Intelligent Vehicle-to-Vehicle Communications with Importance of Fairness and Information Freshness

by

Pronab Ghosh

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE FACULTY OF GRADUATE STUDIES
OF LAKEHEAD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE WITH A SPECIALIZATION IN ARTIFICIAL INTELLIGENCE (AI)

2023
Lakehead University
Thunder Bay, Ontario, Canada

Examining Committee Membership

The thesis of Pronab Ghosh, titled Intelligent Vehicle-to-Vehicle Communications with Importance of Fairness and Information Freshness, is approved:

Supervisor:

Dr. Dariush Ebrahimi
Professor, Department of Physics & Computer Science
Wilfrid Laurier University, Waterloo, Ontario, Canada

Co-supervisor:

Dr. Thiago Eustaquio Alves de Oliveira
Professor, Department of Computer Science
Lakehead University, Orillia, Ontario, Canada

Internal Examiner:

Dr. Xing Tan
Professor, Department of Computer Science
Lakehead University, Thunder Bay, Ontario, Canada

External Examiner:

Dr. Salama Ikki, P.Eng.
Professor, Department of Electrical Engineering
Lakehead University, Thunder Bay, Ontario, Canada

ABSTRACT

Intelligent Transportation Systems (ITS) showcase cutting-edge services designed to revolutionize transportation and mobility, especially within future smart cities. These services play a pivotal role in bolstering traffic safety, traffic flow management, infotainment, and the dependability of edge-assisted autonomous driving. Consequently, ITS introduces the Vehicle-to-Vehicle (V2V) communication paradigm, facilitating continuous connectivity between moving vehicles and their surroundings. Real-time data exchange regarding acceleration, position, speed, and braking status enables collision avoidance and congestion mitigation. V2V communication streamlines communication pathways, resulting in safer and more comfortable driving experiences, particularly in high-risk scenarios. This thesis investigates two distinct challenges within V2V communications:

1. Multi-Group V2V Communications: This study addresses the establishment and scheduling of data streams and packets between vehicles within a multi-group communication setup. In scenarios involving police cars, ambulances, buses, or city fleets, each group of vehicles communicates within itself. The objective is to establish communication links between all vehicle pairs within a group, utilizing WiFi technology to alleviate the load on cellular networks. Since not all pairs have direct communication capabilities, the problem extends to relaying and scheduling data packets through multi-hop transmissions. Resource blocks, including designated channels and time slots, are allocated. The study aims to maximize communication efficiency among vehicle groups while ensuring fairness and allowing resource block reuse under the SINR constraint.

2. Age of Information (AoI) Minimization: Traditional metrics like throughput and latency do not sufficiently capture data stream timeliness and freshness, critical for autonomous driving and accident prevention. This study targets the minimization of AoI across all data streams in autonomous vehicular networks. The goal is to reduce the total or average AoI over a specified timeframe. Unlike the first study, direct data stream connections between vehicle pairs are absent. Instead, a vehicle broadcasts data to nearby vehicles based on data importance. Minimizing AoI requires optimizing relaying decisions, transmission timing, and data packet dropping. Complexity arises from optimizing nodes for data relaying, transmission timing, and prioritizing newer data packets.

In both studies, mathematical formulations employing mixed-integer linear programming (MILP) are initially employed for optimal solutions. Due to optimization model complexity, scalable heuristic methods are proposed for larger networks. To capture dynamic environmental dynamics, both problems are modeled as Markov Decision Processes

(MDP) and tackled using reinforcement learning (RL) techniques such as Qlearning and Double Deep Q-Networks (DDQN).

Additionally, hybrid heuristic-based RL methods are introduced to enhance learning behavior and overall performance. Numerical results underscore the efficacy of hybrid approaches in comparison to optimal solutions, random agents, proposed heuristics, and conventional RL methods across networks of varying sizes.

In conclusion, this thesis contributes to intelligent transportation systems and future smart cities by offering innovative solutions for vehicular communications. These approaches hold the potential to enhance data transmission efficiency and reliability for autonomous vehicles, paving the way for safer and more responsive autonomous driving experiences.

ACKNOWLEDGEMENTS

This thesis work represents the culmination of two years of dedicated effort and I am grateful for the support of numerous individuals, including my Lord Sree Krishna, who made this achievement possible. I would like to express my heartfelt appreciation to my supervisors, Dr. Dariush Ebrahimi and Dr. Thiago E. Alves de Oliveira, for providing me with the opportunity to pursue research under their invaluable guidance. I am grateful for their unwavering support, both morally and financially, and for guidance throughout my research journey over the past two years. To be honest, getting such friendly, gentle, organized, down-to-earth, and helpful-minded professors is one of the best parts of my life.

I am immensely thankful to Lakehead University for granting me access to the resources at the ATAC building, which were instrumental in carrying out this work. I would also like to extend my heartfelt gratitude to my colleagues at my lab. The discussions, and unwavering support from each of you truly boosted my morale and made the research experience more enjoyable. Besides, I would like to acknowledge the financial support provided by NSERC Discovery Grants and NSERC Research Development Fund, which funded the research topic under my supervisor's grant.

To my family, especially my beloved parents and sisters, I am profoundly grateful for their unending support and guidance. Despite being thousands of kilometers away from home, they have always stood by my side and encouraged me throughout my master's journey.

Also, I want to express my thanks and appreciation to all my friends, family, colleagues, teachers, professors, and well-wishers who have supported me in my educational pursuits. Your encouragement and belief in my abilities have been instrumental in my success.

Last not but least, I am truly fortunate to be received such tremendous support from everyone, and I am sincerely grateful for each and every individual who has played a part in shaping my educational and research endeavors.

PUBLICATIONS

- "Maximizing Communications and Fairness Within Groups of Vehicles: The Hybrid Heuristic-based Reinforcement Learning Framework," manuscript submitted to IEEE Transactions on Intelligent Transportation Systems. (Under Review)
- "Maximizing Communications within Groups of Vehicles While Maintaining Fairness," manuscript submitted to IEEE Transactions on Intelligent Transportation Systems. (Under Review)
- "MCFGV: Maximizing Communications and Fairness for Groups of Vehicles," manuscript submitted to IEEE PIMRC 2023: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. (Accepted)
- "Optimizing Information Freshness for Autonomous Vehicular Communication: A Hybrid Reinforcement Learning Strategy," manuscript submitted to IEEE INFOCOM 2024: IEEE International Conference on Computer Communications. (Under Review)
- "Maximizing Group-Based Vehicle Communications and Fairness: The Reinforcement Learning Approach," manuscript prepared and aimed at submitting to IEEE ICC 2024: IEEE International Conference on Communications(ICC).
- "Optimizing Freshness of Data Broadcasting for Autonomous Vehicular Communication: A Hybrid Reinforcement Learning Strategy," manuscript prepared and aimed at submitting to IEEE Internet of Things Journal.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the persistent demand for modern transportation, Intelligent Transportation Systems (ITS) deliver a wide range of state-of-the-art services that mainly focus on drastically enhancing transportation and mobility on roadways for the future generation [1]. These cutting-edge services not only expand traffic safety, flow control, and infotainment but also ensure the reliability, and confidentiality of edge-assisted autonomous driving [2–4].

However, to enable seamless communication between vehicles and dynamically improve overall traffic efficiency, ITS introduces a fully connected component known as a vehicular communication network, which enables vehicle-to-vehicle communication and creates a dynamic and interconnected ecosystem. Then, with the help of this dynamic ecosystem, V2V communication allows moving vehicles to stay online and connected to their surroundings by using wireless fidelity technologies (WiFi), known as Dedicated Short Range Communications (DSRC) [5,6].

While communicating with nearby vehicles considering WiFi protocols, V2V allows vehicles to share a stream of data packets based on three distinct types of communication methods including unicast, multicast, and broadcast, as shown in Fig. 1.1. In that sense, V2V communication also provides several real-time decision-making information on roads such as autonomous driving, road surveillance, source and destination locations, dangerous situation detection, real-time data sharing (vehicle's acceleration, position, speed, braking status, and heading), information about driving behaviors, communication services between neighboring vehicles, and many more [7]. These shared activities result in more efficient, safer, and comfortable driving experiences, especially in high-risk scenarios (i.e., blind spots, highway merging, and intersections), and create new opportunities in various business sectors [8–12]. Hence, the networking industry and academia have shown a deep interest in developing V2V communications and leveraging relevant services.

In the unicast method, if source-destination pairs are in the communication range, this approach can communicate directly through WiFi to send data packets from a specific

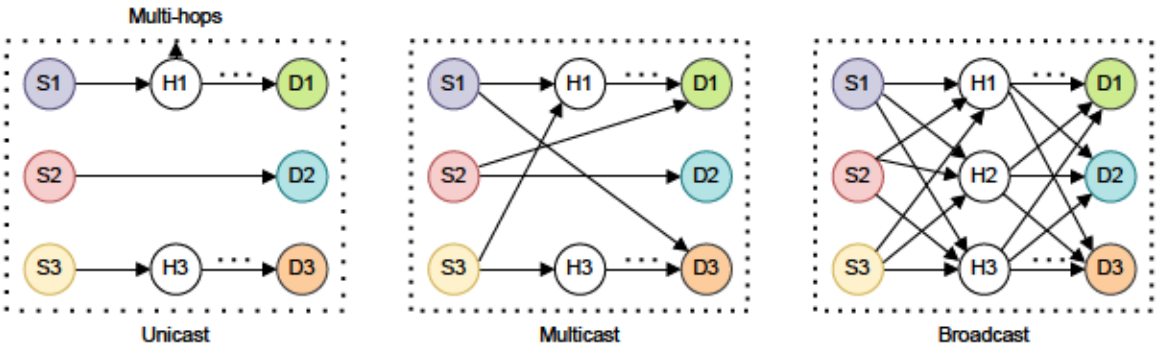


Figure 1.1: An illustration of Unicast, Multicast and Broadcast communication methods where S1, S2, S3 indicate source nodes, D1, D2, D3 indicate destination nodes, and H1, H2, H3 denote relay nodes between source and destination nodes.

source node to a specific destination node and establish a point-to-point connection between them [13,14]. Unlike the first method, the multicast method is mainly responsible for point-to-multipoint communications in which a sender transmits data packets to multiple receivers belonging to a specific set of groups [15,16]. On the other hand, the broadcast method sends packets from a source node to all nearby nodes within the communication range [14,17]. Hence, the unicast and multicast methods are one-to-one and one-to-many communication methods respectively, while the broadcast method is a one-to-all communication method that disseminates information to multiple destination nodes simultaneously. However, in all scenarios of communication methods, not all the source-destination pairs have a direct link because of a communication range (which can be defined by transmission power, distance, and fading). Hence, for some, we need to create a path by relaying transmissions over multi-hop nodes.

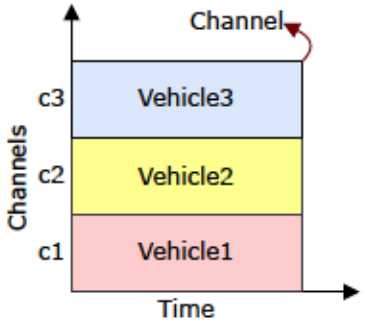


Figure 1.2: An illustration of frequency division multiple access (FDMA) where an FDMA media is divided into three equal orthogonal channels (i.e., c1, c2, and c3). Here, each channel contains a single vehicle (i.e., c1 contains a vehicle1) and we consider the length of the time frame to be 1.

We consider a frequency division multiple access (FDMA) media [18] that is applied to share the available frequency spectrum into different frequency bands or channels (see Fig. 1.2). Then, each resource or vehicle is allocated to a specific channel for transmissions. During the resource allocation phase, the resources might face some factors such as interference and traffic conditions, particularly when two vehicles are in close proximity. Hence, the channel allocation mechanism plays a crucial role in preventing these causing factors, trying to allocate all available channels one after another.

We also consider a time division multiple access (TDMA) media that is used for sharing the same frequency band or channel into different time slots. Here, each time slot is assigned to a specific vehicle. Also, the TDMA mechanism supports multiple channels (i.e., 3)

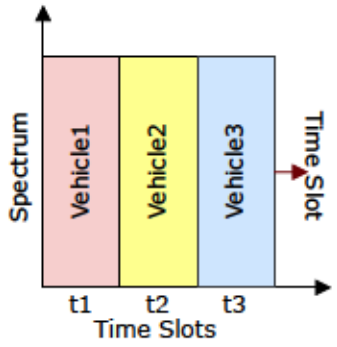


Figure 1.3: An illustration of time division multiple access (TDMA) where a TDMA mechanism is divided into three equal time slots (i.e., t1, t2, and t3). Here, each time slot contains a single vehicle (i.e., t2 contains a vehicle2) and we refer the entire bandwidth to a single channel (c).

as shown in Fig. 1.3, and allows multiple resources to be shared on the same channel. For instance, in a scenario of a three-time slot TDMA, at least three vehicles among all possibilities are allowed to use the current channel without causing significant interference to others.

In modern communication systems, when we combine the principles of both TDMA and FDMA then we get a key concept of the wireless spectrum named Resource Blocks (RBs). Resource blocks are mainly considered to allocate the available resources for communications between vehicles with a stream of data packets. For instance, as depicted in Fig. 1.4, the combination of a specific time slot (t3) in a given time frame and a specific orthogonal channel (c4) from all available channels is called a resource block (RB).

To reduce interference and increase the success rate of V2V communications, we introduce a well-known approach named Half duplex transmission mode. In this mode, a vehicle cannot send or receive multiple data packets at the same time to/from a specific receiver or transmitter. Instead, it can either transmit a packet to a specific receiver or receive a

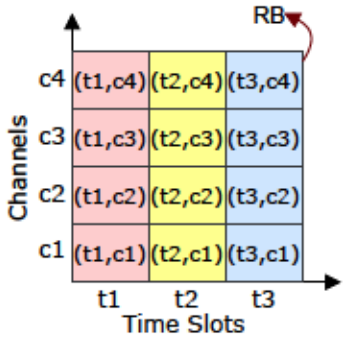


Figure 1.4: An explanation of resource blocks (RBs) where each RB consists of one orthogonal channel and one time slot. Here, we divide a time frame into three time slots (i.e., t1, t2, and t3) and a frequency band into four orthogonal channels (i.e., c1, c2, c3, and c4).

packet from a specific transmitter. As per the discussions above, when the resource block is empty and we have a data packet (i.e., 1 and 2) on a resource block (t1,c1) that is shown in Fig. 1.5 as black color, we can transmit the packet from its source node 1 to its destination node 2. However, owing to the Half-Duplex mode, another packet shown in red cannot be transmitted from its source node (2) to its destination node (1) as both of them are already allocated.

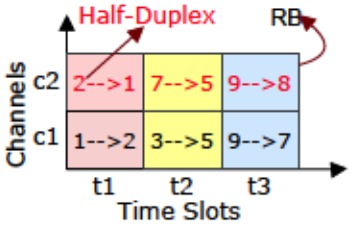


Figure 1.5: An explanation of Half-Duplex (HD) communication mode where a vehicle cannot receive or transmit at the same time. Here, packets shown on resource blocks (RBs) in red colors cannot be transmitted due to the Half-Duplex mode, while the colors in black can successfully be transmitted to their destinations.

To allocate even more resources to resource blocks for obtaining more successful communication, we introduce another crucial metric called Signal-to-Interference-plus-Noise Ratio (SINR). This metric checks the quality of the received signal strength, interference to the newly allocated receivers from other transmitters, and background noise. If the obtained SINR value at receivers is above a certain threshold β, then we can observe simultaneous transmissions in the network. Two different scenarios are shown in Fig. 1.6 where we marked them by black and red. It is observed that we have two simultaneous transmissions on resource blocks of (t1, c2) and (t3, c2). Here, we assume that the simultaneous transmissions on (t1, c2) are allowed due to the obtained SINR value which is equal to or greater than

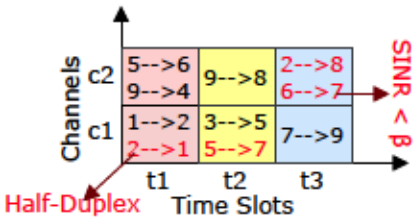


Figure 1.6: An explanation of signal-to-interference plus noise ratio (SINR). Here, we divide a time frame into three time slots (i.e., t1, t2, and t3) and a frequency band into two orthogonal channels (i.e., c1, and c2). Also, packets shown on resource blocks (RBs) in red colors cannot be transmitted due to either Half-Duplex or SINR or both of them, while the colors in black can successfully be transmitted to their destinations.

the SINR threshold β. So, we can do simultaneous transmissions for the available packets on t1, c2. On the other hand, we cannot do simultaneous transmissions even though both packets satisfy the Half-Duplex mode as the achieved SINR value is not equal or higher than the threshold sample. Therefore, only the source node 2 can send its data packet to its receiver node 8.

To show the importance of data and prevent collisions or hazardous situations during communications, we also introduce a metric called Age of Information (AoI) that focuses on how up-to-date the information is from the perspective of the receivers. In a V2V communication mechanism, this metric not only indicates the freshness of data packets from its sender to its receiver but also shows the time difference since the most recently delivered message was generated at the sender node. For instance, we can consider a scenario where a packet is just generated at time slot t=0 by a transmitter. In the next time slot when t=1, the newly generated packet by a transmitter would be ready for the transmission to the nearest receivers within its coverage area. At t=2, one of the destination nodes will receive that transmitted packet. Now, if we calculate the AoI from the perspective of the receivers at time slot t=2, then the most recently delivered message generated at the sender node will be 2 seconds (i.e., units are seconds).

In this thesis, scheduling itself is one of the complex tasks which determine the frequency of each transmission between all vehicles within its coverage area in the network. Then, with the addition of various challenging factors such as multi-hops, Half-Duplex, and SINR is make the scheduling decisions even more difficult. For instance, in some cases of V2V scenarios, the transmitted packet may need to be relayed through multiple V2V nodes before reaching its destination as the routing paths between sources and destinations are affected by communication distance or fading issues. Hence, changing vehicle movement with the

network topology dynamically may affect other transmissions. In another V2V scenario considering the Half-Duplex mode, we may have very limited resources as a node cannot transmit or receive at the same time. The consideration of this constraint poses another major aspect during scheduling. Afterward, due to the dynamic movements of vehicles, scheduling available resources on resource blocks under the SINR constraint is even more difficult as it is affected by potential interference, particularly for those vehicles which are often in close proximity. At that time, finding a correct routing path and schedule links by allocating RBs with a consideration of varying power transmission, half-duplex and SINR constraints in the network make the scheduling decisions even harder. Hence, to handle the arising challenges, we later devise effective machine learning and artificial intelligence-based approaches that play a crucial role in preventing the complexity of the scheduling algorithm while exchanging information in the network.

In this thesis, we discuss two separate studies to design an effective autonomous vehicular transportation system. Our first study (MVGCF, Maximizing V2V Group Communications, and Fairness) described in Chapter 3 is referred to as a unicast communication method, while the second study (Online Age of Information (AoI) Minimization Method) explained in Chapter 4 is offered to as a broadcast communication method [19]. The fundamental objectives are given below.

1. The first study of this thesis is to maximize the total number of communications for groups of vehicles while maintaining fairness among all V2V communication pairs
2. While another crucial study of the thesis is to minimize the total or average AoI of all data streams that are intended for autonomous vehicles in the network within the road segment for the entire time frame

Study 1 (maximizing the total number of communications while maintaining fairness for all V2V communication pairs):

Chapter 3 solves the routing paths (i.e., a set of intermediate links between source and destination pairs), resource allocation, and link scheduling for a stream of packets between vehicles within a multi-group communication configuration. For instance, groups of vehicles (i.e., police cars, ambulances, firefighters, buses, and city fleets) communicate with each member of its group [20]. This chapter aims at setting communication links between all vehicle pairs within a group utilizing WiFi technology to alleviate the load on cellular networks and then schedules transmission links to RBs. Since not all pairs have direct communication capabilities, the problem extends to relaying and scheduling data packets through multi-hop transmissions. The study aims to maximize communication efficiency among vehicle groups while ensuring fairness and allowing resource block reuse under the Half-Duplex and SINR constraint. It is assumed that by maximizing the total number of

data packets, the system will always choose a pair of nodes that are close to each other to save allocated resources. Therefore, a pair of nodes that are far from each other and require multi-hop transmissions, has a very low or maybe zero chance to be considered for a V2V data communication.

Study 2 (minimizing AoI for all data streams):

As traditional metrics like throughput and latency do not sufficiently capture data stream timeliness and freshness [21–23], Chapter 4 focuses on the minimization of AoI across all data streams in autonomous vehicular networks to prevent accidents or hazardous situations. Unlike the first objective, direct data stream connections between vehicle pairs are absent. Instead, a vehicle can participate in relaying or rebroadcasting data streams to nearby vehicles based on data importance while considering the limitations imposed by the half-duplex constraint, transmission range, and SINR thresholds [24,25]. To achieve the mentioned objective of this chapter, an array of decision-making information, including relaying, transmission timing, and data packet dropping, is required to minimize the total/average AoI of all data streams at all nodes for the entire time frame. Hence, several complexities arise in deciding which packets of data streams to broadcast over transmission links, scheduling links on time slots, and ensuring packet order transmission on multi-hop paths.

The contributions of the thesis can be summarized as follows:

- The studies of Multi-Group V2V communications and Age of Information (AoI) minimization are first mathematically formulated as a mixed integer linear programming (MILP) to obtain optimal solutions for static environments.
- Due to the NP-hardness of the introduced problems and overcoming the complexity of both optimization models, two scalable heuristic-based methods are proposed.
- We then formulate both problems as an MDP, and the structured framework for modeling and analyzing the problems is demonstrated.
- To make informed and effective decisions and solve both problems, two reinforcement learning (RL) based methods, namely Qlearning and DDQN are employed.
- To further enhance the learning behaviour of each RL agent and improve the performance of the aforementioned methods, two innovative hybrid heuristic-based RL methods are employed by combining the strength of the heuristic methods and RL algorithms.
- Finally, comprehensive comparisons between proposed approaches; conventional RL methods, hybrid methods, and heuristic methods, are shown with the MILP-based optimal solutions and Random method on small instances for both problems. Then,

we compare the performance of the introduced methods for both problems without the optimization model on both medium and large instances. Later, the effectiveness of the hybrid approaches in terms of the number of successful communications and attainment of max-min fairness is shown for the Multi-Group V2V communications problem. In contrast, the effectiveness of the hybrid approach is evaluated for the AoI minimization problem in terms of efficiently minimizing the expected weighted sum of AoI.

The rest of the thesis is organized as follows.

