yuan, Y.-C. Liang, J. Joung, G. Feng, and E. G. Larsson, "Intelligent reflecting surface-assisted cognitive radio system," *IEEE Trans. Commun.*, 2020.

[117] W. Yan, X. Yuan, Z.-Q. He, and X. Kuai, "Passive beamforming and information transfer design for reconfigurable intelligent surfaces aided multiuser MIMO systems," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1793–1808, 2020.

[118] X. Hu, C. Zhong, Y. Zhang, X. Chen, and Z. Zhang, "Location information aided multiple intelligent reflecting surface systems," *IEEE Trans. Commun.*, 2020.

[119] Y. Ai, L. Kong, M. Cheffena, S. Chatzinotas, and B. Ottersten, "Secure vehicular communications through reconfigurable intelligent surfaces," *arXiv preprint arXiv:2011.14899*, 2020.

[120] N. Mensi, D. B. Rawat, and E. Balti, "Physical layer security for V2I communications: Reflecting surfaces vs. relaying," *arXiv preprint arXiv:2010.07216*, 2020.

[121] H. Long, M. Chen, Z. Yang, B. Wang, Z. Li, X. Yun, and M. Shikh-Bahaei, "Reflections in the sky: Joint trajectory and passive beamforming design for secure UAV networks with reconfigurable intelligent surface," *arXiv preprint arXiv:2005.10559*, 2020.

[122] Q. Wang, S. Leng, H. Fu, and Y. Zhang, "An IEEE 802.11 p-based multichannel MAC scheme with channel coordination for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 449–458, 2011.

[123] Z. Abdullah, G. Chen, S. Lambotharan, and J. A. Chambers, "A hybrid relay and intelligent reflecting surface network and its ergodic performance analysis," *IEEE Wireless Commun. Lett.*, vol. 9, no. 10, pp. 1653–1657, 2020.

[124] M. Samir, M. Elhattab, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Optimizing age of information through aerial reconfigurable intelligent surfaces: A deep reinforcement learning approach," *arXiv preprint arXiv:2011.04817*, 2020.

[125] G. Lee, M. Jung, A. T. Z. Kasgari, W. Saad, and M. Bennis, "Deep reinforcement learning for energy-efficient networking with reconfigurable intelligent surfaces," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[126] Q. Zhang, W. Saad, and M. Bennis, "Distributional reinforcement learning for mmWave communications with intelligent reflectors on a uav," *arXiv preprint arXiv:2011.01840*, 2020.

[127] T. Bai, C. Pan, Y. Deng, M. Elkashlan, A. Nallanathan, and L. Hanzo, "Latency minimization for intelligent reflecting surface aided mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2666–2682, 2020.

[128] J. He, K. Yu, Y. Shi, Y. Zhou, W. Chen, and K. B. Letaief, "Reconfigurable intelligent surface assisted massive MIMO with antenna selection," *arXiv preprint arXiv:2009.07546*, 2020.

[129] S. Abeywickrama, R. Zhang, Q. Wu, and C. Yuen, "Intelligent reflecting surface: Practical phase shift model and beamforming optimization," *arXiv preprint arXiv:2002.10112*, 2020.

[130] S. Li, B. Duo, X. Yuan, Y.-C. Liang, and M. Di Renzo, "Reconfigurable intelligent surface assisted UAV communication: Joint trajectory design and passive beamforming," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 716–720, 2020.

[131] T. E. Humphreys, M. J. Murrian, and L. Narula, "Deep-urban unaided precise global navigation satellite system vehicle positioning," *IEEE Intell. Transp. Syst. Mag.*, vol. 12, no. 3, pp. 109–122, 2020.

[132] A. B. Sediq, R. H. Gohary, R. Schoenen, and H. Yanikomeroglu, "Optimal tradeoff between sum-rate efficiency and jain's fairness index in resource allocation," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3496–3509, 2013.

[133] Y. Cai, Z. Wei, R. Li, D. W. K. Ng, and J. Yuan, "Joint trajectory and resource allocation design for energy-efficient secure UAV communication systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4536–4553, 2020.

[134] Y. Lin, M. Wang, X. Zhou, G. Ding, and S. Mao, "Dynamic spectrum interaction of UAV flight formation communication with priority: A deep reinforcement learning approach," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 892–903, 2020.