- Chapter 2 provides an overview of the related work, such as Routing and Scheduling to maximize communication throughput, Fairness, and a thorough review of the Age of Information. Besides, we provide explanations and limitations of the current studies.
- Chapter 3 gives an introduction and description of the Multi-Group V2V Communications problem. Then, we mathematically model the problem. We also show the hardness of the problem and explanations of the proposed scalable method. Furthermore, we evaluate the performance of the proposed methods with key findings.
- Chapter 4 gives a detailed introduction and description of the Age of Information (AoI) minimization problem. We then formulate the mathematical formulation. We also show the hardness of the problem and explanations of the proposed scalable method. Furthermore, we evaluate the performance of the proposed methods with key findings.
- Chapter 5 concludes the thesis, summarizing the findings and suggesting potential avenues for future research.

Chapter 2

Related Work

In Vehicle-to-Vehicle (V2V) communications, scheduling is considered to be a prominent process in arranging, controlling, and optimizing the operation of the service provider for any application. As a result, the key goal is to construct an effective resource allocation system that allows multiple vehicles to communicate simultaneously with each other [26]. Alongside, another crucial and reliable aspect in V2V communications is to maintain fairness in vehicular networks to ensure equitable and effective sharing of data among various vehicles. Also, the recent rapid growth of AoI scheduling that provides the freshness of information in V2V communications stands as a prominent research challenge because it requires an optimal selection of broadcasted nodes in V2V communications. In the following subsections, we present a detailed overview on each of them and highlight the novelty of our work.

2.0.1 Routing and Scheduling to maximize communication throughputs

The authors in [26] investigated TDMA, in which the time frame is divided into numerous time slots in order to share the same frequency channel without interfering with other nodes in vehicle-to-everything (V2X) and classify them based on their scheduling technique known as fuzzy logic-based resource allocation. The proposed method was accomplished by allocating resources to all RBs, followed by the Half-Duplex characteristics. Similar to our work, the goal is to maximize network throughput. However, they did not consider any priority queues and fairness in their system. The authors in [27] focused on maximizing system throughput through a power control scheme called Belief Propagation based on Real-time Update of Messages (BPRUM), where multiple links can share the same RB.

On the other hand, the authors in [28] proposed a beamforming-based multi-point transmission algorithm to improve system throughput by considering simultaneous transmissions on several channels. Similarly, in [29], the authors proposed an approach to maximize system throughput by reusing an RB. Additionally, they analyzed and programmed the po-

tential reduction in power consumption in a cellular network using MILP and solved it by a Gurobi optimizer. Moreover, the authors in [30] presented an interference-aware relay selection mechanism that schedules transmission and routing in a multi-hop network, aiming to improve network throughput while considering the SINR model.

In [31,32], the authors proposed a RB sharing algorithm that allows multiple vehicles to share a single RB, with the objective of maximizing concurrent data transmissions while adhering to SINR constraints. The authors in [33] examined the compressive data gathering (CDG) and scheduling problem in wireless sensor networks and divided it into two subproblems: tree construction and link scheduling. They also discussed how well the system handled transmissions and gathered delays.

Vehicle-to-everything (V2X) communication is essential for road safety while maintaining high dependability and low latency. A unique strategy of joint power control and resource allocation mode selection was proposed, utilizing two resource allocation methods in [34] to solve the resource allocation problem under diverse networks. Also, they aimed to enhance the overall information value of V2X communication while minimizing SINR and maximizing transmit power.

Similar to this, the study in [35] discussed resource block allocation in D2D-based V2V communications under the restricted condition of SINR. They used function mapping to convert the problem into an inference problem on a factor graph model and explained the message-passing technique. Afterward, to address the issue, a BPRUM algorithm was proposed to maximize the concurrent links while ensuring the satisfaction of quality of service (QoS).

However, they did not formulate their problem as Mixed Integer Linear Programming and consider fairness in their system. Unlike the aforementioned studies, our work strongly emphasizes maintaining max-min fairness among communication pairs while maximizing the total number of V2V communications considering the RB allocation and scheduling techniques. By incorporating fairness as one of the key objectives, our approach aims to ensure equitable utilization of resources among all V2V communication pairs, leading to a more balanced and efficient system operation.

The authors in [36] introduced RL to solve a Constraint Satisfaction Problem (CSP) in cellular networks, where their goal was to ensure quick data transmission while saving power. In [37], they introduced a decentralized method called Fed-MARL that combines deep reinforcement learning (DRL) and federated learning (FL) to improve communications between vehicles (V2V) while maximizing data transmission over cellular links. They created individual "agents" for each V2V pair using dueling double deep Q-network (D3QN) and enabled these agents to work together by sharing information.

In [38], the authors investigated the resource allocation problem in vehicular communications using deep Q-network (DQN) and deep deterministic policy-gradient (DDPG)

approaches. The former was responsible for sub-band assignment, while the latter was used for continuous power allocation. Also, to handle dynamic environments, a meta-based DRL algorithm was introduced to improve adaptability.

In [39], a decoupling approach was considered in vehicular communications for channel allocation and power control schemes. They also proposed a hybrid approach to maximize the systems' efficiency while ensuring the scheduling of V2V links on RBs.

The authors in [40] studied how to allocate channels and control power in vehicular and cellular networks to ensure good quality of services (QoS) for different types of traffic. They proposed a Multi-agent deep deterministic policy gradient (MADDPG) framework to solve this problem for V2V communications, where the primary objective is to maximize the utility of vehicular users while ensuring QoS for all users. However, it is important to note that their framework did not address the issue of fairness among all V2V links. Unlike the problem proposed in this paper, our research emphasizes the importance of fairness among V2V links in addition to system performance.

Similarly, in [41], the authors explored the allocation of sub-channels and power control in a group of connected and self-driving vehicles to ensure stable communication. In their work, two methods have been compared: one where a central station makes decisions based on limited information about the links, and another where each vehicle independently uses RL to make decisions. The goal is to maximize the transmission rate while maintaining stability. However, their approach has limitations in terms of fairness. In [42], the authors focused on a problem known as multi-task offloading (MTO), which involves offloading tasks while varying network requirements. They proposed a method called SMRL-MTO that uses meta-reinforcement learning to adapt quickly to different situations. However, they did not consider throughput and the amount of information that can be transmitted in their system while optimizing task completion time in various offloading scenarios.

In [43], the authors proposed a DRL-based scheme called JoBARS to improve communication in vehicular networks using mmWave base stations. They considered joint beam allocation and relay selection to optimize the total transmission rate. They also introduced a rate punishment restriction and relaying incentive mechanism to ensure high-quality service for vehicles and fair relay selection but did not consider throughput in their system.

Although the reviewed papers presented innovative approaches for resource allocation in vehicular networks and used several RL and/or DRL algorithms for implementation, they did not explicitly address both throughput and fairness for group-based V2V communications in their systems.

2.0.2 Fairness

In order to analyze a drone-assisted vehicular network problem, the authors in [44] tried to maximize network transmission utility while reducing random data transmission. Unlike

our work, despite explaining the system's fairness, they did not maintain any queues based on the priority of requests. The authors in [45] presented a greedy algorithm that aims to maximize throughput while minimizing service disruption for multiple requesting vehicles within the range of roadside Units (RSUs). In order to serve the vehicles' download requests, they developed a joint frequency scheduling and power control scheme over both I2V and V2V communications, where the former is addressed as a linear programming problem. Unlike our study, this paper did not concentrate on improving the amount of fairness based on the request's priority queue while ensuring fairness in vehicular communications.

In [46], the authors investigated a cellular network where D2D nodes are allowed to share the same time slots with cellular user equipments (CUEs) under certain constraints. Afterward, to maximize the normalized sum of throughput for all D2D users, a hybrid spectrum scheme approach, which learned an optimal strategy to allocate resources autonomously, was proposed. Later, to address fairness issues among all D2D users, they considered the double deep Q-network (DDQN) to achieve fairness for all D2D users. However, they did not formulate their problem as MILP and explore the V2V communications using WiFi with the reuse of resource blocks under the Half-Duplex constraint. Also, they failed to devise a heuristic method using priority queues while considering fairness for a large network in their system. Moreover, they did not show the concept of a hybrid deep learning approach to run the experiments over an extremely large network.

Later, authors in [47] showed that not all items were broadcasted several times to maximize the channel bandwidth. Hence, the system created unfairness to others. Due to having such unfairness in the system, they introduced a fairness-friendly (FF) solution to balance the trade-off between fairness and throughput. However, the proposed algorithm provides a fair service if and only if the value between Uniformly Distributed Request Served Percentage (UDRSP) and Skewed Distributed Request Served Percentage (SDRSP) are equal or nearly equivalent.

In a study presented in [48], the authors showed that popular data items are broadcasted multiple times to maximize the bandwidth of the broadcast channel, which may create unfairness for non-popular items. To balance throughput and fairness, they proposed a solution called RoadNet that prioritizes transmissions based on the user satisfaction ratio. In contrast, the authors in [49] analyzed a satellite-terrestrial integrated network (STIN) and showed fairness in user association while maximizing the throughput by reusing RBs.

However, in the above-mentioned studies, none of them used max-min fairness by maximizing the number of communications for the one with the least number of communications. In addition to the fact that none of them studied the combinatorial problem presented in this paper.

2.0.3 Age of Information

The study in [50] investigated the AoI minimization problem in wireless networks with time-varying channels. They formulated the relaxed problem as a constrained Markov decision process (CMDP) and utilized linear programming to find an optimal solution. Building an optimal policy derived from the relaxed problem, they proposed a truncated scheduling policy that adheres to the original strict power constraint while achieving effective AoI minimization. Similar to our work, they also aimed to minimize the total weighted sum of AoI. However, they did not formulate their problem as MILP and explore the V2V communications with the reuse of resource blocks under the Half-Duplex constraint.

In [51], the authors introduced WiFresh, which aimed to achieve nearly optimal information freshness in wireless networks, even in overloaded networks. Their experimental findings demonstrated the effectiveness of WiFresh, which incorporates two strategies in improving information freshness compared to standard WiFi networks to achieve a significant improvement of two orders of magnitude. In [52], they examined a base station that handles traffic streams in an IoT network with mobile edge computing (MEC) assistance. To minimize the expected sum of AoI, the authors initially employed linear programming (LP) to derive an optimal policy. However, due to the complexity of the LP approach, they later introduced low-complexity algorithms.

The authors of [53] investigated an algorithm that calculates the necessary charging time for each source node while considering the weighted AoI in a wireless-powered network. Similar to our work, their goal was to minimize the total weighted sum of AoI. However, they did not formulate their problem as MILP and explore the V2V communications using WiFi with the reuse of resource blocks under the Half-Duplex constraint.

The authors of [54] considered transmission capacity, focusing on a network scenario where a base station regularly updates multiple users with randomly arriving information. By considering offline and online scenarios, the scheduling algorithms were proposed to minimize the average AoI in the wireless network. The authors of [55] studied AoI-oriented scheduling for a wireless multiuser uplink network and formulated the scheduling problem using a partially observable Markov decision process (POMDP). To simplify the problem, they transformed the POMDP into a belief Markov decision process (belief-MDP). Then, using the belief-MDP framework, they developed the POMW policy to minimize the expected weighted sum of AoI in the next slot.

Similarly, the authors of [56] focused on wireless powered sensor networks (WPSNs) and aimed to minimize the average weighted sum of AoI. Then, they formulated it as a multi-stage stochastic non-linear integer programming (NLP) challenge. To tackle this problem effectively, they devised an algorithm named DRLL that combines Deep Reinforcement Learning (DRL) and Lyapunov optimization methods. The DRLL algorithm efficiently

manages the scheduling of energy transfer and packet transmission in the WPSNs. The authors of [54–56] discussed transmission scheduling strategies and introduced RL-based solutions to minimize the total weighted sum of AoI. However, they did not formulate their problem as MILP and explore the V2V communications using WiFi with the reuse of resource blocks under the Half-Duplex constraint.

The authors of [57] investigated the transmission scheduling strategy for autonomous underwater vehicles (AUVs) in an underwater wireless sensor network (UWSN), where AUVs with random lifetime are considered. Also, the AUVs were responsible for selecting an underwater data collection station to update data, which was then uploaded to the surface base station. To optimize the scheduling strategy for the AUV and prove the threshold structure characteristics of the optimal strategy, the problem was formulated as a discounted Markov decision process. Similar to our work, they discussed transmission scheduling strategy and introduced RL-based solutions to deal with the dynamic nature of the environment. However, the goal was not to minimize the weighted sum of AoI allowing multiple transmissions under the Half-Duplex constraint.

The authors of [58] focused on analyzing the AoI performance in a multi-source system. The system consists of multiple sources generating updates, with the constraint that only one update was transmitted to a monitor at any given time. The authors considered four different scheduling policies, including random scheduling, round-robin scheduling, age-greedy scheduling, and the Whittle index-based policy, to compare the AoI performance in such a multi-source system. In [59], the authors considered a base station scheduling that broadcasts status updates containing randomly arriving information to multiple nodes over a shared bandwidth-limited channel. They proposed optimal stationary randomized and Max-Weight policies, where the former is investigated when the transmission feedback is unavailable, and the latter is introduced when the feedback is available. With the help of these approaches, they aimed to minimize the weighted sum of the average AoS of all the nodes while meeting the minimum throughput requirement of each node. Though the goal is to minimize the total weighted sum of AoI by varying several heuristic methods, they failed to validate their performance with the help of an optimal solution. Alongside this, scheduling decision and the reuses of resource blocks under Half-Duplex constraint was not explored.

In this study, unlike mentioned works above, we are mainly focused on the problem of AoI in V2V communications, where multiple transmissions are allocated to resource blocks while adhering to the Half-Duplex constraint. This paper aims to model the problem as a Markov Decision Process (MDP). Furthermore, we devise a hybrid heuristic-based reinforcement learning method to find an efficient solution that minimizes the total AoI in intelligent transportation systems.

To the best of our knowledge, we are the first who studies the combinatorial problem

of routing and scheduling data packets in two different communication methods (Unicast and Broadcast). In the MVGCF problem, we consider routing and scheduling for multiple V2V communication pairs of vehicles that communicate in groups to maximize the total number of successful communications while maintaining fairness, while the OAMM problem was responsible for traversal of broadcasted nodes between sources and destinations to minimize the total or average AoI of all data streams for autonomous vehicles. Both of them pose significant challenges, where the MVGCF problem has to find appropriate paths through multi-hops to establish communication between a pair of vehicles, schedule links and allocate RBs, reuse RBs by tuning power transmission and considering SINR for simultaneous transmissions. On the other hand, the OAMM problem takes into account resource allocation under the Half-Duplex constraint, traversal of broadcasted nodes between sources and destinations, link scheduling, and the reuse of resource blocks.

Chapter 3

Maximizing Communications and Fairness Within Groups of Vehicles

Intelligent vehicle-to-vehicle (V2V) communications offer promising solutions to the future of vehicular networks and autonomous driving. This chapter investigates the challenge of establishing efficient communications among pairs of vehicles using WiFi technology. It is assumed that there are multiple groups, and vehicles which belong to a group have a dedicated data stream to each vehicle in their group. As a result, each source-destination pair within a group is associated with a distinct data stream, providing an efficient and reliable means of data transmission. However, due to communication range limitations, not all pairs can communicate directly, and they have to relay data packets through multi-hops. The objective of this chapter is to maximize the total number of communications while ensuring fairness among V2V communication pairs. To achieve this, the problem of scheduling and relaying data through multi-hop vehicle nodes for source-destination pairs by sharing resource blocks within a time frame under the constraint of signal-to-interference-plus-noise ratio (SINR) is investigated. In this chapter, first, the problem of Maximizing V2V Group Communications and Fairness (MVGCF) is mathematically formulated to find optimal solutions. Then, it is shown that the problem is NP-hard, and owing to its complexity, a scalable method is proposed for more extensive networks. To tackle the dynamic nature of the environment and vehicle mobility, the problem is modeled as MDP, and two reinforcement learning (RL) algorithms, namely Qlearning and Double Deep Q-Networks (DDQN), are proposed to solve it. Furthermore, to improve the performance of both methods, two hybrid heuristic-based RL methods, namely MVGCF-Qlearning and MVGCF-DDQN, are devised. The numerical results demonstrate the effectiveness of the hybrid methods in terms of the number of successful communications and max-min fairness when compared to a state-of-the-art heuristic method and the conventional RL methods for small, medium, and large networks.

3.1 Introduction

Intelligent transportation systems (ITS) are developing quickly to offer cutting-edge services for vehicles, such as infotainment, traffic control and safety among others. ITS also foster intelligent vehicular environments through a fully connected paradigm known as vehicular communication networks [5], which enable moving vehicles to remain online and linked to their surroundings while traveling. In that sense, vehicular communication networks provide various activities such as autonomous driving, road surveillance, source and destination locations, dangerous situation detection, data sharing, information about driving behaviors, communication services between neighboring vehicles, and many more [7]. These activities result in more efficient, safer, and comfortable driving experiences and create new opportunities in various business sectors. Therefore, the networking industry and academia have expressed a strong interest in developing vehicular communication networks and leveraging relevant services.

The recent growth of group communication applications has been widely studied because of vehicles' high data packet delivery ratios, and throughput. As a result, the communications between groups of vehicles (i.e.; police cars, ambulances, fire-fighters, buses, and city fleets) are required in the network to provide high throughputs, and fewer network congestions [20]. For example, when a patient requests an ambulance in case of emergency, a nearest group member of ambulances can quickly respond via WiFi to the requested place to provide services and learn about high mobility and traffic congestion from other members on the road within the communication range by sharing its current location. Also, until reaching the final destination, it can automatically learn about the road's ongoing mobility and traffic from its group members within the coverage, significantly making traveling more comfortable.

In this chapter, we consider group communications, where any pair of nodes (vehicles) that belong to the same group has a data stream to transmit to each other. We refer to any source-destination pair as a communication pair. Because of a communication range (which can be defined by transmission power and fading), not all the communication pairs have a direct link. Hence, for some, we need to create a path by relaying transmissions over multi-hop nodes (vehicles). We consider a time division multiple access (TDMA) medium, where the time frame is divided into equal time slots. The size of the time frame is considered very small such that the position of vehicles, based on their maximum speed, will not significantly change to affect the broadcasting of data even when having multi-hop transmissions. Therefore, we can consider a new time frame for the future vehicle position change. We also consider that the bandwidth [18] is divided into equal orthogonal channels, and we assume that the size of data packets is fixed that can be fitted and transmitted using one resource block (i.e., one channel and one time slot). However, we might have

simultaneous transmissions in a resource block (RB) if signal-to-interference-plus-noise ratio (SINR) at receivers allows that.

In this chapter, our problem is first to find a route (path) between each pair of nodes in a group, and then schedule transmissions (links) by allocating RBs. The objective here is to maximize the total number of data packets while maintaining fairness among all communication pairs. It is observed that by maximizing the total number of data packets, the system will always choose a pair of nodes which are close to each other to save allocated resources. Therefore, a pair of nodes which are far from each other and require multi-hop transmissions, has a very low or maybe zero chance to be considered for a V2V data communication. Hence, it is also essential to maintain communication fairness by maximizing the total number of V2V communication packets for a pair of nodes which have communicated the least within a time frame (i.e., maximizing the minimum V2V communication pair). In other words, it tries to achieve fairness by maximizing the V2V communication pair with a minimum number of communications, promoting equitable communication opportunities across all pairs of nodes.

The problem of routing and scheduling data packets for multiple V2V communication pairs in multiple groups of vehicles to maximize the total number of successful communications while maintaining fairness (MVGCF, Maximizing V2V Group Communications and Fairness) is a combinatorial and challenging problem; need to assign links for communication pairs, schedule links on RBs, and consider RB reuse for simultaneous transmissions by considering SINR while maintaining fairness among all communication pairs. Therefore, solving such a combinatorial problem is not a trivial task, and to the best of our knowledge, no such problem has been tackled and solved before. However, we have modeled the problem mathematically and solved it using the optimization model and the MVGCF method. In this chapter, to consider the dynamic nature of the environment more precisely, we model the problem as Markov Decision Process (MDP), and solve it using two reinforcement learning (RL) algorithms, namely Qlearning and Double Deep Q-Networks (DDQN). Furthermore, to improve the performance of both methods, we merge our heuristic algorithm MVGCF with them and propose two hybrid heuristic-based RL methods, namely MVGCF-Qlearning and MVGCF-DDQN.

The rest of the chapter is organized as follows. The system model and problem description are presented in Section 3.2, while the mathematical formulation is given in Section 3.3. The hardness of the problem and explanations of the proposed scalable method are given in Section 3.4. Section 3.5 presents the explanations of both RL and hybrid RL methods. Section 3.6 evaluates the performance of the proposed methods, and Section 3.7 summarizes the chapter with key findings and suggesting potential avenues for future research.

3.2 System Model and Problem Description

3.2.1 System Model

We consider a road structure, as shown in Fig. 3.1, where vehicles are grouped in different groups and that each vehicle has a data stream with another vehicle in its group within a fixed range. The size of this range depends on the importance of data to be shared with other group members, which can vary from a few hundred meters to a few kilometers. In the figure, for example, we have three groups shown with red, blue, and yellow colors; a member of each group has a data stream with each and every vehicle that belongs to the same group. We consider the system over multiple time frames. Each frame is partitioned into equal time slots, $t = 1, 2, ..., T$. The total number of time slots in a frame is T. We consider our system at each time frame, as shown in the figure, as a graph $\mathbb{G} = (N, E)$, where N is a set of nodes (vehicles) in the road segment, and E is a set of edges (links), which based on different factors such as, the distance, maximum communication power, presence of obstacles and fading, connect any two nodes. In the graph, edges are shown with dotted lines. The graph $\mathbb{G}$ is constructed in advance at the beginning of each time frame. To be noted that the size of the time frame T is considered very small such that the position of vehicles, based on their maximum speed, does not significantly change to affect the constructed graph $\mathbb{G}$. Therefore, a new graph will be constructed for the next time frame. The vehicles' speed is considered to follow a truncated Gaussian distribution ranging from ν_{min} to ν_{max} [60], and vehicles travel at random speed [30,61]. Also, the vehicles' arrival into the road segment is considered to follow a Poisson distribution with density ρ Vehicle/Km [62].

For simplicity, we assume in the network we have M communication pairs to transmit data packets from a source to a destination, and since not all the sources and destinations have a direct communication link, sometimes a data packet must be transmitted over multi-hops as shown in the figure with arrows. We consider that the bandwidth is divided into multiple orthogonal channels, $c = 1, 2, .., C$, where C is the total number of channels. We also consider a Time Division Multiple Access (TDMA) medium access where time is divided into slots of equal length as explained earlier. We assume any data packet can be fitted and transmitted over one resource block (i.e, channel c and time slot t). We assume at each time slot a node can either transmit or receive one data packet because of the Half-Duplex transmission mode. However, we might have simultaneous transmissions in the network if different channels have been considered or/and the signal-to-interference plus noise ratio (SINR) at receivers is above a certain threshold β. Let P_{ij}, G_{ij}, and α be respectively the transmission power, distance, and power decay from transmitter i to receiver j, then, the SINR under the physical interference model [31,63] in the presence of

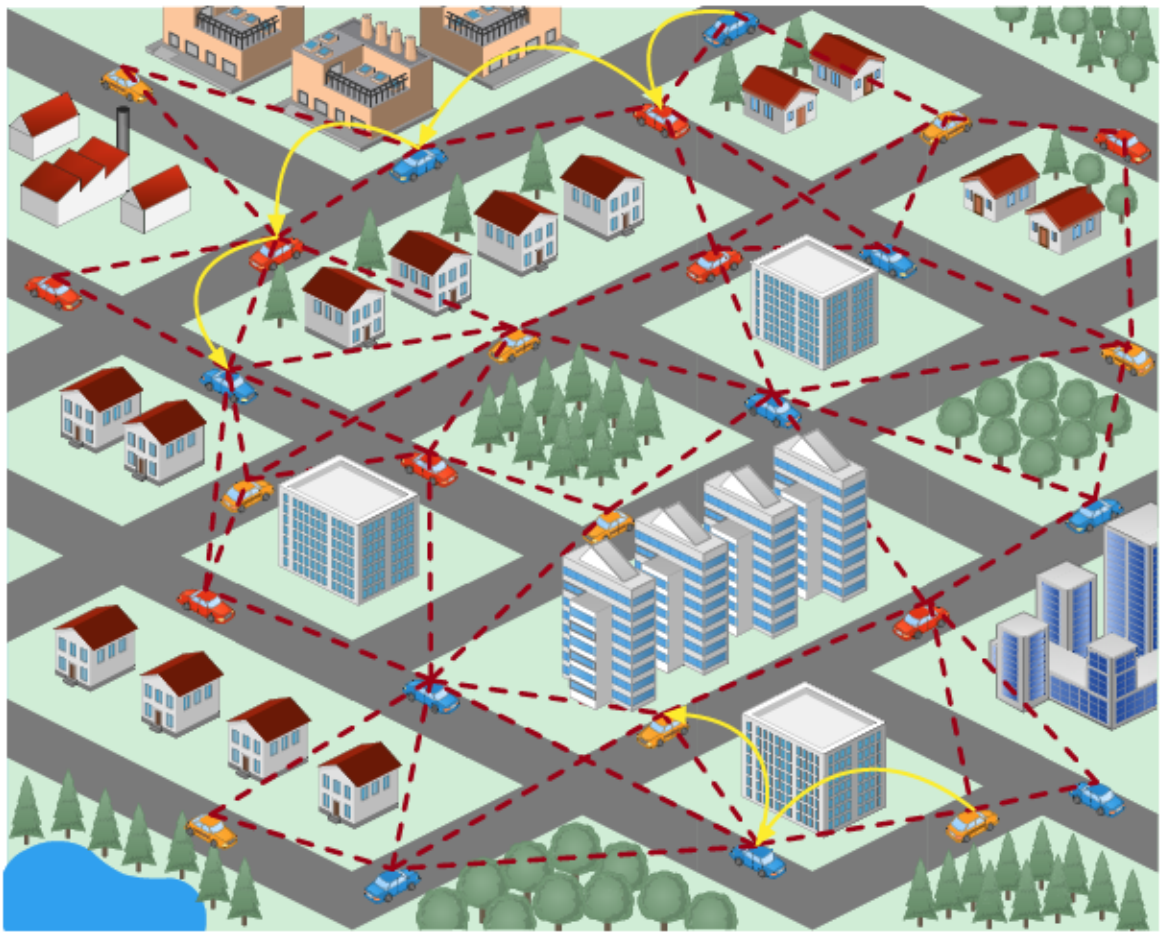


Figure 3.1: Illustration of the system model; the wireless communication between vehicles is shown by dotted lines and the packet transmission for a V2V communication pair is shown by multiple arrows from a source to a destination.

concurrent transmissions is obtained as follows:

$$SINR_{(i,j)} = \frac{P_{ij} G_{ij}^{-\alpha}}{\eta + \sum_{\forall (h,k) \in E: h \neq i} P_{hj} G_{hj}^{-\alpha}} \geq \beta, \forall (i,j) \in E \tag{3.1}$$

where η is the background noise.

3.2.2 Problem Definition

We are interested in maximizing the total number of communication packets in the network within a time frame T for all V2V communication pairs while maintaining fairness. For most of the V2V communication pairs, source nodes may have no direct links with destination nodes, hence a packet has to be relayed and retransmitted over multi-hops. Therefore,

a node, other than its own data packets, may retransmit packets that belong to other communication pairs. Consequently, there should be a scheduler that coordinates these transmissions for all nodes and for all different packets over multiple channels and time slots (resource blocks). However, a node due to the Half-Duplex characteristics cannot receive and transmit more than a packet at each time slot even though there are multiple orthogonal channels. Also, a node cannot retransmit a packet unless it receives it first. So, the order of transmissions in a multi-hop path from a source to a destination is at most an important constraint.

Now, maximizing the total number of communications might affect the fairness in the system. If the objective is to maximize the total number of communication packets, then the system will mostly allocate resource blocks to V2V communication pairs that are close to each other in order to save resources for other transmissions. In this way, V2V communication pairs, which are far away from each other and require long routing paths, might have a very low, even zero chance to be considered for data transmissions. Therefore, the scheduler should count the total number of transmitted data packets for each communication pair and tries to maximize it if it has been scheduled the least among all V2V communication pairs (i.e., maximizing the minimum V2V communication pair) to maintain fairness among all communication pairs within a time frame T.

Problem Definition (MVGCF)

Given a graph $\mathbb{G}$ of N nodes connected through E edges, the problem of MVGCF is to allocate resource blocks (time slots and channels) for M V2V communication pairs to transmit a large number of data packets D in a road segment within a time frame T (where the frame is partitioned into multiple equal time slots) such that the total number of communication packets in the network is maximized while maintaining fairness among all V2V communication pairs.

3.3 Problem Formulation Using Linear Programming

In this section, we mathematically formulate the problem as a mixed integer linear programming (MILP). The used notations are listed in Table 3.1. Let $X_m^d \in \{0,1\}$ be the indicator of successful transmission of data packet d for V2V communication pair m, and F be the minimum number of data packets that a V2V communication pair has sent among all communication pairs to ensure fairness. The objective of the optimization model is to maximize the total number of V2V communication packets while maintaining fairness. It

Table 3.1: Notations used in problem formulation

Parameters		
N		Set of nodes.
E		Set of edges (links).
M		Set of V2V communication pairs.
T		Time frame (total number of time slots).
C		Total number of sub-channels.
B		Large constant bigger than T.
K		Large constant.
G_{ij}		Distance from transmitter i to receiver j.
β		SINR threshold.
η		Background noise.
P_{MAX}		Maximum transmission power.
P_{MIN}		Minimum transmission power.
D		Maximum number of data packets that are predicted to be transmitted within the time frame T.
Variables		
X_m^d	$\in \{0,1\}$	Indicate whether data packet d for V2V communication pair m has been transmitted.
F	≥ 0	Minimum number of data packets that a V2V communication pair has sent among all pairs.
$P_{ij,m}^{d,c,t}$	≥ 0	Transmission power for data packet d of V2V com. pair m on link (i,j) at time slot t using channel c.
$R_{ij,m}^d$	$\in \{0,1\}$	Indicate whether the data packet d for V2V Com. pair m has been traverse on link (i,j).
$S_{ij,m}^{d,c,t}$	$\in \{0,1\}$	Indicate whether the link (i,j) for data packet d of V2V com. pair m is scheduled at time slot t using sub-channel c.

can be mathematically written as follows:

$$Maximize \qquad \sum_{m \in M} \sum_{d=1}^{D} X_m^d + F \tag{3.2}$$

subject to: (3.3) - (3.15), where these constraints are derived in details in Sections 3.3.1 to 3.3.8.

The first term in the objective function corresponds to the total number of successful transmitted data packets, and the second one maximizes the minimum transmitted data packets for V2V communication pairs to ensure fairness. The details of the constraints are given in the following:

3.3.1 Fairness

Here, max-min fairness is considered. Max-min fairness is a principle that seeks to distribute resources equitably among multiple users or applications in a system. By maintaining a

minimum allocation of resources for each user before distributing any remaining resources to others, this approach not only prevents any user from being completely deprived of resources but also promotes efficient allocation [64]. In order to obtain max-min fairness, the following constraint is required to find the minimum number of data packets that a V2V communication pair has sent among all communication pairs:

$$F \leq \sum_{d=1}^{D} X_m^d \qquad \forall m \in M. \tag{3.3}$$

3.3.2 Data Routing

Let $R_{ij,m}^d \in \{0,1\}$ be the indicator for a data packet d to be traversed on link (i,j) for a V2V communication pair m. The following constraints are required in order to construct a routing-path for a data packet d to be traversed from the source to the destination of each V2V communication pair m:

$$\sum_{j:(i,j)\in E} R_{ij,m}^d - \sum_{j:(j,i)\in E} R_{ji,m}^d = \begin{cases} 1, & i = Source_m; \\ -1, & i = Dest._m; \\ 0, & otherwise. \end{cases} \qquad \forall m \in M, d = 1...D. \tag{3.4}$$

The above constraints obtain the difference between the number of incoming and outgoing transmissions of the data packet d on node j. If node i is the source of the communication pair m, denoted by $Source_m$, it originates a data packet, and hence the difference between the number of outgoing and incoming data packet d is one. If node i is the destination of the communication pair m, denoted by $Dest_m$, then the difference is -1, since the number of incoming active links to node j is zero and the number of outgoing active links is one. Consequently, when node i is a relay node or neutral (none of the above), the difference is zero; that is, if there is an incoming link to a relay node, there should be an outgoing link, and if there is no incoming link, there must be no outgoing link as well.

In addition, the following constraint ensures that a link cannot act as a bidirectional link in a routing data for loop avoidance:

$$R_{ij,m}^d + R_{ji,m}^d \leq 1 \qquad \forall (i,j) \in E, \forall m \in M, d = 1...D. \tag{3.5}$$

3.3.3 Link Scheduling

Let $S_{ij,m}^{d,c,t}$ indicates whether link (i,j) for data packet d of V2V communication pair m in sub-channel c is scheduled at time slot t or not. The following constraint enforces a link

not to be scheduled within a time-frame T if it is a non-active link:

$$S_{ij,m}^{d,c,t} \leq R_{ij,m}^{d} \qquad \forall (i,j) \in E, \forall m \in M, d = 1...D, \; c = 1..C, t = 1..T. \tag{3.6}$$

3.3.4 Transmitted Data

The following constraints assert that the data packet d for the communication pair m is transmitted if the links on the routing path from the source to the destination have been scheduled; we can assure that by checking whether the outgoing/incoming link to/from source/destination has been scheduled:

$$X_m^d \leq \sum_{t=1}^{T} \sum_{c=1}^{C} \sum_{(i,j) \in E : i = Source_m} S_{ij,m}^{d,c,t} \qquad \forall m \in M, \; d = 1...D. \tag{3.7}$$

$$X_m^d \leq \sum_{t=1}^{T} \sum_{c=1}^{C} \sum_{(i,j) \in E : j = Dest._m} S_{ij,m}^{d,c,t} \qquad \forall m \in M, \; d = 1...D. \tag{3.8}$$

3.3.5 Simultaneous Transmissions

To reduce interference and increase the successful transmission rate, we restrict a node not to receive multiple data packets from different transmitters at the same time. Similarly, we restrict a node to transmit several packets simultaneously.

3.3.5.1 Simultaneous Receiving

This constraint ensures that a receiver does not receive data packets from multiple transmitters simultaneously:

$$\sum_{m \in M} \sum_{d=1}^{D} \sum_{c=1}^{C} \sum_{i:(i,j) \in E} S_{ij,m}^{d,c,t} \leq 1 \qquad \forall j \in N, t = 1...T. \tag{3.9}$$

3.3.5.2 Simultaneous Transmitting

The following constraint restricts a transmitter not to transmit several data packets to different receivers at the same time:

$$\sum_{m\in M}\sum_{d=1}^{D}\sum_{c=1}^{C}\sum_{j:(i,j)\in E} S_{ij,m}^{d,c,t} \leq 1 \qquad \forall i \in N, t = 1...T. \tag{3.10}$$

3.3.5.3 Simultaneous Receive and Transmit

Using the following constraint, we ensure that a node does not transmit and receive at the same time:

$$\sum_{m\in M}\sum_{d=1}^{D}\sum_{c=1}^{C} S_{ij,m}^{d,c,t} + \sum_{m\in M}\sum_{d=1}^{D}\sum_{c=1}^{C} S_{ij,m}^{d,c,t} \leq 1 \qquad \forall (i,k)\&(k,j) \in E, t = 1...T. \tag{3.11}$$

3.3.6 Power Limits

In the following, we force the power of non active links to be zero; that is when $S_{ij,t}^{m,n} = 0 \Rightarrow P_{ij,t}^{m,n} = 0$, and $P_{ij,t}^{m,n} > 0$, only when $S_{ij,t}^{m,n} = 1$:

$$P_{ij,m}^{d,c,t} \leq P_{MAX} S_{ij,m}^{d,c,t} \qquad c = 1...C, t = 1...T, \quad \forall (i,j) \in E, \forall m \in M, d = 1...D. \tag{3.12}$$

$$P_{ij,m}^{d,c,t} \geq P_{MIN} S_{ij,m}^{d,c,t} \qquad c = 1...C, t = 1...T, \quad \forall (i,j) \in E, \forall m \in M, d = 1...D. \tag{3.13}$$

where P_{MAX} and P_{MIN} are the maximum and minimum allowed power transmissions, respectively.

3.3.7 SINR

The following constraint ensures that the SINR for active links are above the SINR threshold β:

$$P_{ij,m}^{d,c,t} G_{ij}^{-\alpha} + K(1 - S_{ij,m}^{d,c,t}) \geq \beta(\eta + \sum_{\overline{m} \in M} \sum_{\overline{d}=1}^{D} \sum_{\substack{(k,h) \in E \\ k \neq i}} P_{kh,\overline{m}}^{\overline{d},c,t} G_{kj}^{-\alpha} S_{kh,\overline{m}}^{\overline{d},c,t}) \tag{3.14}$$
$$\forall (i,j) \in E, \forall m \in M, d = 1...D, c = 1...C, t = 1...T.$$

To simplify the above constraint, let K be a big constant that satisfies the following: $K \geq \eta + \sum_{(i,j) \in L} P_{ij} G_{ij}^{-\alpha}$. If link (i,j) is active in time slot t (i.e., $S_{ij,m}^{d,c,t} = 1$), then (3.14) reduces to SINR expression (3.1).

3.3.8 Transmission Order

This constraint is required to ensure that a relay node cannot transmit unless it receives data from its previous node. That is, a link (k,j) for the V2V communication pair m can be scheduled at time slot t, if link (i,k) for the same packet has been scheduled before (i.e., in a time slot between 1 and $t-1$).

$$\sum_{\overline{c}=1}^{C} \sum_{\overline{t}=1}^{t-1} \sum_{i:(i,k) \in E} S_{ik,m}^{d,\overline{c},\overline{t}} + B(1 - S_{kj,m}^{d,c,t}) \geq \sum_{i:(i,k) \in E} R_{ik,m}^{d} \tag{3.15}$$
$$\forall (k,j) \in E, \forall m \in M, d = 1...D, c = 1...C, t = 1...T.$$

where B is a big constant, which is bigger than the number of time slots in T. When $S_{kj,m}^{d,c,t} = 0$, inequality (3.15) is always satisfied. But, when $S_{kj,m}^{d,c,t} = 1$, (3.15) reduces to $\sum_{\overline{c}=1}^{C} \sum_{\overline{t}=1}^{t-1} \sum_{i:(i,k) \in E} S_{ik,m}^{d,\overline{c},\overline{t}} \geq \sum_{i:(i,k) \in E} R_{ik,m}^{d}$, which implies that any link carrying packet for communication pair m coming to node k had to be activated at a time slot between 1 and $t-1$, otherwise, node k can not transmit (or, link (k,j) can not be active, i.e., $S_{kj,m}^{d,c,t} \neq 1$) at time slot t.

3.4 NP-Hardness of the MVGCF Problem and the Proposed Algorithmic Method

In this section, we first prove the NP-hardness of the MVGCF problem, and then propose a heuristic method to solve it.

3.4.1 NP-Hardness of the MVGCF Problem

The maximization of V2V group communications and fairness problem is complex because it must address multiple challenges: 1) finding the connected routing path for each V2V communication pair, (2) scheduling links over RBs by considering the Half-Duplex and SINR constraints for each V2V communication pair, and 3) maintaining fairness among all communication pairs. Note that the scheduling problem under SINR is generally an NP-hard problem [65]. If we consider our problem to be a series of scheduling a maximum number of V2V links in each time slot, we can decompose our scheduling problem into a series of Max-Connections Scheduling [66] sub-problems. The Max-Connections Scheduling problem is to choose appropriate transmission power levels to maximize the number of successful connections, which has been shown to be NP-hard in [66]. Without loss of generality, the link scheduling problem over time-frame T with the precedence constraints is NP-hard by reducing from the Preemptive Scheduling problem, which is proven to be NP-hard in [31]. The Preemptive Scheduling is to schedule a set of P tasks - within a deadline $T \in \mathbb{Z}+$, where each task $y \in$ P is subdivided into sub-tasks $y_1, y_2, ..., y_n$ such that $\sum_{i=1}^{n} L(t_i) = L(t)$, where $L(t)$ is the length of task t, and the scheduling of t_i precedes t_{i+1}. For example, P is a set of V2V communication pairs where each pair can have a long routing path y. Each routing path y can have one or more links from y_1 to y_n to be allocated within RBs, representing the Preemptive Scheduling maps problem. Since our problem is a combination of multiple NP-hard sub-problems explained above, so without a doubt it is an NP-hard problem. To overcome the complexity of the MVGCF problem, in the next subsection, we propose a scalable algorithmic method.

3.4.2 The Proposed MVGCF Algorithm

The main concept behind the MVGCF method, described in Algorithm 1, is to maintain the priority queue PQ of communication pairs that must be scheduled in the system to achieve fairness (i.e., to maximize the number of communication packets for pairs that have transmitted the least. In other words, maximizing the minimum communication pair). The priority queue PQ is inherently sorted by the number of communications NC. Another central aspect of the method is to use a sorted communication pair list CPL sorted by the length of communication paths to maximize the V2V communications. The algorithm also makes use of a temporary list of completed communications LCC which is sorted by the communication path length and number of communications NC on insertion.

The input to the algorithm is the list of all communication pairs and their respective paths CPs. Each pair in CPs, consists of a source s and a destination d, as well as a path of transmission links which routes a packet from s to d.

The outputs of the algorithm are the total number of communications TNC, the fairness

F, and the resource block allocation RB. At initialization, the priority queue PQ and the list of completed communication pairs LCC are empty. $CP_{(s,d)}^{(i,j)}$ refers to the first link (i,j) of the communication path $CP_{(s,d)}$ from source s to destination d. The path of the communication pair $CP_{(s,d)}$ is pre-calculated from graph $\mathbb{G}(N,E)$. The total number of successful communications NC for all pairs (s,d) is zero. All communication pairs are also sorted and stored in CPL according to the length of their shortest communication path.

In line 2 of the algorithm, the time slot t iterates over the time frame T. For each time slot, we initialize the sets of transmitters (T_{ran}) and receivers (R_{ec}) to empty (see line 3). In line 4, when we start the algorithm at $t=0$, the priority queue PQ is empty, so lines 5-6 will not be executed. In subsequent time slots (i.e., $t>0$) when the priority queue PQ is not empty for each communication pair $CP_{(s,d)}$ in priority queue PQ, the algorithm scans all orthogonal channels (line 5) to allocate transmission link (i,j) from communication pair $CP_{(s,d)}$ on channel c. To do so, the algorithm calls function *Schedule* presented in Algorithm 2 to allocate the transmission link into the resource block RB_t^c, where t and c are respectively the time slot and channel. This function also checks whether a packet that belongs to a communication pair has already arrived at its destination and completed its communication or not. The explanation of Algorithm 2 about function *Schedule* will be given below. After evaluating any pairs in priority queue PQ, the algorithm moves to allocate transmissions from the list of communication pairs CPL. When transmission links for all communication pairs in PQ have been considered for scheduling, Algorithm 1 checks to allocate transmissions for communication pairs in list CPL (lines 7-9). To be noted that this list includes all the communication pairs when $t=0$. At the end of each time slot, when the list of communication pairs CPL is empty, the list of completed communication pairs LCC is copied into the list CPL to be considered for future scheduling in the next round, and then the LCC list is set to null (line 10). The algorithm, at the end, finds the communication pair that has transmitted the least number of packets in the system to calculate the fairness F (line 11) and sums the total number of communications TNC from all communication pairs counter $NC_{(s,d)}$ (line 12).