[135] H. Shiri, J. Park, and M. Bennis, "Communication-efficient massive UAV online path control: Federated learning meets mean-field game theory," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6840–6857, 2020.

[136] M. Samir, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Online altitude control and scheduling policy for minimizing AoI in UAV-assisted IoT wireless networks," *IEEE Trans. Mobile Comput.*, 2020.

[137] R. Islambouli and S. Sharafeddine, "Optimized 3D deployment of UAV-mounted cloudlets to support latency-sensitive services in iot networks," *IEEE Access*, vol. 7, pp. 172 860–172 870, 2019.

[138] C. Zhan and Y. Zeng, "Energy-efficient data uploading for cellular-connected uav systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7279–7292, 2020.

[139] D. Callegaro and M. Levorato, "Optimal edge computing for infrastructure-assisted UAV systems," *IEEE Trans. Veh. Technol.*, 2021.

[140] Y. Liu, K. Liu, J. Han, L. Zhu, Z. Xiao, and X.-G. Xia, "Resource allocation and 3D placement for uav-enabled energy-efficient IoT communications," *IEEE Internet Things J.*, 2020.

[141] Y. Hu, X. Yuan, J. Xu, and A. Schmeink, "Optimal 1D trajectory design for UAV-enabled multiuser wireless power transfer," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5674–5688, 2019.

[142] L. Yang, F. Meng, J. Zhang, M. O. Hasna, and M. Di Renzo, "On the performance of RIS-assisted dual-hop UAV communication systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10 385–10 390, 2020.

[143] C. You, Z. Kang, Y. Zeng, and R. Zhang, "Enabling smart reflection in integrated air-ground wireless network: IRS meets UAV," *arXiv preprint arXiv:2103.07151*, 2021.

[144] X. Liu, Y. Liu, and Y. Chen, "Machine learning empowered trajectory and passive beamforming design in UAV-RIS wireless networks," *IEEE J. Sel. Areas Commun.*, 2020.

[145] Z. Wei, Y. Cai, Z. Sun, D. W. K. Ng, J. Yuan, M. Zhou, and L. Sun, "Sum-rate maximization for IRS-assisted UAV OFDMA communication systems," *IEEE Trans. Wireless Commun.*, 2020.

[146] T. Shafique, H. Tabassum, and E. Hossain, "Optimization of wireless relaying with flexible UAV-borne reflecting surfaces," *IEEE Trans. Commun.*, 2020.

[147] Y. Pan, K. Wang, C. Pan, H. Zhu, and J. Wang, "UAV-assisted and intelligent reflecting surfaces-supported terahertz communications," *arXiv preprint arXiv:2010.14223*, 2020.

[148] N. Kouzayha, Z. Dawy, J. G. Andrews, and H. ElSawy, "Joint downlink/uplink RF wake-up solution for IoT over cellular networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1574–1588, 2017.

[149] K. H. Kim and H.-D. Kim, "Deep sleep mode based nodemcu-enabled humidity sensor nodes monitoring for low-power iot," *Transactions on Electrical and Electronic Materials*, vol. 21, no. 6, pp. 617–620, 2020.

[150] X. Wang, X. Chen, Z. Li, and Y. Chen, "Access delay analysis and optimization of NB-IoT based on stochastic network calculus," in *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*. IEEE, 2018, pp. 23–28.

[151] A. Kozłowski and J. Sosnowski, "Energy efficiency trade-off between duty-cycling and wake-up radio techniques in IoT networks," *Wireless Personal Communications*, vol. 107, no. 4, pp. 1951–1971, 2019.

[152] D. J. Dechene and A. Shami, "Energy-aware resource allocation strategies for LTE uplink with synchronous HARQ constraints," *IEEE Trans. Mobile Comput.*, vol. 13, no. 2, pp. 422–433, 2012.

[153] O. A. Wahab, H. Otrok, and A. Mourad, "A dempster–shafer based tit-for-tat strategy to regulate the cooperation in vanet using qos-olsr protocol," *Wireless personal communications*, 2014.

[154] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Federated learning for ultra-reliable low-latency V2V communications," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[155] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy demand prediction with federated learning for electric vehicle networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[156] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4734–4746, 2020.