In Algorithm 2, the function *Schedule* takes as an input a link (i,j) which belongs to a communication pair $CP_{(s,d)}$, time slot t, and channel c. This function checks the half-duplex and SINR constraints and whether node j is the destination of the communication pair $CP_{(s,d)}$ (i.e., whether $j=d$). The half-duplex and SINR constraints are checked in line 2 of the algorithm, which first ensures that both nodes i and j are not in the sets of transmitters T_{ran} and receivers R_{ec}, and also makes sure that by allocating link (i,j) in the resource block RB_t^c, the newly added transmission will not cause harmful interference on the receivers of all other simultaneous transmissions which already have been allocated into the resource block RB_t^c; that is the SINR at all receivers (R_{ec}) as well as the new added one (i.e., j) is not below the threshold β. If both constraints are satisfied then transmission

Algorithm 1: Maximizing the total number of V2V communications while maintaining fairness (MVGCF)

Data: CPs
Result: TNC, F, RB
1 Initialize: $PQ = \varnothing, CP_{(s,d)}^{(i,j)} = firstLink(CP_{(s,d)})\ \forall (s,d) \in CPs, LCC = \varnothing, NC_{(s,d)} = 0\ \forall (s,d) \in CPs, CPL = sorted(CPs)$.
2 **for** $t = 0; t <= T; t++$ **do**
3 $T_{ran} = \varnothing$, $R_{ec} = \varnothing$;
4 **for** $CP_{(s,d)}$ *in* PQ **do**
5 **for** $c = 1; c <= C; c++$ **do**
6 Schedule($CP_{(s,d)}^{(i,j)}, t, c$)
7 **for** $CP_{(s,d)}$ *in* CPL **do**
8 **for** $c = 1; c <= C; c++$ **do**
9 Schedule($CP_{(s,d)}^{(i,j)}, t, c$)
10 $CPL = LCC$, $LCC = \varnothing$
11 $F = min(NC_{(s,d)})\ \forall (s,d) \in CPs$
12 $TNC = \sum_{\forall (s,d) \in CPs} NC_{(s,d)}$

link (i, j) will be allocated to the resource block RB_t^c, and nodes i and j respectively will be inserted into T_{ran} and R_{ec} (lines 3-4). In case the link (i, j) has been allocated into RB_t^c, the algorithm checks whether node j is the destination of the communication pair $CP_{(s,d)}$ (i.e., if $j = d$, line 5); if yes, then it means that all the links for the communication pair $CP_{(s,d)}$ have been scheduled. Hence, the algorithm in lines 6-8 increments the total number of successful packet transmissions of $NC_{(s,d)}$, reset the communication pair $CP_{(s,d)}$ to its first link on its path for the future scheduling, and adds it into the completed communication pair list LCC. If in line 5, node j is not the destination node of the communication pair $CP_{(s,d)}$ (i.e., $j \neq d$) however the transmission link (i, j) has been scheduled, then the next transmission link on the path of the communication pair $CP_{(s,d)}$ is considered for the future time slot scheduling and the communication pair is inserted into the priority queue PQ to be given the highest priority (lines 9-11). If link (i, j) in line 2 of the algorithm has not been scheduled in the first place, then the algorithm inserts the communication into the priority queue PQ to be given a high priority for the future time slot scheduling (line 12-13).

3.4.3 Time Complexity of the MVGCF Method

To obtain the time complexity of the MVGCF method, we first drive the time complexity of Algorithm 2, and then calculate the running time of Algorithm 1. For the function *Schedule* in Algorithm 2, the *if* statement in line 2 takes in the worst case $O(n^2)$, where in total there

Algorithm 2: Scheduling a Resource Block

1 **Function** Schedule($CP^{(i,j)}_{(s,d)}$, t, c):

2 **if** $i \notin \mathrm{T_{ran}}$ && $\mathrm{j} \notin \mathrm{T_{ran}}$ && $i \notin \mathrm{R_{ec}}$ && $\mathrm{j} \notin \mathrm{R_{ec}}$ && $\dfrac{P_{(u,h)}G_{(u,h)}}{\eta + \sum\limits_{\forall w \in T_{ran}} P_{(w,h)}G_{(w,h)}} \geq \beta$, $\forall u \in \{T_{ran} \cup i\}, \forall h \in \{R_{ec} \cup j\}$ **then**

3 Allocate (i, j) to RB^c_t

4 Insert i into T_{ran} and j into R_{ec}

5 **if** $j == d$ **then**

6 $NC_{(s,d)} = NC_{(s,d)} + 1$

7 $CP^{(i,j)}_{(s,d)} = \mathrm{firstLink}(CP_{(s,d)})$

8 Move $CP_{(s,d)}$ to sorted LCC

9 **else**

10 $CP^{(i,j)}_{(s,d)} = \mathrm{nextLink}(CP_{(s,d)})$

11 Move $CP_{(s,d)}$ to PQ if not already in PQ

12 **else**

13 Move $CP_{(s,d)}$ to PQ if not already in PQ

are $n/2$ transmitters and the same number of receivers. The statements in lines (3–4) take constant time $O(1)$. In lines (5-8) the *if* statement takes in the worst case $O(p)$, where p is the total number of communication pairs. Again, the statements in lines (9-11) take constant time $O(1)$. Similarly, the statement in line 13 takes constant time $O(1)$. Hence, the time complexity of the function in Algorithm 2 is $O(n^2+p)$. Since $n^2 > p$, then the time complexity of the algorithm can be reduced to $O(n^2)$. As for Algorithm 1, for each time slot, the *for* loop in line 4 contributes to the complexity of the method with $O(p)$ where p is the number of pairs in priority queue PQ. The third nested *for* loop in line 5 contributes with $O(C)$ because it checks all C orthogonal channels. Subsequently, the function in line 6 takes in the worst case $O(n^2)$ as explained earlier. Hence, the running time of lines (4-6) all together is $O(n^2)$. Similarly, the running time of lines (7-9) is $O(n^2)$. The time complexity of each of the statements in lines (11-12) is $O(p)$. Hence, the time complexity of Algorithm 1 is $O(T * ((p * C * n^2) + (p * C * n^2))) + p$, which can be simplified to $O(T * C * p * n^2)$.

Lemma 1. *Correctness of the Algorithms.*

Proof: In Algorithm 1, the main *for* loop in line 2 terminates when all the time slots are covered, similarly the two *for* loops in lines (5-8) for multiple channels. The two *for* loops in lines (4-7) respectively will terminate when the priority queue PQ and communication pair list CPL are empty. The only concern for the algorithm not to terminate is when more elements are inserted into these queues than removing them. Since there are no statements in Algorithm 1 indicates insertion of elements into queues except for CPL in line 10 which

is safe because it is outside the *for* loop. Hence the termination of Algorithm 1 depends on the execution of function *Schedule* in Algorithm 2. There is no *for* loop in Algorithm 2 that might get stuck into a loop and avoid termination of the algorithm. However, we need to check any insertion into priority queue PQ and the communication pair list CPL. There is no insertion into CPL in Algorithm 2, however when a link has not been scheduled due to the Half-Duplex or SINR, or the end of the path of a communication pair is not reached, Algorithm 2 inserts that communication pair into PQ to be considered and given a high priority for the future scheduling. However, any communication pairs in PQ will eventually be allocated in one of the available resource blocks and thus the termination of Algorithm 1.

3.5 The Introduced Reinforcement Learning and Hybrid Approaches

Reinforcement learning (RL) is an approach where an agent interacts with an environment to learn a policy that maximizes its long-term rewards. The agent takes actions from the given states in the environment, obtains feedback in the form of rewards, and uses the information to update its policy.

On the other hand, Deep reinforcement learning (DRL) takes this concept further by using neural networks to learn from data. DRL has proven to be effective in solving intricate decision-making problems and optimizing resource allocation. The agent updates its neural networks based on the rewards it receives and stores past experiences in a replay buffer. During training, batches of experiences are sampled from the replay buffer to update the neural network parameters [67].

In our method, we formulate our MVGCF problem as Markov Decision Process (MDP) to allocate resources into RBs within a time frame. An MDP is represented by a tuple (S, A, γ, P, R), where: S is a finite set of states, denoted as $s_t \in S$ at time slot t; A is an action space such that if we let $\mathcal{A}$ to be a set of all possible actions in state s_t and a_t is one of the actions at any time slot t, then $a_t \in A$; $\gamma \in [0, 1]$ is the discount factor, which determines the weight of future rewards in the decision-making process; P is a Markovian transition model, denoted as $P(s_{t+1} \| s_t, a_t)$, which represents the probability of transitioning from state s_t to state s_{t+1} when an action a_t is taken; R is the reward distribution, denoted as $P(r_t \| s_t, a_t)$, which gives the immediate reward $r_t \in R$ after an action a_t is taken in a state s_t at time slot t. The state, action, and reward functions under the MDP framework are given as follows:

3.5.0.1 Agent

Roadside Unit (RSU) is considered to be an agent.

3.5.0.2 State

Each state s_t is defined as a tuple of multiple vectors: i) a vector of the total number of communications for different V2V pairs at time slot t, $nc_t = \{nc_t^1, nc_t^2, ..., nc_t^M\}$; ii) a vector of the current packet position for different V2V pairs in using a multi-hop path at time slot t, $cpp_t = \{cpp_t^1, cpp_t^2, ..., cpp_t^M\}$; iii) a vector of the number of hops in the shortest path for different V2V pairs at time slot t, $mh_t = \{mh_t^1, mh_t^2, ..., mh_t^M\}$; and respectively vectors of the iv) transmitters and v) receivers of all V2V pairs at time slot t, $tr_t = \{tr_t^1, tr_t^2, ..., tr_t^M\}$ and $rec_t = \{rec_t^1, rec_t^2, ..., rec_t^M\}$. Thus the system state s at time slot t can be expressed as:

$$s_t = (nc_t, cpp_t, mh_t, tr_t, rec_t). \tag{3.16}$$

3.5.0.3 Action

Each action a_t is defined as a tuple of multiple vectors: i) a vector used to assign V2V communication pairs to transmit at time slot t, $cp_t = \{cp_t^1, cp_t^2, ..., cp_t^M\}$; ii) a vector of scheduled transmission links for different V2V pairs at time slot t, $tp_t = \{tp_t^1, tp_t^2, ..., tp_t^M\}$; iii) a vector of considered channels for different V2V pairs at time slot t, $c_t = \{c_t^1, c_t^2, ..., c_t^M\}$; and iv) a vector of transmission power levels for different V2V pairs at time slot t, $p_t = \{p_t^1, p_t^2, ..., p_t^M\}$. The system action a at time slot t can be expressed as:

$$a_t = (cp_t, tp_t, c_t, p_t). \tag{3.17}$$

3.5.0.4 Reward

We use a reward function to provide feedback on each action a_t taken in a given state s_t to the RL agent. The agent selects an action a_t from a set of possible actions $\mathcal{A}_t$ at time slot t, where $\mathcal{A}$ represents the available resource allocation choices. Let r_t be the immediate reward at each time slot t. The reward function $r_t(s_t, a_t)$ at time slot t can be expressed as follows:

$$r_t(s_t, a_t) = \begin{cases} 0, & \text{if } \sum_{nc_{t+1}} = \sum_{nc_t}. \\ 1, & \text{if } \sum_{nc_{t+1}} = \sum_{nc_t} + 1. \\ B, & \text{if } \min(nc_{t+1}) > \min(nc_t). \end{cases} \tag{3.18}$$

$r_t(s_t, a_t) = 0$, if a communication from a source to a destination is not completed; 1, if a communication from a source to a destination is completed, but note that here the reward is considered when fairness is not achieved; whereas $r_t(s_t, a_t) = B$ (a very big reward), if

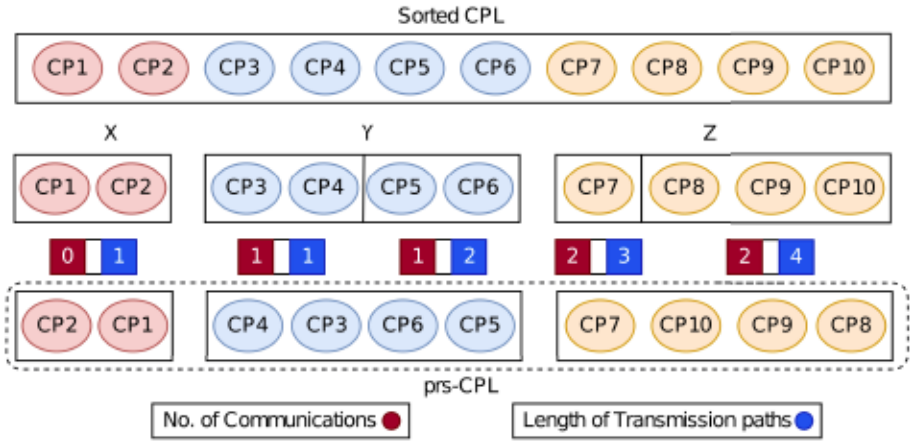


Figure 3.2: The generation process of pseudo-random sorted List $prsCPL$.

a communication from a source to a destination is completed and fairness is achieved by completing a communication for a pair with the least number of V2V packet transmissions.

3.5.0.5 Initial MVGCF Solution

We start by generating our initial sorted communication pairs list CPL, as described in [68]. This list is sorted based on the communication path length and the number of communications upon insertion. Fig. 3.2 illustrates three distinct groups, labeled as X, Y, and Z. Each group consists of a list of communication pairs, where the red color represents the number of communications and the blue color indicates the communication path length. It is worth noting that the values of individual communication pairs in group Y are lower than those in group Z but higher than those in group X, both in terms of the number of communications and communication path length. Group X contains only one subgroup, since communication pair 1 (i.e., CP1) and communication pair 2 (i.e., CP2) exhibit a similar number of communications and communication path lengths. On the other hand, groups Y and Z have two different subgroups due to their varying communication path lengths, as shown in blue color. To ensure randomness, we then shuffle the communication pairs within each subgroup and construct a new List called pseudo-random sorted $prsCPL$, maintaining the order of initial communication pairs.

Algorithm 3 takes a sorted list of communication pairs as an input and generates a sequence of states, actions, and rewards, stored in the $SARlist$ list. Initially, the $SARlist$ is empty (line 1). The algorithm then iterates over all episodes, starting from $k = 1$ (line 2). In each episode, a pseudo-random sorted list $prsCPL$ is generated from the given sorted CPL list (line 3). The $prsCPL$ list is used as an input for the MVGCF method, which runs on this list to obtain a sequence of states (S), actions (A), and rewards (R) (line 5). Finally, the obtained sequences of S, A, and R are stored in the initialized $SARlist$ (line 6).

Algorithm 3: Initial MVGCF Solution

Data: Sorted CPL
Result: $SARlist$
1 Initialize: $SARlist = \varnothing$
2 **for** $k \leftarrow 1 : K$ **do**
3 Generates a pseudo-random sorted list $prsCPL$.
4 Execute the MVGCF method using $prsCPL$ list as an input.
5 Observe the MVGCF method execution to collect a sequence of states (S), actions (A), and rewards (R).
6 Store the sequences (S, A, R) into the $SARlist$.

3.5.1 The Qlearning Approach

In our approach, we use a Qlearning network [67] that utilizes an off-policy method and runs for 75,000 episodes to allocate resources into RBs within a time frame T. Here, the input to Algorithm 4 is an interface of the environment. The outputs of the algorithm are the total number of communications TNC, the fairness F, and the resource block allocation RB. At initialization, the Q table $Q(s, a)$ consisting of states and actions is empty. In line 2 of the algorithm, episode k iterates across all episodes, K. Afterward, line 3 iterates over the time slots as long as the terminal state s_T is not reached. At each time slot, we initialize the sets of transmitters (T_{ran}) and receivers (R_{ec}) to empty (see line 4). In line 5, when we start the algorithm at $t = 1$, we observe a set of state components, consisting of the number of communications (nc_t), current packet position (cpp_t), min-hops (mh_t), transmitters (tr_t) and receivers (rec_t) from the environment. In line 6, we choose an action a_t from a list of possible actions $\mathcal{A}_t$ in a given state s_t utilizing an ϵ *Greedy* policy. We then allocate a_t into a resource block ($RB_{t,c}$). In line 8, after choosing an action a_t, we receive a reward (r_t) from the environment, and then we update the $Q(s_t, a_t)$ value in the Q-table (see line 9).

In line 9 of the algorithm, α represents the learning rate (which is 0.0001), and γ signifies the discount rate applied to future rewards (set to 0.99). The Q-value for action a_t in the current state s_t is updated by adding the existing value $Q(s_t, a_t)$ which determines the best action in the current state s_t. Qlearning continuously updates the Q-value for each state s_t based on a policy and transitions to the next state using the equation given in line 9. This process is repeated multiple times until the overall Q-value converges. The algorithm, when it reaches the final state, it executes the trained agent in the environment (line 10), determines the fairness F by finding the communication pair with the least number of packet transmissions in the entire time frame T, and sums the total number of communications TNC (line 11).

Algorithm 4: Proposed Qlearning-based Solution

Data: Environment Interface
Result: TNC, F, RB
Initialize: $Q(s,a) = \varnothing$
for $k \leftarrow 1 : K$ **do**
  for $t \leftarrow 1 : T$ *and* $s_t \neq s_T$ **do**
    $T_{ran} = \varnothing$, $R_{ec} = \varnothing$.
    Observe s_t $(nc_t, cpp_t, mh_t, tr_t, rec_t)$ from the environment.
    Choose a_t (cp_t, tp_t, c_t, p_t) from a list of possible actions $\mathcal{A}_t$ using an ϵ Greedy policy and allocate a_t into $RB_{t,c}$.
    Receive a Reward (r_t) from the environment.
    Update $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_{\mathcal{A}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$.
Execute trained $DDQN$ agent on environment interface.
Given terminal state $s_T, F = min(nc_T)$ *and* $TNC = \sum_{p=1}^{M} nc_T^p$.

MVGCF-Qlearning

The conventional Qlearning follows an exploration and exploitation approach to gather information about state-action pairs until it converges. The sizes of state and action spaces can be calculated as follows:

$$S = O(nc \times cpp \times mh \times tr \times rec) \tag{3.19}$$

$$A = O(cp \times tp \times c \times p). \tag{3.20}$$

In fact, since the number of options for s_t and a_t is extremely high, exploring the entire S and A spaces becomes challenging and computationally expensive, leading to poor performance and slow convergence. To handle this issue, we propose MVGCF-Qlearning, a heuristic-based reinforcement learning method, where we first find the initial solutions by varying a randomly sorted CPL list in Algorithm 3. We initially run this method for 10,000 episodes and update the Q-table with the set of states and actions observed from the output of the heuristic method. These state-action samples guide the exploration to a high-quality state-action subspace. By integrating the heuristic MVGCF insights into the learning process, the MVGCF-Qlearning agent accelerates its learning by understanding which actions in specific states yield the best rewards. This approach delegates the model to enhance the quality of solutions obtained to converge faster toward optimal solutions.

Table 3.2: SIMULATION PARAMETERS

Parameters	Values
Activation Functions	ReLU
Number of Neurons	256, 128, 64, 32, 16
Number of Hidden Layers	8
Learning Rate	0.0001
Optimizer	Adam
Total Number of Episodes	225
Decay Rate	1/Episodes * 2
Discounted Reward	0.99

3.5.2 The Double Deep Q-Networks (DDQN) Approach

The traditional DDQN [67], as shown in Fig. 3.3 and Fig. 3.4, is an extension of the DQN algorithm of DRL which runs multiple episodes to allocate resources into RBs within a time frame T. The used notations are listed in Table 3.2. Here, the input and outputs to Algorithm 5 are similar to Algorithm 4. At initialization, the primary network Q is initialized with random weights, while the target network $\hat{Q}$ is initialized by directly copying the weights from the primary network Q.

All the steps in lines 2-7 are similar to Algorithm 4 which have been explained earlier. In line 8, after choosing an action a_t, we receive a reward r_t for moving to the next state s_{t+1} which includes a list of next possible actions $\mathcal{A}_{t+1}$ from the environment. Line 9 stores those values obtained at time slot t as one single transition $(s, a, r, s', \mathcal{A})$ in the replay memory (RM). In line 10, a random minibatch of transitions $(s_j, a_j, r_j, s'_j, \mathcal{A}_j)$ is sampled from the replay memory RM. For each transition in the minibatch (line 11), the target Q-value $Q_t(s_j, a_j)$ is computed using the target network $\hat{Q}$. The $Q_t(s_j, a_j)$ is calculated as the sum of the immediate reward r_j and the discounted maximum Q-value $\hat{Q}(s'_j, \mathcal{A}_j)$ from the next state s'_j using the primary network Q (line 13). We then perform gradient descent on the difference between the target Q-value $Q_t(s_j, a_j)$ and the primary network Q (line 14). Next, target Q-network weights $(\hat{\theta})$ are updated. The update process involves adjusting the target network parameter θ' towards the primary network parameter θ, where the rate of averaging value τ is typically set to 0.01 (line 15). In this way, target network weights $(\hat{\theta})$ are updated by copying the weights from the primary network. Once the termination condition is met, the algorithm proceeds to the next episode until all episodes are completed. In the end, similar to Algorithm 1, the final agent is executed in the environment (line 16), and the total number of successful communications TNC and fairness F are calculated (line 17).

MVGCF-DDQN

When the conventional DDQN has very large state s_t and action a_t spaces (refer to equations (4.32) and (4.33)), exploring the entire state and action spaces becomes challenging

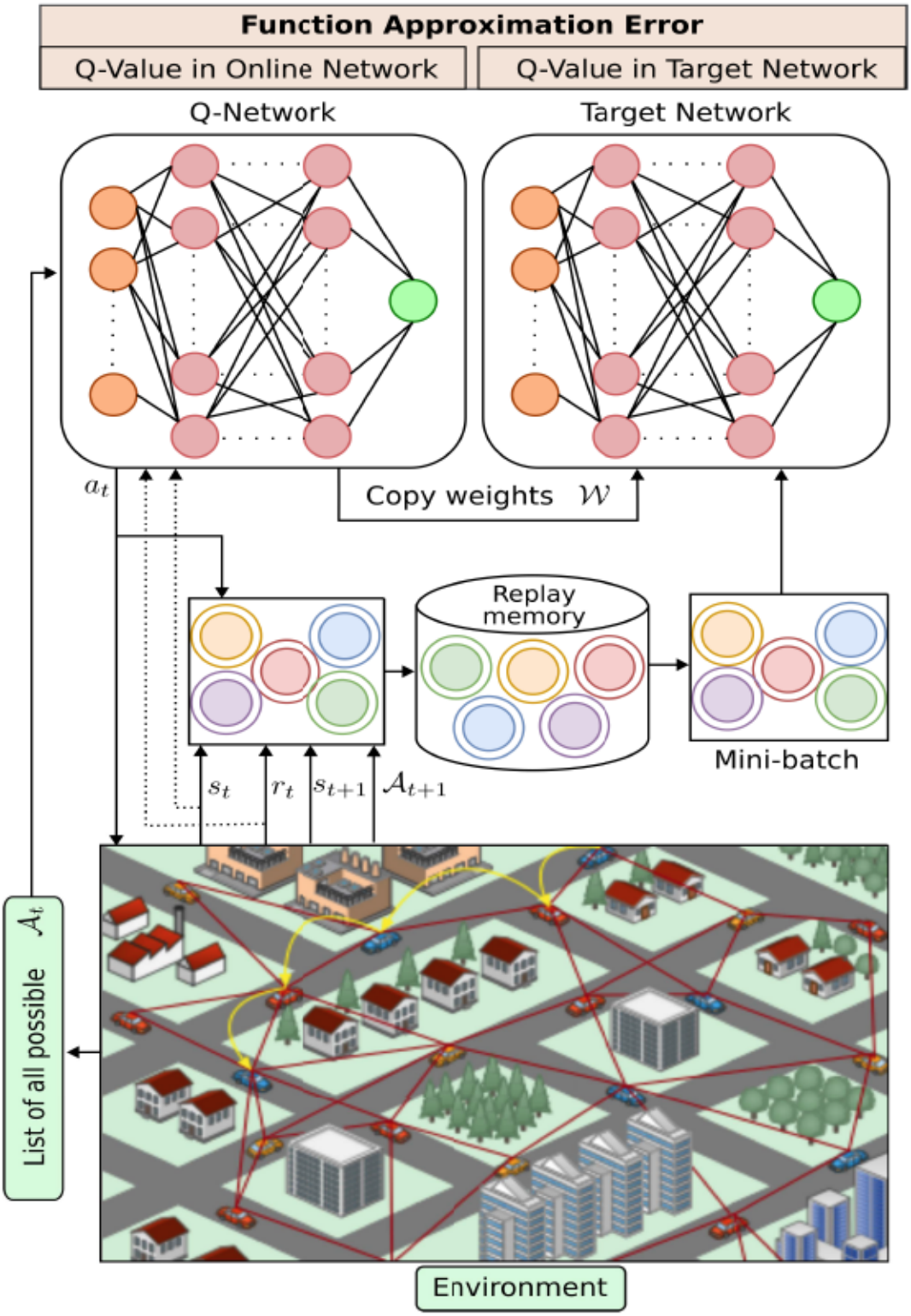


Figure 3.3: The proposed DRL approach to obtain the reward policy.

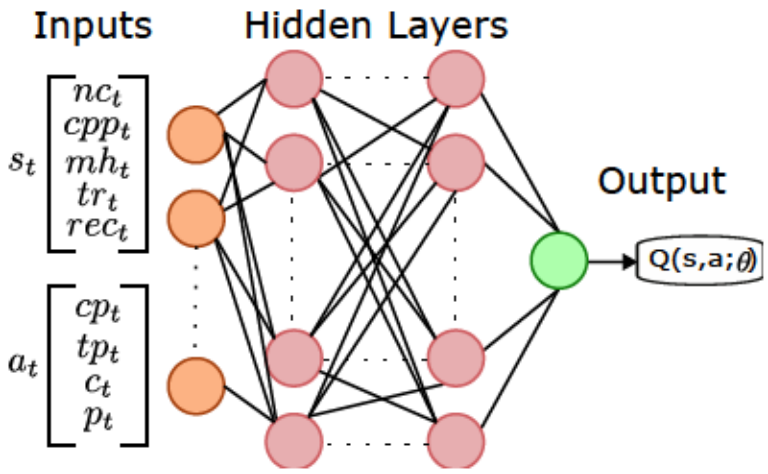


Figure 3.4: The proposed Q-Network.

and computationally expensive, which will lead to poor performance and slow learning convergence. To address such an issue, we introduce another heuristic-based DRL method called MVGCF-DDQN. We first find solutions for the MVGCF problem using the heuristic method proposed in [68] for 10,000 episodes. By leveraging the heuristic MVGCF method, we sequentially populate the replay memory (RM) with a subset of promising states, actions, rewards, next states, and next actions. This targeted approach reduces the amount of time spent on exploration, allowing the agent to focus on learning from these high-quality states. Hence, the MVGCF-DDQN agent explores and exploits the state-action space more effectively, since it understands which actions in specific states yield the best rewards, and accelerates learning convergence by leveraging prior knowledge.

3.6 Performance Evaluation

To evaluate the performance of the proposed methods, in this section, we first compare the Random (the explanations of the method are explained below), the heuristic OAMM, Qle arning, DDQN, the hybrid OAMM-Qlearn., and the hybrid OAMM-DDQN with the optimal solutions obtained from the optimization model on small networks. Then, we consider all of them except the optimization model on both medium and large instances. The performance of these methods is compared with respect to the number of V2V communication packets and Fairness.

Fig. 3.5(a) and Fig. 3.5(b) illustrate an example of the considered V2V networks, where the number of groups is set to be half the total number of V2V nodes. Here, each color represents a group of vehicles, where if two colors are similar to each other then it means that they belong to a similar vehicle's group; otherwise, they are just acting as a relay node to send the received packets to its destination node.

Algorithm 5: Proposed Double Deep Q networks-based solution

Data: Environment Interface
Result: TNC, F, RB

1 Initialize: $Q \leftarrow$ RandomWeights(), $\hat{Q} \leftarrow Q$
2 **for** $k \leftarrow 1 : K$ **do**
3 **for** $t \leftarrow 1 : T$ *and* $s_t \neq s_T$ **do**
4 $T_{ran} = \varnothing$, $R_{ec} = \varnothing$.
5 Observe s_t $(nc_t, cpp_t, mh_t, tr_t, rec_t)$ from the environment.
6 Choose a_t (cp_t, tp_t, c_t, p_t) from a list of possible actions $\mathcal{A}_t$ using an ϵ Greedy policy and allocate a_t into $RB_{t,c}$.
7 Obtain action a_t, reward r_t, next state s_{t+1}, and next action $\mathcal{A}_{t+1}$.
8 Store $(s_t, a_t, r_t, s'_{t+1}, \mathcal{A}_{t+1})$ as one transition in RM.
9 Sample random minibatch of transitions $(s, a, r, s', \mathcal{A})$ from RM.
10 **for** *Each transition* $(s_j, a_j, r_j, s'_j, \mathcal{A}_j)$ *in minibatch* **do**
11 Compute target Q value using $\hat{Q}$ network:
12 $Q_t(s_j, a_j) \leftarrow r_j + \gamma \cdot Q(s'_j, \arg\max_{\mathcal{A}} \hat{Q}(s'_j, \mathcal{A}))$
13 Perform gradient descent step on: $(Q_t(s_j, a_j) - Q(s_j, a_j))^2$
14 Update target network weights: $\hat{\theta} \leftarrow \tau \cdot \theta + (1 - \tau) \cdot \hat{\theta}$.
15 Execute trained $DDQN$ agent on environment interface.
16 Given terminal state s_T, $F = min(nc_T)$ *and* $TNC = \sum_{p=1}^{M} nc_T^p$

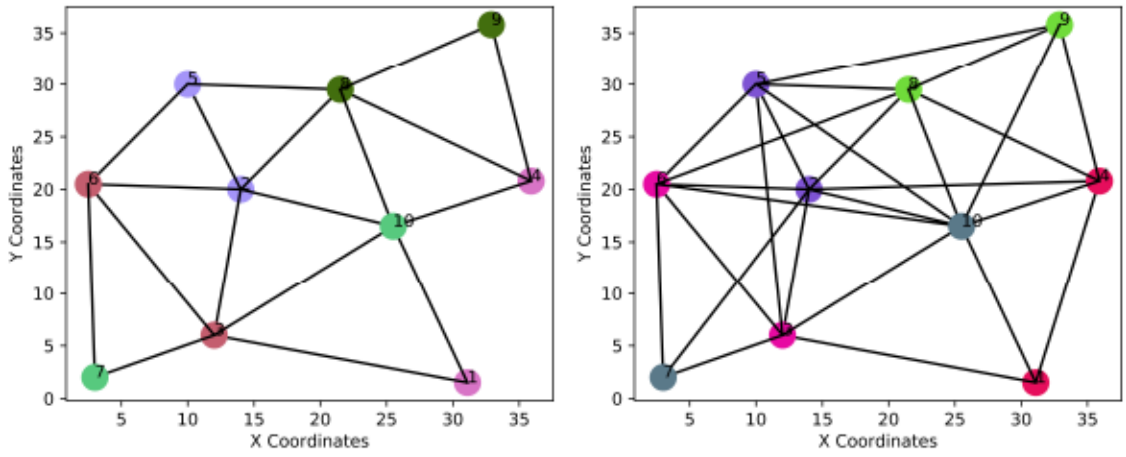
Random Method

The inputs and outputs of the Random algorithm are similar to the MVGCF method, where the variables are considered to be the same in both methods. However, instead of sorting all communication pairs in a list, they are randomly stored. The main differences between the MVGCF and the Random method are that line 1 of Algorithm 1 do not execute the sorting instructions, and lines 6 and 11 of the *Schedule* function inserts into a not sorted LCC.

For wireless communications, the background noise is considered as η = -111 dBm/Hz, the power decay as $\alpha = 2.5$, the threshold as $\beta = 5$ dB, and the maximum and minimum transmission powers as $P_{MAX} = 20$ dB and $P_{MIN} = 0$ dB, respectively [69]. We then use CPLEX to solve our optimization model and Python3 to simulate the operation of our algorithms. We run our program on Intel(R) Xeon(R) CPU E5-2637 v4 @ 3.50GHz (2 processors) and 64.0 GB memory. The results are averaged over five runs.

3.6.1 Evaluation Over Small Networks

In this subsection, the performance of different methods (Random, MVGCF, Qlearning (Qlearn.), DDQN, MVGCF-Qlearning (MVGCF-Qlearn.), MVGCF-DDQN, MILP-based



(a) (b)

Figure 3.5: Examples of medium networks while considering the V2V Nodes to 10 by varying network density to: (a) 0.4, and (b) 0.6.

solution: Optimum) is evaluated over the total number of successful communications and achieved fairness by varying other parameters, such as number of V2V nodes, time frame (number of time slots), number of channels, and maximum data communication range. For all evaluations, we fix the parameters unless otherwise stated: we set the number of nodes to 4 and 6, time slots to 10, number of channels to 2, and the V2V communication range to 12 unit distance, which causes the total number of links in the network.

Fig. 3.6(a) and Fig. 3.7(a) respectively provide a visual representation of the total number of communications and fairness by varying the number of nodes. Here, the number of groups is set to be half the number of nodes. The figures reveal that when considering two nodes, all methods achieve a similar number of communications (i.e., 10) and an equal level of fairness (i.e., $F = 5$), whereas with the Random method, we achieve lower fairness (i.e., $F = 4$) due to not scheduling all communication pairs in a sequence to maintain fairness. However, as the number of V2V nodes increases, a consistent increase in the number of communications is observed across all methods. Nonetheless, when the total number of V2V nodes reaches 6, a slight decrease in fairness is observed specifically for MVGCF and Qlearning. This is attributed to the increasing number of routing paths with the rise of V2V nodes, which poses challenges in maintaining sequential scheduling of communication pairs to ensure fairness. However, the Optimum and hybrid heuristic-based RL methods consistently demonstrate performance stability, maintaining a fairness level of $F = 5$ throughout the experiments.

The obtained results presented in Fig. 3.6(b) and Fig. 3.7(b) illustrate the impact of varying the number of time slots on the total number of communications and fairness in small networks. As plotted in the figures, the total number of communication packets and

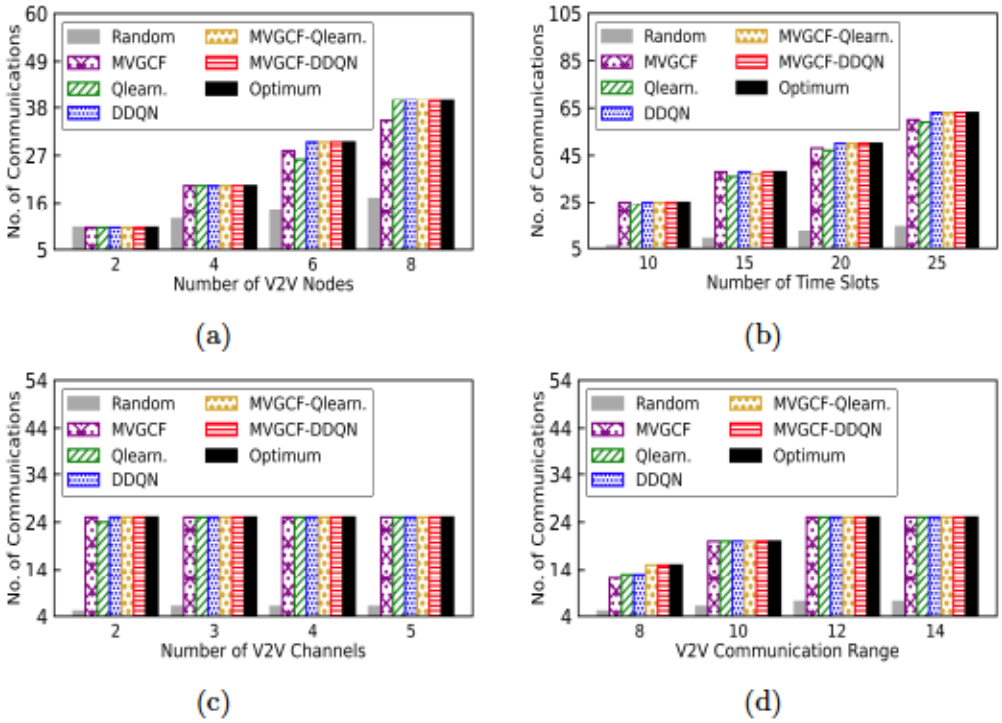


Figure 3.6: Total number of communications as a comparison metric for different methods (Random, MVGCF, Qlearn., DDQN, MVGCF-Qlearn., MVGCF-DDQN, and MILP-based solution: Optimum solution) by varying (a) number of V2V nodes, (b) number of time slots, (c) number of V2V channels, and (d) V2V communication range.

fairness as expected increases with the number of time slots because there is more time to allocate packets into resource blocks RBs. Notably, when the number of time slots is set to 10, the MVGCF, DDQN, MVGCF-Qlearning, MVGCF-DDQN, and Optimum methods yield similar outcomes with 25 communications and achieve an equivalent level of fairness (5). In contrast, the Random and Qlearning method exhibits a relatively lower number of communications and fairness. However, as the number of time slots increases, the DDQN, MVGCF-Qlearning, MVGCF-DDQN, and Optimum methods continue to generate similar outcomes. On the other hand, the Random, MVGCF, and Qlearning methods show a slight decrease in fairness over time. It is noteworthy that the MVGCF method outperforms the Qlearning method and attains a better fairness due to the large size of the Q-table in the Qlearning method and also because there wasn't enough time to train the agent. However, the Random method consistently shows poor performance in terms of communications and fairness throughout the experiments.

Fig. 3.6(c) and Fig. 3.7(c) respectively show the total number of communications and fairness by varying the number of channels. The results in the figures indicate that all methods outperform Qlearning in terms of the total number of communications and fairness

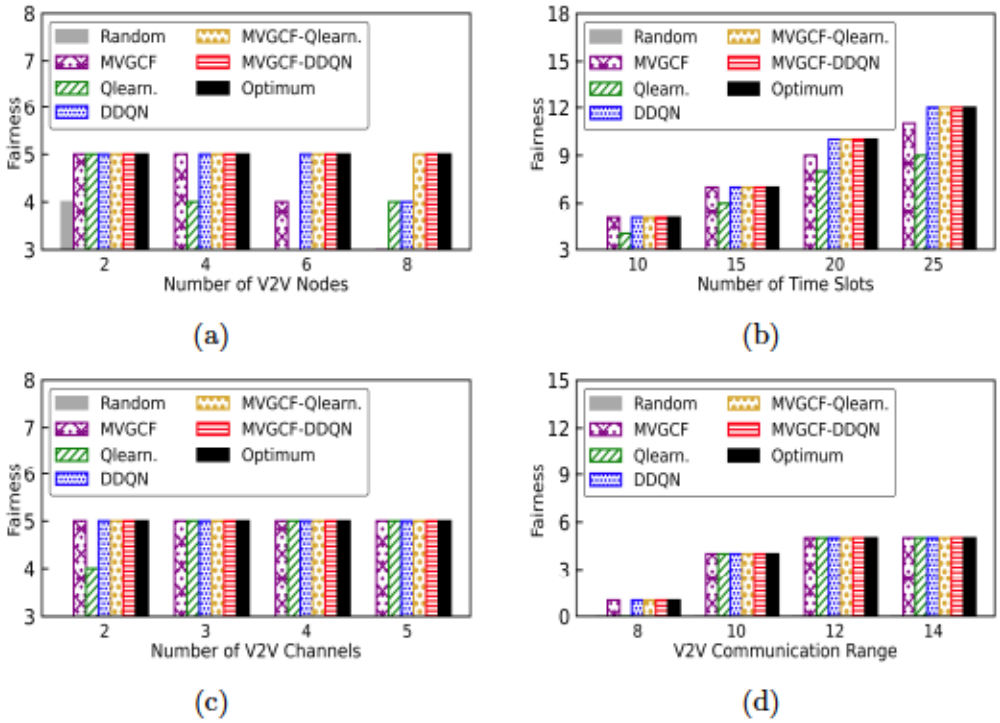


Figure 3.7: Results of fairness (total number of successful communications rounds for all pairs) for all methods by varying (a) number of V2V nodes, (b) number of time slots, (c) number of V2V channels, and (d) V2V communication range.

when the number of channels is 2; however, a significantly lower number of communications and fairness is observed for the Random method. However, with 3 channels, an equal number of communications (i.e., 25) and fairness (i.e., 5) is achieved by both the Optimum and MVGCF methods, but the Random method fails to maintain fairness. The reason is because there are plenty of orthogonal RBs available to allocate simultaneous transmissions (links), and the agent requires less training and it is easier to schedule transmissions. The number of communications remains unchanged when 4 to 5 channels are available, as no possible packets can be allocated into resource blocks.

Fig. 3.6(d) and Fig. 3.7(d) respectively show the total number of communications and fairness by varying the V2V communication range. As plotted in the figures, the total number of communications and fairness increases with the data communication range due to the fact that there are more available links to be allocated into resource blocks. Notably, the Optimum, MVGCF-Qlearning and MVGCF-DDQN method demonstrate superior performance in terms of the total number of communications, achieving 15 successful communications when the V2V communication range is 8 compared to 12 or less number of communications for other methods. However, with regard to fairness, all methods result in similar values for different communication ranges, except for the Qlearning method

when the V2V communication range is 8. Besides, the Random method does not achieve fairness. Furthermore, it is worth noting that when the vehicle's communication range exceeds 12 unit distance, the total number of communications and fairness do not increase because the growth in the communication range does not increase the total number of links (transmissions) in the network.

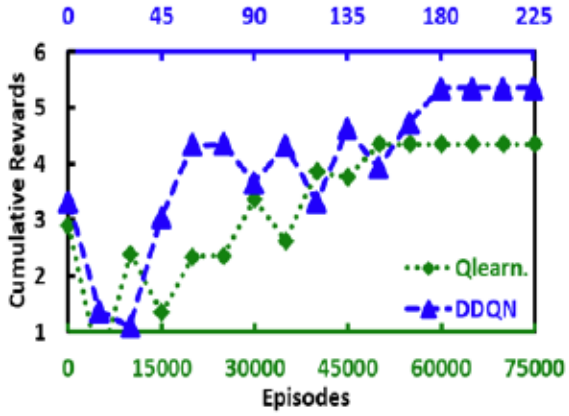


Figure 3.8: Learning curves of Reinforcement Learning algorithms: Qlearn. vs DDQN.

To show how the conventional Qlearning and DDQN agents learn and converge, in Fig. 3.8 we illustrate the learning curves of both methods. The x-axis describes the episode number, while the y-axis shows the cumulative rewards. The results were obtained using a network with 8 nodes with 80% link connectivity, using a time frame divided into 10 equal time slots, and with 2 channels. At the beginning of the episode, the Qlearning method begins with a cumulative reward of around 2.9, while the DDQN method starts with a higher cumulative reward of approximate 3.32. This inequality arises because the DDQN method uses neural networks to train a batch of samples, which yields a more promising initial result. As both methods sustain to explore the environment further, thus accumulate more rewards over time. However, the DDQN method displays faster learning compared to the Qlearning method; it rapidly adjusts to the environment, achieving higher cumulative rewards at earlier episodes. As the learning curves progress, the performance gap between both methods becomes more pronounced. By the end of the training, specifically at episode 225 for the DDQN and at episode 75,000 for the Qlearning method, the DDQN method outperforms the Qlearning method with a cumulative reward gap of around 1; the DDQN method receives a cumulative reward of 5.35, while the Qlearning method arrives to a cumulative reward of 4.35.

Fig. 3.9 shows the CPU run time for both Optimum and MVGCF by varying the number of V2V nodes from 2 to 10. By increasing the number of V2V nodes, the optimization model requires more time to execute, and the processing time grows exponentially, while the execution time of the heuristic method is in milliseconds. The Optimum model failed

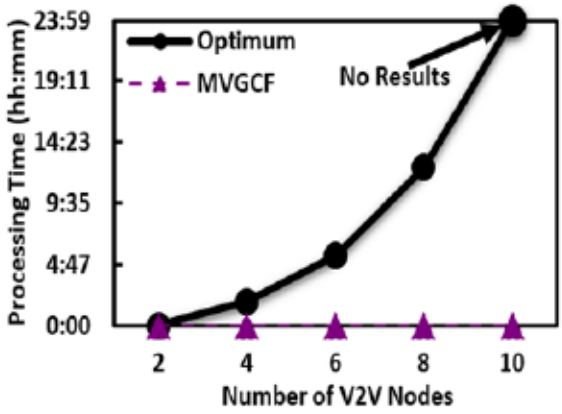


Figure 3.9: Computation time of optimization model (Optimum) vs our proposed heuristic method (MVGCF).

to obtain results for 10 or more nodes. Hence, the optimization model is not scalable or good for large networks. Therefore, in the next subsection, we evaluate the performance of our proposed methods without the optimization model using medium and large instances.

3.6.2 Evaluation Over Medium Networks

In this subsection, the performance of different methods (Random, MVGCF, MVGCF-Qlearn., and MVGCF-DDQN) is evaluated over the total number of successful communications and achieved fairness by varying different parameters, such as the number of V2V nodes, time frame (number of time slots), number of channels, and the density of links in the network. For all evaluations, we fix the parameters unless otherwise stated: we set the number of nodes to 10, time slots to 30, the number of channels to 4, and the density of links in the network to 90% (to be noted that a 100% density means a fully connected network). It should be noted that, for medium instances, we couldn't obtain results for conventional RL methods like Qlearning, and DDQN because they require a very long training time which we couldn't afford.

Fig. 3.10(a) and Fig. 3.11(a) respectively provide insights into the total number of communications and fairness metrics when the number of nodes is varied. Here, the number of groups is set to be half of the total number of nodes. As depicted in the figures, when considering 10 nodes, both the MVGCF-Qlearn. and MVGCF-DDQN methods achieve a similar number of communications (i.e., 149) and equal level of fairness (i.e., $F = 14$). In contrast, the MVGCF method yields a slightly lower number of communications and noticeably very less fairness primarily due to the lack of sequence scheduling of transmissions for communication pairs. The figures also show that as the number of V2V nodes increases, an upward trend in the number of communications and a downward trend in fairness are observed across all methods, where both the MVGCF-Qlearn. and MVGCF-DDQN meth-

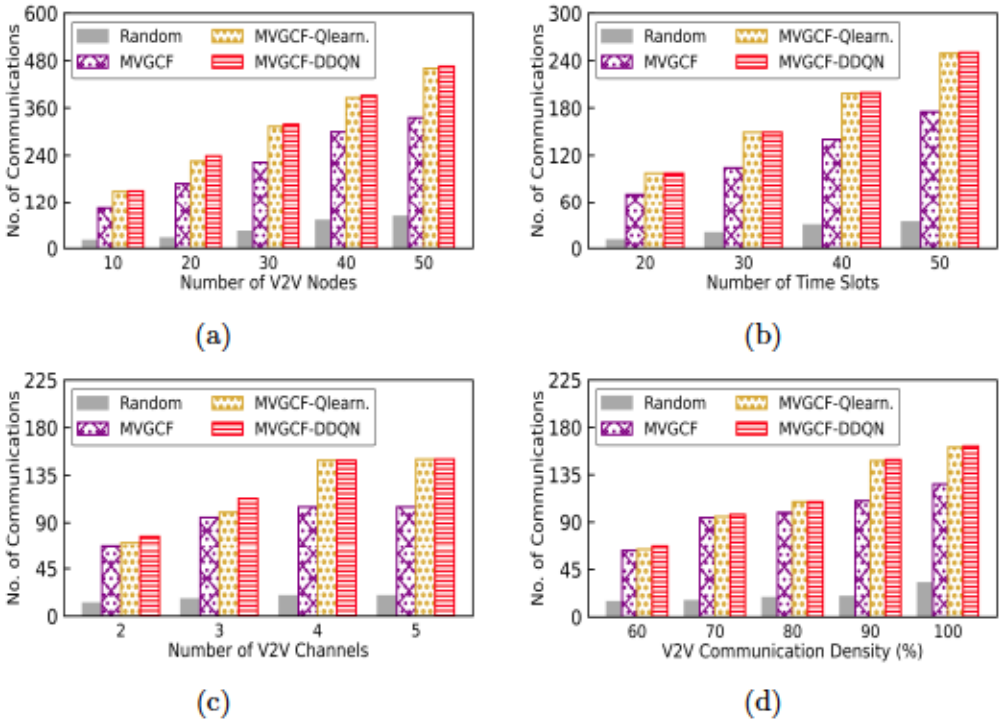


Figure 3.10: Total number of communications as a comparison metric for different methods (Random, MVGCF, MVGCF-Qlearn., and MVGCF-DDQN) by varying (a) number of V2V nodes, (b) number of time slots, (c) number of V2V channels, and (d) V2V communication range.

ods generate an almost similar result, a slight decrement in fairness for the MVGCF-Qlearn. method, a rapid decrement for the Random method is observed because of the rising number of routing paths with the increased V2V nodes. However, the MVGCF-DDQN method shows improved performance compared to the MVGCF-Qlearn. method with a fairness of $F = 7$.

With an increase in the number of time slots, there is an expected increase in the total number of communications and fairness among all methods, as shown in Fig. 3.10(b) and Fig. 3.11(b). When there are 20 time slots, both the MVGCF-Qlearn. and MVGCF-DDQN methods achieve a similar number of communications (i.e., 97) and equal fairness (i.e., $F = 9$). In contrast, the MVGCF method achieves fewer communications (i.e., 72) and fairness ($F = 3$). As the number of time slots increases, both the MVGCF-Qlearning and MVGCF-DDQN methods continue to perform similarly. However, a slight decrease in fairness is observed for the MVGCF-Qlearning method when there are 40 to 50 time slots. On the other hand, the MVGCF method exhibits poor performance compared to others, reaching a fairness value of eight (i.e., $F = 8$) when there are 50 time slots. Additionally, the Random method consistently shows a worse result compared to others and ended up

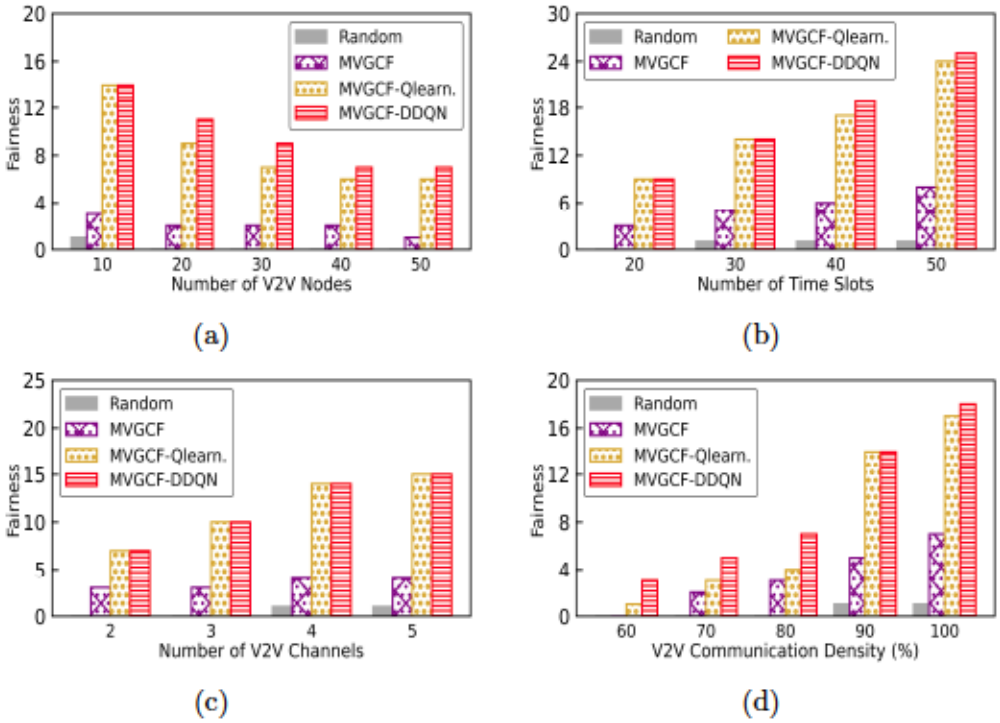


Figure 3.11: Results of fairness (total number of successful communications rounds for all pairs) for all methods by varying (a) number of V2V nodes, (b) number of time slots, (c) number of V2V channels, and (d) V2V communication range.

with a fairness equal to one ($F = 1$) when dealing with 50 time slots.

Fig. 3.10(c) and Fig. 3.11(c) illustrate the impact of varying the number of channels on the total number of communications and fairness. The MVGCF-DDQN method outperforms the MVGCF-Qlearn. method in terms of both the total number of communications and fairness when 2 or 3 channels are utilized. In contrast, the MVGCF method exhibits significantly lower values in terms of the number of communications and fairness. When there are 4 or 5 channels, both the MVGCF-Qlearn. and MVGCF-DDQN methods achieve the same number of communications (i.e., 150) and fairness (i.e., $F = 14$ when $c = 4$ and $F = 15$ when $c = 5$). This improvement can be attributed to the increased availability of RBs for transmitting data. However, there is a slight improvement in the number of successful communications and fairness for the Random and MVGCF methods.

Fig. 3.10(d) and Fig. 3.11(d) illustrate the performance of the proposed methods as the density of the network connection varies. It can be observed that the total number of communications and fairness increase as the network density becomes denser. The reason is because there are more links available to be allocated into RBs. As depicted in the figures, the total number of communications and fairness of all methods increase while dealing with denser networks. The reason is that there are more links to be allocated into RBs. For

example, the fairness score of the MVGCF method starts with zero at 60% network density and gradually increases until it reaches $F = 7$ when the network density is 100%. while the MVGCF method initially starts with a fairness score of 0, it demonstrates improvement in fairness with a score of $F = 7$ when the network density reaches 100 %. Similarly, the Random method starts with a F score of 0, it eventually shows an improvement in the fairness score with $F = 1$ when the network density is 100 %. However, despite this notable improvement, the Random and MVGCF methods fall behind the RL methods in terms of both the number of communications and fairness.

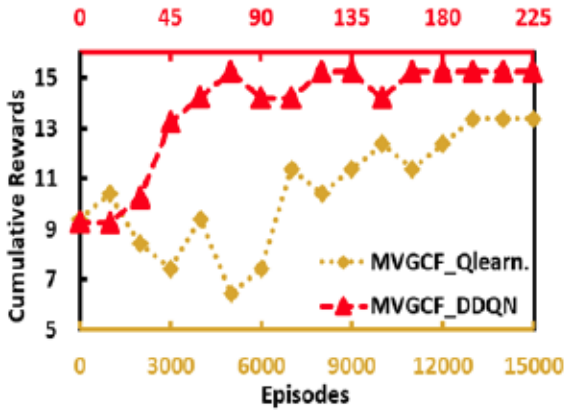


Figure 3.12: Learning curves of Reinforcement Learning algorithms: MVGCF-Qlearn. vs MVGCF-DDQN.

Fig. 3.12 shows the learning curve of the hybrid heuristic-based RL methods: MVGCF-Qlearn. and MVGCF-DDQN. The x-axis represents the episode number, while the y-axis represents the cumulative rewards. The results were received using a network with 10 nodes with 90% link connectivity, using a time frame divided into 30 equal time slots, and with 4 channels. At the beginning, both methods begin with 9 cumulative rewards in the first episode because of the result obtained from the heuristic MVGCF method, and gradually accumulate rewards as they explore the environment and acquire more knowledge. However, the MVGCF-DDQN approach shows faster learning compared to the MVGCF-Qlearning approach; it rapidly adapts to the environment, obtaining higher cumulative rewards at earlier episodes. As the learning curves progress, the performance gap between the two methods becomes more visible. By the end of training, particularly at episode 225 and 15000 for the MVGCF-DDQN and MVGCF-Qlearning methods respectively, the MVGCF-DDQN method outperforms the MVGCF-Qlearning method with a cumulative reward gap of 2; the MVGCF-DDQN method obtains a cumulative reward of 15, while the MVGCF-Qlearning method reaches a cumulative reward of 13.

3.6.3 Evaluation Over Large Networks

In this subsection, the performance of different heuristic methods (Random, and MVGCF) is evaluated over the total number of successful communications and achieved fairness by varying parameters such as number of V2V nodes, time frame (number of time slots), number of channels, and the density of communication links. For all evaluations, we fix the parameters unless otherwise stated: we set the number of nodes to 100, number of time slots to 7000, number of channels to 4, and the density of links in the network to 80% (to be noted that a 100% density means a fully connected network).

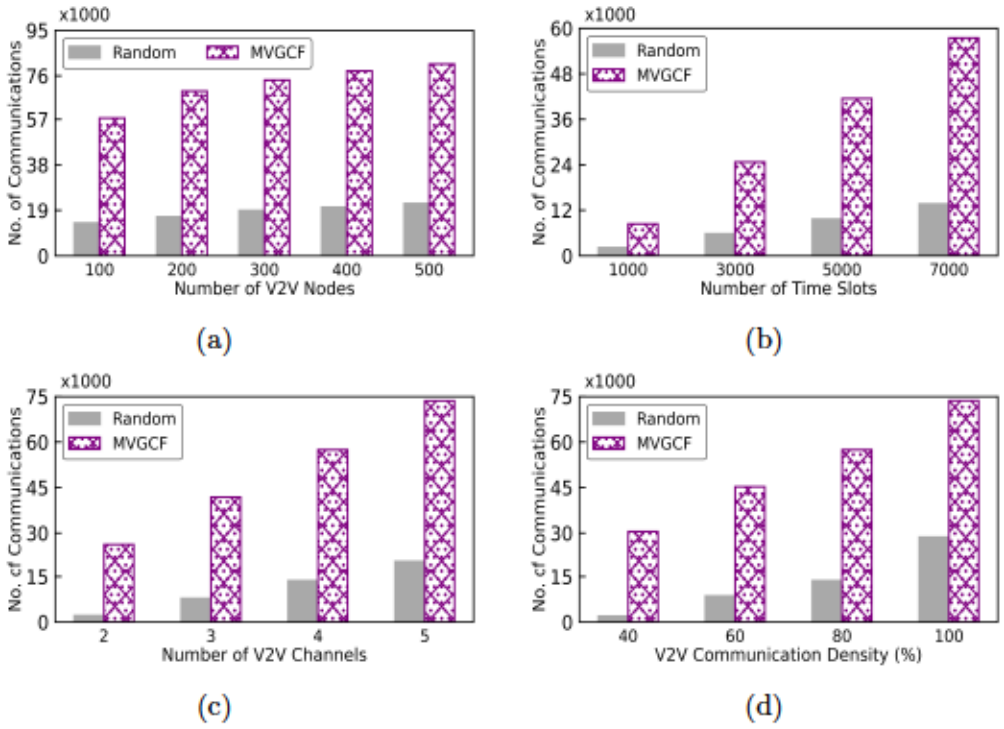


Figure 3.13: Total number of communications as a comparison metric for different methods (Random, and MVGCF) by varying (a) number of V2V nodes, (b) number of time slots, (c) number of V2V channels, and (d) data communication range.

Fig. 3.13(a) and Fig. 3.14(a) respectively show the total number of communications and fairness by varying the number of nodes between 100 and 500. By setting the number of groups to be 20 % of the total number of nodes, the figures show that the communications gradually become harder to achieve for a larger number of nodes due to failure in maintaining fairness as the routing paths become increasingly long. Nevertheless, the MVGCF method still outperforms the Random method by a significant margin, with 57,428 successful communications compared to 13,619 for 100 nodes, and an even greater disparity of 80,816 to 21,816, respectively. When it comes to fairness, the MVGCF method boasts an

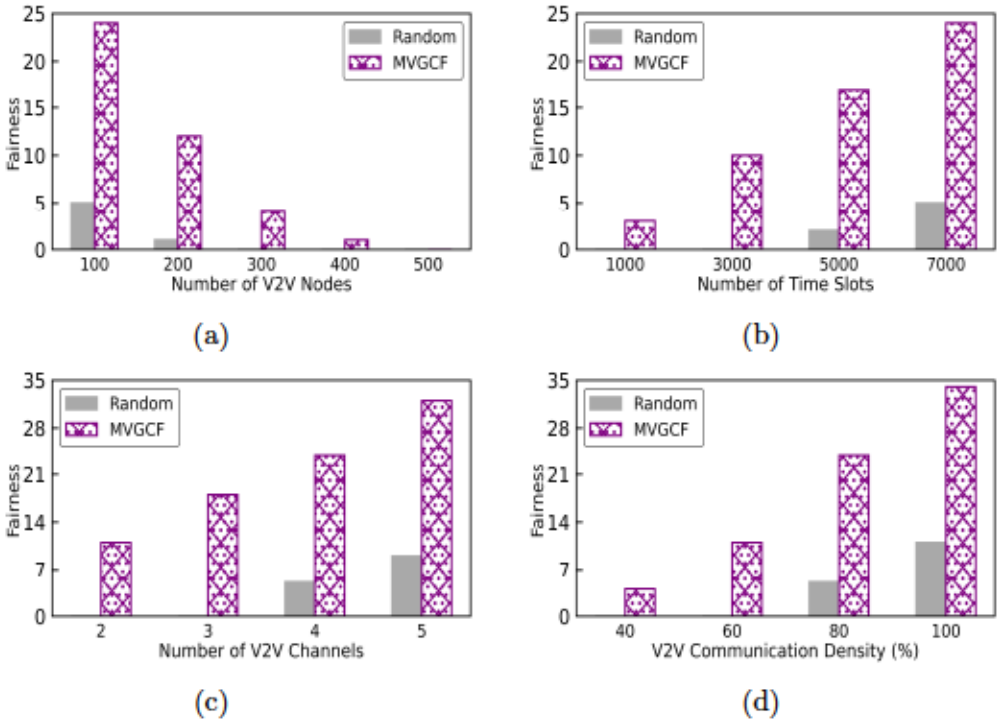


Figure 3.14: Results of fairness (total number of successful communications rounds for all pairs) for all methods by varying (a) number of V2V nodes, (b) number of time slots, (c) number of V2V channels, and (d) data communication range.

impressive F score of 24 for 100 nodes, which, unfortunately, drops to 0 as the number of nodes reaches 500. The Random method, on the other hand, starts with a F score of 5 for 100 nodes, but then fails to exhibit any fairness (i.e., $F = 0$) when the number of nodes increases to 500.

Fig. 3.13(b) and Fig. 3.14(b) respectively show the total number of communications and fairness by varying the number of time slots between 1000 and 7000. As plotted in the figures, the total number of communications and fairness of MVGCF and Random methods increases as expected with the number of time slots due to having more RBs to allocate transmissions. The MVGCF method outperforms the Random method with 3x (three times) more successful communications when the total number of time slots is 1000, and this performance gap gradually increases as the number of time slots increases. The performance gap reaches 4x (four times) when the number of time slots reaches 7000.

Fig. 3.13(c) and Fig. 3.14(c) respectively show the total number of communications and fairness by varying the number of channels between 2 and 5. As depicted in the figures, the total number of communications and fairness of the MVCCF and Random methods increases with the increment of channels. For instance, the total number of successful communications (respectively fairness) is 26,203 (respectively $F = 11$) when there are only

two channels, and it is 73,582 (respectively $F = 32$) when the number of channels is 5. The reason for this increase is that there are more resource blocks to allocate transmissions. However, there is a slight improvement in the number of successful communications and fairness for the Random method.

The figures presented in the study, namely Fig. 3.13(d) and Fig. 3.14(d), shed light on the performance of both methods by varying the density of network connections. As depicted in the figures, the total number of communications and fairness of the MVCCF and Random methods increase while dealing with denser networks. The reason is that there are more links to be allocated into resource blocks. While the Random method starts with a F score of 0, it eventually reaches a maximum fairness score of $F = 11$ when the network density is 100 %. However, despite this accomplishment, the Random method pales in comparison to the MVGCF method in terms of the number of communications.

The MVGCF method always outperforms the random method for large networks because it prioritizes fairness while maximizing the number of V2V communications.

3.7 Summary

This chapter aims at maximizing the total number of V2V communications while maintaining fairness for groups of vehicles, where in each group, vehicles are interested in communicating with each other. The complexity of the problem lies in the fact that multiple factors need to be addressed, including finding a multi-hop routing path for each source-destination V2V communication pair, transmission power control, link scheduling, and taking into account resource allocation under the Half-Duplex and SINR constraints. We mathematically formulated the problem and implemented an optimal solution using the mixed integer linear programming (MILP). After proving the NP-hardness of the problem and owing to its complexity, we proposed a scalable method named Maximizing V2V Group Communications and Fairness (MVGCF) to get approximate solutions for large networks. The main concept behind the MVGCF method is to maintain a priority queue, which is used to achieve fairness by prioritizing the V2V communication pairs that have a fewer number of successful communications. To address the objective, it is modeled as a Markov Decision Process (MDP), and two RL algorithms, namely Qlearning and DDQN methods are introduced. However, for faster learning and better performance, two heuristic-based RL methods, namely MVGCF-Qlearning and MVGCF-DDQN are proposed. Through numerical results, the heuristic-based RL methods demonstrated significant improvements compared to the conventional RL methods on small, medium, and large instances. In addition, the MVGCF-DDQN method outperformed other methods in terms of both the total number of V2V communications and fairness. By contributing innovative solutions for maximizing V2V communications while maintaining fairness in AV-assisted vehicular networks, this

chapter contributes to the advancement of intelligent transportation systems and smart cities.

Chapter 4

Optimizing Information Freshness for Autonomous Vehicular Communication

Autonomous Vehicles (AVs) are expected to play a crucial role in intelligent transportation systems, especially in future smart cities. AV-assisted vehicular networking research has primarily focused on throughput and latency as performance metrics to support their operations. However, these conventional metrics do not adequately capture the time-sensitiveness of data streams and the freshness of information, which is critical for services such as autonomous driving and accident prevention. Hence, this chapter addresses the problem of minimizing the age of information (AoI) of all data streams in AV-assisted vehicular networks. We also consider a scenario where sensors like LiDARs and cameras on vehicles generate time-sensitive data streams, which are utilized to collect and process this data while maintaining a minimum AoI. Our objective is to minimize the total or average AoI of all data streams for autonomous vehicles over a specified time frame. We first mathematically formulate the problem as a mixed integer linear programming (MILP) to obtain the optimal solutions. However, due to its complexity, we propose a scalable heuristic method named the Online Age of Information Minimization Method (OAMM) to solve the problem for large networks. To incorporate the dynamics of the environment, we model the problem as a Markov decision process (MDP) and solve it using one of the reinforcement learning (RL) algorithms called Qlearning. Furthermore, to enhance the learning behavior of the RL agent and improve overall performance, we introduce a hybrid approach named OAMM-Qlearning, combining both the heuristic-based and Qlearning methods. Our numerical results demonstrate the effectiveness of the hybrid approach in efficiently minimizing the expected weighted total or average AoI compared to a Random method, the OAMM method, and the conventional RL method over small and large networks.

4.1 Introduction

In the rapidly evolving landscape of wireless networks, the advent of autonomous vehicles (AVs) is expected to revolutionize the way we interact with transportation systems, especially in the context of future smart cities. As AV-assisted vehicular networking becomes increasingly prevalent, addressing various applications' diverse quality of service (QoS) requirements becomes a critical challenge. Traditionally, performance metrics such as wireless communication latency, throughput, and service reliability have been used to evaluate system efficiency and support AV operations. However, for real-time applications that heavily rely on the timeliness and freshness of information, conventional metrics may fall short of accurately capturing the effectiveness of data delivery. To bridge this gap, researchers have introduced the concept of the Age of Information (AoI) or status age, which quantifies the time elapsed since the most recent status update. AoI provides a novel performance metric to assess the freshness of collected information, offering valuable insights for time-sensitive applications [21–23].

Ensuring the freshness of information is vital for effectively functioning Intelligent Transportation System (ITS) applications, including autonomous intersection management, traffic control, and autonomous driving. These applications primarily rely on real-time information such as drivers' behavior and emergency braking, which is generated by numerous LiDAR sensors installed in intelligent and internet-connected vehicles. Consequently, collecting timely and fresh information is crucial for enhancing driving assistance and ensuring safety [70–74]. The information can be collected through WiFi technology, which can subsequently be processed and analyzed at edge servers to derive meaningful insights.

In this chapter, each node possesses distinct data streams that need to be broadcasted. However, due to restrictions imposed by transmission power and fading within the communication range, not all nodes can establish a direct link with the source node. Therefore, for others, we may need to re-broadcast the data stream over multiple nodes (vehicles). We consider a Time Division Multiple Access (TDMA) medium access scheme, dividing the time frame into equal-length time slots. The duration of the time frame is intentionally set to be very small, ensuring that the vehicles' positions, determined by their maximum speed, do not significantly change during the data broadcasting process, even in the case of multi-hop transmissions. Therefore, we can allocate a new time frame to account for future changes in vehicle positions. We assume that each packet belonging to a data stream can be accommodated and transmitted within a single time slot. It is important to note that due to the half-duplex nature of the system, a node can either transmit or receive a data packet at a given time. However, simultaneous transmissions can occur within the network if the signal-to-interference plus noise ratio (SINR) at the receivers surpasses a predefined threshold.

The primary objective of this chapter is to minimize the total or average AoI of all data streams for autonomous vehicles over the entire time frame while considering the limitations imposed by the half-duplex constraint, transmission range, and SINR thresholds. All the nodes (vehicles and road infrastructure) can participate in relaying or rebroadcasting data streams so that farther nodes can receive them. Here, each node may broadcast data packets for different data streams, and therefore a scheduling scheme is required to allocate time slots for data broadcasting such that the total/average AoI of all data streams at all nodes is minimized for the entire time frame. This combinatorial problem involves deciding which packets of data streams to broadcast over transmission links, scheduling links on time slots, ensuring packet order transmission on multi-hop paths, and allowing simultaneous transmissions under the SINR constraint. To the best of our knowledge, no such combinatorial problem has been tackled and solved before. However, we have modeled the problem mathematically and solved it using the optimization model and the OAMM method. In this chapter, to consider the dynamic nature of the environment more precisely, we model the problem as Markov Decision Process (MDP), and solve it using two reinforcement learning (RL) algorithms, namely Qlearning and Double Deep Q-Networks (DDQN). Furthermore, to improve the performance of both methods, we merge our heuristic algorithm OAMM with them and propose two hybrid heuristic-based RL methods, namely OAMM-Qlearning and OAMM-DDQN.

The system model and problem description are presented in Section 4.2, while the mathematical formulation is given in Section 4.3. The Online Age of Information Minimization Method (OAMM) is detailed in Section 4.4, and the hybrid RL method, OAMM-Qlearning, is explained in Section 4.5. Section 4.6 presents the performance evaluation, and Section 4.7 summarizes the chapter with key findings and suggests potential avenues for future research.

4.2 System Model and Problem Description

4.2.1 System Model

We consider a road structure consisting of several IoT devices like traffic lights, cameras, LiDAR, radars, sensors, and a set of autonomous driving vehicles, where all these nodes can collect data from their surroundings and share it with all other nodes in a vicinity V. This vicinity V can be from a few hundred meters to one or two kilometers depending on the importance of data for autonomous driving. We consider the system over multiple time frames. Each frame is segmentized into equal time slots, $t = 1, 2, ..., T$; the total number of time slots in a frame is T. The size of the time frame T is considered very small such that the position of vehicles, based on their maximum speed, will not significantly change to affect the broadcasting of data even with multi-hop transmissions. Therefore, we can

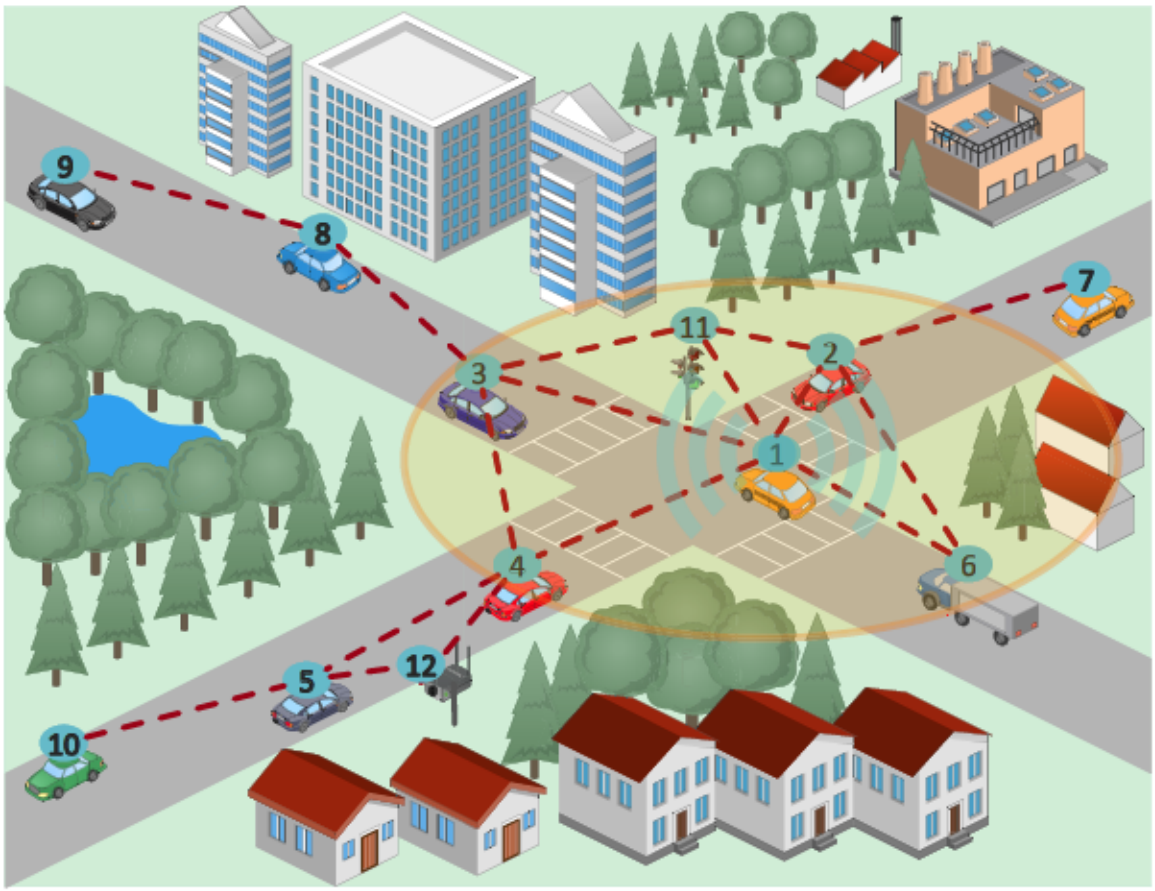


Figure 4.1: Illustration of the system model; the wireless communications between vehicles as well as road IoT devices such as traffic lights and cameras are shown by dotted lines, and the communication range is shown by a circle around a node.

consider a new time frame for the future vehicle position change. The vehicles' speed is considered to follow a truncated Gaussian distribution ranging from ν_{min} to ν_{max} [60], and vehicles travel at random speed [30,61]. Also, the vehicles' arrival into the road segment is considered to follow a Poisson distribution with density ρ Vehicle/Km [68]. We consider our system at each time frame, as shown in Fig. 4.1, as a graph $\mathbb{G} = (N, E)$, where N is a set of nodes in the road segment and E is a set of edges (links) connecting any two nodes residing within each other's communication radius. For simplicity, the power transmission is assumed to be fixed equal to P. Hence, the graph $\mathbb{G}$ is constructed in advance at the beginning of each time frame.

We assume each node at random generates and broadcasts data stream to help vehicles in the vicinity V to better decide on their autonomous driving. Since not all the nodes in this vicinity may have a direct link with the source node, the data stream may be re-broadcasted over multiple hops. Hence, each node might have to broadcast data that belongs to different streams. We consider a Time Division Multiple Access (TDMA) medium access where time

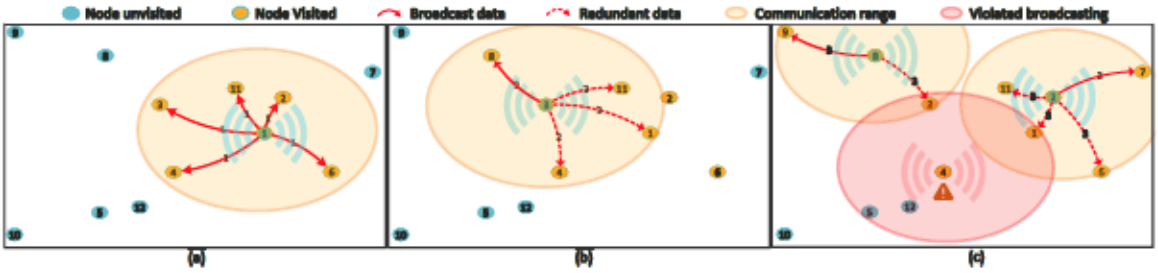


Figure 4.2: Illustration of data scheduling for node 1 to broadcast its packet to all nodes in the network: a) Node 1 is scheduled first to broadcast its data to all nodes in its communication range. b) Node 3 has been scheduled next to broadcast the data packet of node 1. c) Nodes 2 and 8 are scheduled to broadcast simultaneously since their communication ranges do not collide with each other. Note that node 4 can not be scheduled with either node 2 or 8 since its communication range collides with them.

is divided into slots of equal length as explained earlier. We assume that any packet of a data stream can be fit and transmitted over a one-time slot. It is noted that due to the half-duplex mechanism a node can only transmit or receive one data packet at a time. However, we might have simultaneous transmissions in the network if the signal-to-interference plus noise ratio (SINR) at receivers is above a certain threshold β. Let e_{ij} be the Euclidean distance between two nodes i and j, and α be the path loss exponent. Then, the SINR under the physical interference model [31,63] in the presence of concurrent transmissions is obtained as follows:

$$SINR_{(i,j)} = \frac{e_{ij}^{-\alpha} P}{\eta + \sum_{\forall (h,k) \in E: h \neq i} e_{hj}^{-\alpha} P} \geq \beta \qquad \forall (i,j) \in E \tag{4.1}$$

where η is the background noise. For simplicity, we assume that a node has a communication range L so that it can communicate with other nodes within this range successfully if its communication range does not collide with communication ranges of other simultaneous transmissions (see Fig. 4.2(c), node 4 cannot broadcast simultaneously with either node 2 or node 8, whereas, both nodes 2 and 8 can simultaneously transmit since their communication areas do not collide with each other). To simplify the broadcasting scheduling and satisfy the SINR for simultaneous transmissions, we calculate here in offline mode the safety margin δ; where a node cannot transmit if its interference on an active third-party receiver may disturb the SINR in the receiver to a level below the threshold β.

4.2.1.1 Determining the set of silent nodes for a given transmitter that guarantees that all receives successfully receive packets

If L and P are respectively the communication range and transmission power for our system, the set of nodes N_j that have to be silent for any transmitter j can be determined as follows.

Given the half-duplex constraint of the environment under study, we can calculate the interference upper bound by considering half of the nodes being transmitters and, as a consequence, causing interference on third-party receivers. With this assumption, we identify the node i most susceptible to interference within j's communication range L by calculating the cumulative interference of all third-party transmitters in the network. It is worth noting that this set of third-party transmitters is outside the j's and i's communication range due to the half-duplex constraint.

The upper bound for the cumulative interference on node i for the transmitter j is:

$$I_{ij} = \sum_{k \notin (N_j \cup N_i)} e_{ik}^{-\alpha} P w_k \tag{4.2}$$

where N_j and N_i are the set of nodes within the communication range L for the nodes i and j, $w_k \in [0, 1]$ is an indicator function that assumes that the nodes outside of the set N_j and N_i are sorted according to their distances to i and that the nearest node to i is a transmitter (to achieve the interference upper-bound) followed by an alternating sequence transmitters with $w_k = 1$ and not transmitters with $w_k = 0$.

With the upper bound interference estimate, we can calculate $SINR_{ij}$ and compare it to the threshold β.

$$SINR_{ij} = \frac{e_{ij}^{-\alpha} P}{I_{ij} + \eta} \tag{4.3}$$

if $SINR_{ij} \geq \beta$ then the transmission safety margin $\delta = 0$. If $SINR_{ij} < \beta$, then $\delta_{ij} = e_{ih} - L$, where h is the transmitter node that is nearest to i and h is inserted into N_j so that it is not considered a transmitter in a new iteration of interference and SINR estimations.

The transmission safety margin δ_{ij} is then iteratively estimated by the execution of equations 4.2 and 4.3. Any other transmitter node in the network can use this safety margin because the i is the receiver most susceptible to interference, and j is the least favorable transmitter for i by being the farthest possible transmitter. It is worth noting that at the end of this process, N_j will contain the broadcast set for j (i.e. the set of nodes that receive a packet if j is actively transmitting) and a set of nodes that must be silent for i to successfully receive a transmission from j.

Similarly, for each node h in the network, we calculate N_h whenever the graph G is created, before the beginning of a time frame. The sets N can then be used to constrain the number of active transmitters in the same manner as the half duplex constraint while guaranteeing that the SINR constraint is satisfied for all receivers, i.e., nodes in N_h with distance from h smaller than L would receive a packet if transmitted by h, nodes in N_h with distances greater than L would be marked as silent if h is an active receiver.

4.2.2 AoI Definition in Vehicular Networks

The concept of AoI describes the freshness of information from the perspective of the receivers, and it is defined as the time difference since the most recently delivered message was generated. To track the AoI, we define $A^t_{n,d}$ as the AoI of data stream d at receiver node n at time t. It is to be noted that data stream d refers to the node that generates this stream. A data packet for stream d is generated with probability $\lambda_d \in (0, 1], \forall d$. If we let $G^t_d \in \{0, 1\}$ be the indicator of generating new data for stream d at time slot t, then the probability of $G^t_d = 1$ follows Bernoulli distribution with probability λ_d. When a node n generates a new packet for data stream d, the AoI of that data stream and the node is set to zero (i.e $A^t_{n,d} = 0$), since the data stream d is generated and intended for node n (i.e., Node n is the generator and receiver of data stream d). Otherwise, as one time slot passes by, either the data packet is waiting in a queue of a node or in transmission from node to node through multi-hop, the AoI increases by one, until the data is delivered to the destination. There is a possibility that while a data packet is being delivered to a destination node, a new data packet is generated for the same stream. Hence, it might be more efficient to drop the old data which has not been delivered yet and schedule the newly generated data to be transmitted and delivered to the destination so as to reduce the average AoI of the network.

4.2.3 Problem Definition

We are interested in minimizing the total or average AoI of all data streams that are intended for autonomous vehicles in the network within the road segment for the entire time frame. All the nodes (vehicles and road infrastructure) can participate in relaying or rebroadcasting data streams so that farther nodes can receive them. Here, each node may broadcast data packets for different data streams, and therefore a scheduling scheme is required to allocate time slots for data broadcasting such that the total/average AoI of all data streams at all nodes is minimized for the entire time frame.

Problem Definition (minimizing AoI for all data streams): *Given a graph $\mathbb{G}$ of N nodes connected through E edges with neighbors within the communication range L, the problem of minimizing AoI for all data streams is to schedule time slots for broadcasting data streams generated by all nodes in the network within the read segment to be delivered to all autonomous vehicles within the vicinity V around each generated node in the current time frame T (where the frame is partitioned into multiple equal time slots) such that the total/average AoI of all data streams at all nodes is minimized.*

Here, each node generates a data stream to broadcast to all nodes in the vicinity V. We illustrate the data broadcasting for one node in Fig. 4.2 on a sample network shown in

Table 4.1: Notations Used in problem formulation

Parameters		
N		Set of nodes.
E		Set of edges (links).
D		Total number of data streams.
T		Time frame (total number of time slots).
B		Large constant larger than any AoI in the system.
G_d^t		Indicates that when $G_d^t = 1$ a new data for stream d is generated at time t, and $G_d^t = 0$ otherwise.
Variables		
$A_{n,d}^t$	≥ 0	AoI of data stream d at receiver node n at time t.
$X_{ij,d}^t$	$\in \{0,1\}$	Indicates whether link (i,j) for data stream d is scheduled at time t or not.
$P_{ij,d}^t$	≥ 0	Equal to $A_{i,d}^t$, if $X_{ij,d}^t = 1$ & $A_{i,d}^{t-1} < A_{j,d}^{t-1}$; and zero otherwise.
$H_{ij,d}^t$	$\in \{0,1\}$	Equal to one, if $X_{ij,d}^t = 1$ & $A_{i,d}^{t-1} < A_{j,d}^{t-1}$; and zero otherwise.
$Q_{j,d}^t$	≥ 0	Equal to $A_{j,d}^t$, if $\sum_{i:<i,j>} H_{ij,d}^t = 0$; and zero otherwise.
$R_{ij,d}^t$	$\in \{0,1\}$	Equal to one, if $A_{i,d}^{t-1} < A_{j,d}^{t-1}$; and zero otherwise.

Fig. 4.1; namely, we illustrate the scheduling of the data stream of node 1 and highlight the importance of nodes that should participate in relaying or rebroadcasting data so that fewer number of time slots is used and hence the faster the data is delivered to all nodes. Note that the same procedure is done concurrently for other data streams in the network. Here we show the data broadcasting for only one node. It is to be noted that the scheduling problem will be more complex when we consider multiple data streams instead of one. In the first time slot, illustrated in Fig. 4.2(a), node 1 broadcasts its data packet. In the second time slot, any node that has received the data packet may relay or rebroadcast it, however, a node must be chosen that will result in delivering data to more new nodes (see Fig. 4.2(b)) and will increase the chance for simultaneous transmissions in the future time slot scheduling (see Fig. 4.2(c)). Also, there is a possibility that while a data packet that belongs to the stream is on its way to a destination, a new data packet for the same data stream is generated. Hence, it might be more efficient to drop the old data packet and deliver the new one. Thus, the scheduler should optimize 1) nodes that should participate in broadcasting, 2) when nodes should transmit/broadcast, and 3) which data packets to be broadcasted so that the AoI of the network is minimized.

4.3 Problem Formulation

In this section, we mathematically formulate the problem as a mixed integer linear program (MILP). The used notations are listed in Table 4.1.

Let $A^t_{n,d} \geq 0$ be the AoI of data stream d in node n at time t. The objective of the optimization model is to minimize the AoI of all data streams $d = 1, 2, ..., D$ on all nodes $n = 1, 2, ..., N$ at all time slots $t = 1, 2, ..., T$. It can be mathematically written as follows:

$$Minimize \qquad \sum_{t=1}^{T} \sum_{d=1}^{D} \sum_{n=1}^{N} A^t_{n,d} \tag{4.4}$$

subject to: (4.5) - (4.9), (4.13), (4.17), (4.22), (4.24) - (4.28), where these constraints are derived in detail in Sections 4.3.1 to 4.3.4.

4.3.1 Simultaneous transmissions

To reduce interference and increase the successful transmission rate, we avoid a node receiving multiple data packets from different transmitters or transmitting multiple data packets at the same time. Similarly, we avoid a node transmitting and receiving simultaneously. Let $X^t_{ij,n} \in \{0, 1\}$ indicates whether link (i, j) for data stream d is scheduled at time t or not.

4.3.1.1 Simultaneous Receiving

This constraint ensures that a receiver does not receive data packets from multiple transmitters simultaneously:

$$\sum_{d=1}^{D} \sum_{i:(i,j)\in E} X^t_{ij,d} \leq 1 \qquad \forall j \in N, t = 1...T. \tag{4.5}$$

4.3.1.2 Simultaneous Transmitting

The following constraint prevents a transmitter from transmitting different data packets to multiple receivers at the same time. In other words, a node cannot transmit more than one data packet at the same time:

$$\sum_{d=1}^{D} X^t_{ij,d} \leq 1 \qquad \forall (i, j) \in E, t = 1...T. \tag{4.6}$$

4.3.1.3 Simultaneous Receive and Transmit

Using the following constraint, we ensure that a node does not transmit and receive at the same time:

$$\sum_{d=1}^{D} X_{ik,d}^{t} + \sum_{d=1}^{D} X_{kj,d}^{t} \leq 1 \qquad \forall (i,k)\&(k,j) \in E, t = 1..T. \tag{4.7}$$

4.3.2 Data Broadcasting

If a node is scheduled to transmit a data packet, based on the nature of the data broadcasting, all the outgoing links from the node should be scheduled at the same time, and vice versa, if a node is not scheduled to transmit, none of its outgoing links should be scheduled.

$$X_{ij,d}^{t} = X_{ik,d}^{t} \qquad \forall (i,j)\&(i,k) \in E, d = 1..D, t = 1..T. \tag{4.8}$$

4.3.3 Initial AoI

The initial AoI for all data streams and nodes is set to an initial value at time $t = 0$ and updated at time $t = 1, 2, ..., T$. Hence, the AoI for all data streams and nodes at time $t = 0$ (i.e., $A_{n,d}^{0}$) is set to $Initial$ except for nodes $n = d$ that might generate a new packet at time $t = 0$ as follows:

$$A_{n,d}^{0} = Initial \qquad \forall n = 1..N, d = 1..D : d \neq n. \tag{4.9}$$

When new data that belongs to the same node is generated, the AoI of that node is set to zero (i.e., $A_{n,d}^{t} = 0$, here node $n = d$ since the same node generates a new data packet, thus we may write it as $A_{d,d}^{t} = 0$). Let $G_{d}^{t} \in \{0, 1\}$ be an indicator that new data for stream d is generated at time t. Hence, when $G_{d}^{t} = 1$ and $n = d$, the AoI of node n is set to zero (i.e., $A_{n,d}^{t} = 0$). At the initial time $t = 0$, the AoI is given as follows:

$$A_{d,d}^{0} = (1 - G_{d}^{0})Initial \qquad \forall d = 1..D. \tag{4.10}$$

When $G_{d}^{0} = 0$, the AoI $A_{d,d}^{0}$ for node and data stream d is set to the initial value. However, when $G_{d}^{0} = 1$, the AoI $A_{d,d}^{0} = 0$. Similarly, at other time slots, when $t \neq 0$, the AoI is set to zero (i.e., $A_{d,d}^{t} = 0$) when a new packet is generated at time t (i.e., $G_{d}^{t} = 0$). The constraint is given in the next subsection.

4.3.4 AoI Updates

As time goes by, the AoI is incremented by one unit as one-time slot passes by (i.e., $A^t_{j,d} = A^{t-1}_{j,d}+1$). When a node receives a data packet for data stream d, if the AoI of the transmitter is smaller than the receiver, then it updates its AoI to be equal to the AoI of the transmitter plus one. In other words, if node j receives a new data packet of stream d from node i at time t (i.e., $X^t_{ij,d} = 1$), and $A^{t-1}_{i,d} < A^{t-1}_{j,d}$, then $A^t_{j,d} = A^{t-1}_{i,d} + 1$; this one AoI unit is added because the data transmission from the transmitter to the receiver takes one-time unit which should be added to the AoI of the receiver. The following equations show how the AoI of a node is updated:

$$A^t_{j,d} = \begin{cases} 0, & \text{if } G^t_d = 1 \quad \& \quad d = j; \\ A^{t-1}_{i,d} + 1, & \text{if } X^t_{ij,d} = 1 \quad \& \quad A^{t-1}_{i,d} \leq A^{t-1}_{j,d}; \\ A^{t-1}_{j,d} + 1, & \text{otherwise.} \end{cases} \tag{4.11}$$
$$\forall (i,j) \in E, d = 1..D, t = 1..T.$$

As explained earlier, when a new packet is generated for data stream d at time t, the AoI of the node that has generated the new packet is set to zero as shown below:

$$A^t_{d,d} = \begin{cases} 0, & \text{if } G^t_d = 1 \\ A^{t-1}_{d,d} + 1, & \text{otherwise.} \end{cases} \tag{4.12}$$
$$\forall d = 1..D, t = 1..T.$$

which can be linearized as follows:

$$A^t_{d,d} = (1 - G^t_d)(A^{t-1}_{d,d} + 1) \qquad \forall d = 1..D, t = 1..T. \tag{4.13}$$

However, when a node with a smaller AoI transmits to a larger AoI node as explained above, the updated AoI of the receiver will be equal to the AoI of the receiver at the time of transmission plus one. Otherwise, the AoI of a node increments by one as shown here:

$$A^t_{j,d} = \begin{cases} \sum_{i:<i,j>} P^t_{ij,d} + 1, & \text{if } \sum_{i:<i,j>} P^t_{ij,d} \neq 0, \\ A^{t-1}_{j,d} + 1, & \text{otherwise.} \end{cases} \tag{4.14}$$
$$\forall j = 1..N, d = 1..D : d \neq j, t = 1..T.$$

where $P^t_{ij,d}$ is equal to the AoI of the transmitter (i) on link $< i, j >$ (i.e., $A^t_{i,d}$), if $X^t_{ij,d} = 1$ and $A^{t-1}_{i,d} < A^{t-1}_{j,d}$, and zero otherwise. So, if node j receives a data packet from a node with smaller AoI, then the AoI of node j will be $\sum_{i:<i,j>} P^t_{ij,d}+1$, which is equal to $A^{t-1}_{i,d}+1$, since at time t, node j can receive a packet from only one transmitter based on constraint 4.5).

Equation (4.14) can be written in a linearized form as follows:

$$A^t_{j,d} = \sum_{i:<i,j>} P^t_{ij,d} + A^{t-1}_{j,d}(1 - \sum_{i:<i,j>} H^t_{ij,d}) + 1 \qquad \forall j = 1..N, d = 1..D : d \neq j, t = 1..T. \tag{4.15}$$

or,

$$A^t_{j,d} = \sum_{i:<i,j>} P^t_{ij,d} + A^{t-1}_{j,d} - A^{t-1}_{j,d} \sum_{i:<i,j>} H^t_{ij,d}) + 1 \qquad \forall j = 1..N, d = 1..D : d \neq j, t = 1..T. \tag{4.16}$$

or,

$$A^t_{j,d} = \sum_{i:<i,j>} P^t_{ij,d} + A^{t-1}_{j,d} - Q^t_{j,d} + 1 \qquad \forall j = 1..N, d = 1..D : d \neq j, t = 1..T. \tag{4.17}$$

where, $H^t_{ij,d}$, $P^t_{ij,d}$, and $Q^t_{j,d}$, are given bellow:

$$H^t_{ij,d} = \begin{cases} 1, & \text{if } X^t_{ij,d} = 1 \quad \& \quad A^{t-1}_{i,d} < A^{t-1}_{j,d}; \\ 0, & \text{otherwise.} \end{cases} \qquad \forall (i,j) \in E, d = 1..D, t = 1..T. \tag{4.18}$$

$$P^t_{ij,d} = \begin{cases} A^{t-1}_{i,d}, & \text{if } H^t_{ij,d} = 1; \\ 0, & \text{otherwise.} \end{cases} \qquad \forall (i,j) \in E, d = 1..D, t = 1..T. \tag{4.19}$$

$$Q^t_{j,d} = \begin{cases} A^{t-1}_{j,d}, & \text{if } \sum_{i:<i,j>} H^t_{ij,d} = 1 \\ 0, & \text{otherwise.} \end{cases} \qquad \forall j = 1..N, d = 1..D, t = 1..T. \tag{4.20}$$

Equation (4.18) can be written as follows:

$$H^t_{ij,d} = X^t_{ij,d} R^t_{ij,d} \qquad \forall (i,j) \in E, d = 1..D, t = 1..T. \tag{4.21}$$

and linearized by the following constraints:

$$\begin{cases} H^t_{ij,d} \leq X^t_{ij,d} \\ H^t_{ij,d} \leq R^t_{ij,d} \\ H^t_{ij,d} \geq X^t_{ij,d} + R^t_{ij,d} - 1 \end{cases} \quad \forall (i,j) \in E, d = 1..D, t = 1..T. \tag{4.22}$$

where,

$$R^t_{ij,d} = \begin{cases} 1, & \text{if } A^{t-1}_{i,d} < A^{t-1}_{j,d}; \\ 0, & \text{otherwise.} \end{cases} \quad \forall (i,j) \in E, d = 1..D, t = 1..T. \tag{4.23}$$

The above equation can be linearized using the following constraints:

$$R^t_{ij,d} \leq \frac{A^{t-1}_{j,d} - A^{t-1}_{i,d}}{B} + 1 \quad \forall (i,j) \in E, d = 1..D, t = 1..T. \tag{4.24}$$

$$R^t_{ij,d} \geq \frac{A^{t-1}_{j,d} - A^{t-1}_{i,d}}{B} \quad \forall (i,j) \in E, d = 1..D, t = 1..T. \tag{4.25}$$

$$R^t_{ij,d} \leq \mid A^{t-1}_{j,d} - A^{t-1}_{i,d} t \mid \quad \forall (i,j) \in E, d = 1..D, t = 1..T. \tag{4.26}$$

where B is a big value constant larger than any AoI in the system. Consequently, equation (4.19) can be linearized using the following constraints:

$$\begin{cases} P^t_{ij,d} \leq A^{t-1}_{i,d} \\ P^t_{ij,d} \leq BH^t_{ij,d} \\ P^t_{ij,d} \geq B(H^t_{ij,d} - 1) + A^{t-1}_{i,d} \end{cases} \quad \forall < i,j > \in E, d = 1..D, t = 1..T. \tag{4.27}$$

and equation (4.20) can be linearized using the following constraints:

$$\begin{cases} Q^t_{j,d} \leq A^{t-1}_{j,d} \\ Q^t_{j,d} \leq B \sum_{i:<i,j>} H^t_{ij,d} \\ Q^t_{j,d} \geq B(\sum_{i:<i,j>} H^t_{ij,d} - 1) + A^{t-1}_{j,d} \end{cases} \quad \forall j = 1..N, d = 1..D, t = 1..T. \tag{4.28}$$

4.4 The Online Age of Information Minimization Method (OAMM)

From the fact that the problem of minimizing the AoI of data streams is NP-hard, and obtaining the optimal solutions using the optimization model given above is very complex and non-scalable, in this section, we propose a heuristic method to solve the problem online for consequent time frames with any vehicle location changes. The details of the heuristic method are given in Algorithm 6.

The inputs to the algorithm are comprised of a set of nodes (N), a set of edges or links (E), the total number of data streams (D), the total number of time slots in a given time frame (T), and packet generation probability with Bernoulli distribution (λ_d). The outputs of the algorithm are the total sum of AoI ($TotalAoI$), and the resource block allocation RB. At initialization, the AoI for all data streams and nodes is set to an initial value at time slot $t = 0$ (i.e., $A^0_{(n,d)} = 0$), except for nodes (n) that might generate new packets for their data streams (d), that is $n = d$.

In line 2, the algorithm calls function $SampleG$, detailed in Algorithm 7, to sample G (probability distribution of new packet generation) which follows Bernoulli distribution with probability λ_d. This function takes as inputs AoI A, an indicator of generating new data for all data streams G, and time slots t. Eventually, the function creates a new packet (p) when $G^t_d = 1$, where packet p is comprised of a packet id, a set of transmitted nodes, a set of received nodes, and a counter to track the number of nodes that have not received the packet. Then an ID for the packet p is generated by subtracting the current time t from the frame size T (i.e., $p.id = T - t$, line 6 of Algorithm 7) to track the packet number for each data stream. Consequently, the transmitted set of packet p is set to null because this packet has just been generated without being transmitted yet (i.e., $p.transmitted = \varnothing$, line 7), and it will be ready to be transmitted for the next time slot. As a new packet, it is inserted into the received set (line 8), and the number of non-visited nodes for this packet p is decreased by one (line 9). Eventually, packet p is inserted into the packet list ($d.packlist$) of data stream d which has generated this new packet (line 10), and the AoI of the stream d is set to zero (i.e., $A^0_{d,d} = 0$, line 11).

In line 3 of Algorithm 6, the time slot t iterates over the time frame T. For each time

Algorithm 6: The OAMM method

Data: N, E, D, T, λ_d;
Result: $TotalAoI$, RB;

1 Initialize: $A^0_{(n,d)} = Initial$;
2 SampleG(A,G,0);
3 **for** $t = 1; t <= T; t++$ **do**
4 $T_{ran} = \varnothing$, $R_{ec} = \varnothing$, $\mathcal{PA}_t = \varnothing$;
5 **for** *data stream* d *in* D **do**
6 **for** *packet* p *in* $d.packlist$ **do**
7 **for** *packet* n *in* $p.received$ **do**
8 Insert the tuple (d, n, p) into $\mathcal{PA}_t$;
9 Sort d based on the sum of AoI of each data stream;
10 Sort $\mathcal{PA}_t$ in order based on the following:
11 a) Data stream d;
12 b) Maximum length of the broadcast/transmitter set;
13 c) Newest packet ID;
14 **for** *each possible action* a *in* $\mathcal{PA}_t$ **do**
15 **if** $a.n \notin (T_{ran} \cup R_{ec})$ **then**
16 **if** $a.n.broadcast \cap (T_{ran} \cup R_{ec}) == \varnothing$ **then**
17 Allocate a into RB;
18 Insert $a.n$ into T_{ran};
19 Insert $a.n.broadcast$ into R_{ec};
20 Move $a.n$ from $a.p.received$ into $a.p.transmitted$;
21 $W_{rec} = a.n.broadcast \setminus (a.p.transmitted \cup$ a.p.received);
22 Update $A^t_{w,d}$ $\forall w \in W_{rec}$ according to Eq.4.11;
23 $a.p.notvisited = |N| - |a.p.received| + |a.p.trasmitted|$;
24 **if** $a.p.notvisited == 0$ **then**
25 Remove p from $a.d.packlist$;
26 Drop older packets covered by newer packets;
27 SampleG(A,G,t);
28 $TotalAoI = \sum_{n=1}^{N} \sum_{d=1}^{D} \sum_{t=1}^{T} A^t_{d,n}$;

slot starting with $t = 1$, in line 4, the algorithm first empties the transmitter set (T_{ran}), receiver set (R_{ec}), and the list of possible actions ($\mathcal{PA}_t$). Then, for each data stream d and for each packet p in the data stream list ($d.packlist$) and received list ($p.received$), it inserts the tuple (d, n, p) into the list of possible action $\mathcal{PA}_t$, where p is the packet id in short (referred to $p.id$ in Algorithm 7), and n is one of the nodes that receives packet p and ready to broadcast it. In line 9, the algorithm computes the sum of AoI for each data stream and then sorts them in descending order. The list of possible actions $\mathcal{PA}_t$ is first sorted based

Algorithm 7: Sample G from bernoulli distribution of lambda

```
1  Function SampleG(A, G, t):
2      Sample G^t ~ Bernoulli(λ_d);
3      for data stream d in D do
4          if G_d^t = 1 then
5              Create new packet p;
6              p.id = T − t;
7              p.transmitted = ∅;
8              Insert d into p.received;
9              p.notvisited = |N|−1;
10             Insert p into d.packlist;
11             A_{d,d}^t = 0;
```

on data streams, and then based on the maximum length of the broadcast/transmitter set followed by the newest packet ID in case of tie (lines 10-13).

In line 14, for each action a in $\mathcal{P}A_t$, the algorithm checks the half-duplex constraint, which ensures that a node $a.n$ is not in the sets of transmitters T_{ran} and receivers R_{ec} (line 15). Line 15 is essentially checking whether the intersection between the broadcast set of a transmitter for action a and the union of the sets of transmitters T_{ran} and receivers R_{ec} is an empty set. If the condition is satisfied, it means that there are no overlapping nodes between the broadcast set and the transmitters or receivers. So, it is possible to allocate the action a into the resource block (RB). Once the resource block allocation is done, both transmitter $a.n$ and the set of receivers who received the data packet through the broadcasting (i.e., set of $a.n.broadcast$) are respectively inserted into T_{ran} and R_{ec} (lines 18-19). In addition, the transmitter $a.n$ is removed from the received packet set to the transmitted packet set (line 20). To keep track of the nodes that have not yet received data packets for different data streams, in line 21, a new set, W_{rec}, is constructed by taking the difference between the broadcast set of a transmitter $a.n$ (a.n.broadcast) and the union of two sets, including the set of packets already transmitted $a.p.transmitted$ and the set of packets already received $a.p.received$.

In line 22, an update process is taken into consideration for each element w in the W_{rec} set; the update is performed based on Eq.4.11, which was explained earlier. The set of packets for unvisited nodes is obtained by removing received and transmitted node sets ($|a.p.received|$) and ($|a.p.transmitted|$) from the total number of nodes ($|N|$). If the difference is equal to zero, then it means all nodes in the network are either considered as the receiver or transmitter of the packet. Hence, we remove the packet from that data stream and update the packet list ($a.d.packlist$) (lines 24-25).

The algorithm in line 26 before going to the next time slot drops and replaces the

old packets if newer packets have arrived for the same data streams, and then in line 27 function *SampleG* is called to generate new packets based on the probability distribution G. Finally in line 28, the algorithm sums up the AoI values for each time slot, ranging from t=1 to t=T, resulting in the TotalAoI. This provides us with a comprehensive understanding of the overall freshness of the information gathered over the time period.

4.5 Reinforcement Learning and Hybrid Approaches

Reinforcement learning (RL) is an approach where an agent interacts with an environment to learn a policy that maximizes its long-term rewards. The agent takes actions from the given states in the environment, obtains feedback in the form of rewards, and uses the information to update its policy.

In our method, we formulate our maximization problem as Markov Decision Process (MDP) to allocate resources into RBs within a time frame. An MDP is represented by a tuple (S, A, γ, P, R), where S is a finite set of states, denoted as $s_t \in S$ at time slot t; A is an action space such that if we let $\mathcal{A}$ to be a set of all possible actions in state s_t and a_t is one of the actions at any time slot t, then $a_t \in A$; $\gamma \in [0, 1]$ is the discount factor, which determines the weight of future rewards in the decision-making process; P is a Markovian transition model, denoted as $P(s_{t+1} \| s_t, a_t)$, which represents the probability of transitioning from state s_t to state s_{t+1} when an action a_t is taken; R is the reward distribution, denoted as $P(r_t \| s_t, a_t)$, which gives the immediate reward $r_t \in R$ after an action a_t is taken in a state s_t at time slot t. The state, action, and reward functions under the MDP framework are given as follows:

4.5.0.1 Agent

Roadside Unit (RSU) is considered to be an agent.

4.5.0.2 State

Each state s_t is defined as a tuple of multiple vectors: i) a matrix containing the current AoI at time slot t, aoi_t; ii) a vector containing the current mean of AoI of each data stream at time slot t, $mean_t$; iii) a vector of the current median of AoI of each data stream at time slot t, $median_t$. Thus the system state s at time slot t can be expressed as:

$$s_t = (aoi_t, mean_t, median_t). \tag{4.29}$$

4.5.0.3 Action

Each action a is defined as a tuple composed of i) An identifier used to assign each data stream to transmit, $d \in D$; ii) An identifier of a scheduled transmitter for that data stream, $h \in N$; iii) An identifier of a considered packet for that data stream, p. The agent action a can be expressed as:

$$a = (d, h, p). \tag{4.30}$$

4.5.0.4 Reward

We use a reward function to provide feedback on each action a_t taken in a given state s_t by the RL agent. The agent selects an action a_t from a set of possible actions $\mathcal{A}_t$ at time slot t, where $\mathcal{A}_t$ represents the available resource allocation choices at the timestep t. Let r_t be the immediate reward at each time slot t. The reward function $r_t(s_t, a_t)$ at time slot t can be expressed as follows:

$$r_t(s_t, a_t) = \frac{\sum_{A^t\ 1} - \sum_{A^t}}{max(max(A^{t-1}), max(A^t))} \tag{4.31}$$

If the maximum value of the previous AoI (A^{t-1}) is greater than the maximum value of the current AoI (A^t), we normalize both of them by dividing over the maximum value of the previous AoI. Otherwise, we normalize both by dividing over the maximum value of the current AoI. The reward $r_t(s_t, a_t)$ is then computed by obtaining the difference between the normalized values of the previous AoI and the current AoI.

4.5.1 The Qlearning Approach

For the RL approach, we use the Qlearning network [67] given in Algorithm 8 that utilizes an off-policy method and runs for K episodes to allocate resources into resource blocks within a time frame T. The input to the algorithm is the interface of the environment, and the outputs are the total sum of AoI and scheduled resource blocks RB. Initially, the Q-value for each state s and action a is zero. The algorithm iterates over several episodes (i.e., K), and at each episode k, for each time slot t, while the terminal state is not reached, it initializes the transmitter set T_{ran} and the receiver set R_{ec} to null (line 4). In line 5, the algorithm at the start observes a set of state components, consisting of the tuple $(aoi_t, mean_t, median_t)$, from the environment. Then, in line 6, the algorithm chooses an action a_t from a list of possible actions $\mathcal{A}_t$ in a given state s_t utilizing an ϵ-$Greedy$ policy, and allocates a_t into the resource block RB_t. In line 7, after taking an action, the algorithm retrieves a reward (r_t) from the environment and updates the $Q(s_t, a_t)$ value in the Q-table by choosing the

future action that returns the highest expected value (see line 8), taking into account a learning rate α and discount factor γ. The Q-value table is repeatedly updated at multiple iterations so that it converges and there will be no significant improvements. In the end, the algorithm executes the trained agent in the environment (line 9), and obtains the total sum of AoI $TotalAoI$ (line 10).

Algorithm 8: Proposed Qlearning based Solution

Data: Environment Interface;
Result: $TotalAoI$, RB;
1 Initialize: $Q(s,a) = \varnothing$;
2 **for** $k \leftarrow 1 : K$ **do**
3 **for** $t \leftarrow 1 : T$ *and* $s_t \neq s_T$ **do**
4 $T_{ran} = \varnothing$, $R_{ec} = \varnothing$;
5 Observe s_t $(aoi_t, mean_t, median_t)$ from the environment;
6 Choose a_t (ds_t, tr_t, p_t) from a list of possible actions $\mathcal{A}_t$ using an ϵ-Greedy policy and allocate a_t into RB_t
7 Receive a Reward (r_t) from the environment;
8 Update $Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_t + \gamma \max Q(s_{t+1}, a_{t+1} \in \mathcal{A}_{t+1}) - Q(s_t,a_t)]$;
9 Execute trained agent on environment interface;
10 Calculate $TotalAoI = \sum_{t=1}^{T} A^t$;

4.5.2 OAMM-Qlearning

The conventional Qlearning approach follows an exploration and exploitation approach to gather information about state-action pairs until it converges. The sizes of state and action spaces can be calculated as follows:

$$O(S) = O(A \times mean \times median) \tag{4.32}$$

$$O(\mathcal{A}) = O(D \times N \times T). \tag{4.33}$$

In fact, since the size of the state-action spaces is extremely large which necessitates creating a very large Q-value table, exploring the entire state-action spaces to update the $Q(s_t, a_t)$ value for each state-action becomes challenging and computationally expensive, leading to poor performance and slow convergence. Hence, to overcome this issue, we propose OAMM-Qlearning, a heuristic-based reinforcement learning approach, which is the integration of the heuristic OAMM method and the conventional RL method. By integrating the heuristic OAMM insights into the learning process of the RL approach, the agent accelerates its learning by understanding which actions in specific states yield the best

rewards. This approach delegates the model to enhance the quality of solutions obtained to converge faster toward optimal solutions. In brief, the OAMM-Qlearning method solves the problem using the heuristic OAMM method, generates a sequence of states, actions, and rewards, required for RL implementation, and stores these initial results in a list called $SARlist$. Then it updates the Q-table with the set of states and actions observed from the list $SARlist$. Finally, the RL agent continues training similar to the conventional Qlearning technique given in Algorithm 8. The details of the OAMM-Qlearning method are given in Algorithm 9. The algorithm takes a set of nodes (N), a set of edges or links (E), the total number of data streams (D), the total number of time slots in a given time frame (T), and packet generation probability with Bernoulli distribution (λ_d) as inputs. Initially, the $SARlist$ is empty (line 1). The algorithm executes the OAMM method using the inputs (line 2), obtains a sequence of states (S), actions (A), and rewards (R) (line 3), and stores them in list $SARlist$ (line 4). Then, in line 5, the algorithm iterates over the results stored in the list to update the Q-table. Finally, it trains and executes the RL agent similar to lines 2-10 given in Algorithm 8.

Algorithm 9: OAMM-Qlearning

Data: N, E, D, T, λ_d;
Result: $TotalAoI$, RB;
1 Initialize: $SARlist = \varnothing$;
2 Execute the OAMM method taking N, E, D, T, λ_d as inputs;
3 Observe the OAMM method execution to collect a sequence of states (S), actions (A), and rewards (R);
4 Store the sequences (S, A, R) into the $SARlist$;
5 Iterate over the $SARlist$ to update the Q table according to Line 8 from Algorithm 8.
6 Train and execute the Qlearning agent according to Lines 2-10 of Algorithm 8.

4.5.3 The Double Deep Q-Networks (DDQN) Approach

The traditional DDQN [67], as shown in Fig. 4.3 and Fig. 4.4, is an extension of the DQN algorithm of DRL which runs multiple episodes to allocate resources into RBs within a time frame T. The used notations are listed in Table 4.2. Here, the input and outputs to Algorithm 10 are similar to Algorithm 8.

At initialization, the primary network Q is initialized with random weights, while the target network $\hat{Q}$ is initialized by directly copying the weights from the primary network Q.

All the steps in lines 2-7 are similar to Algorithm 8 which have been explained earlier. In line 8, after choosing an action a_t, we receive a reward r_t for moving to the next state s_{t+1} which includes a list of next possible actions $\mathcal{A}_{t+1}$ from the environment. Line 9 stores those values obtained at time slot t as one single transition $(s, a, r, s', \mathcal{A})$ in the replay memory

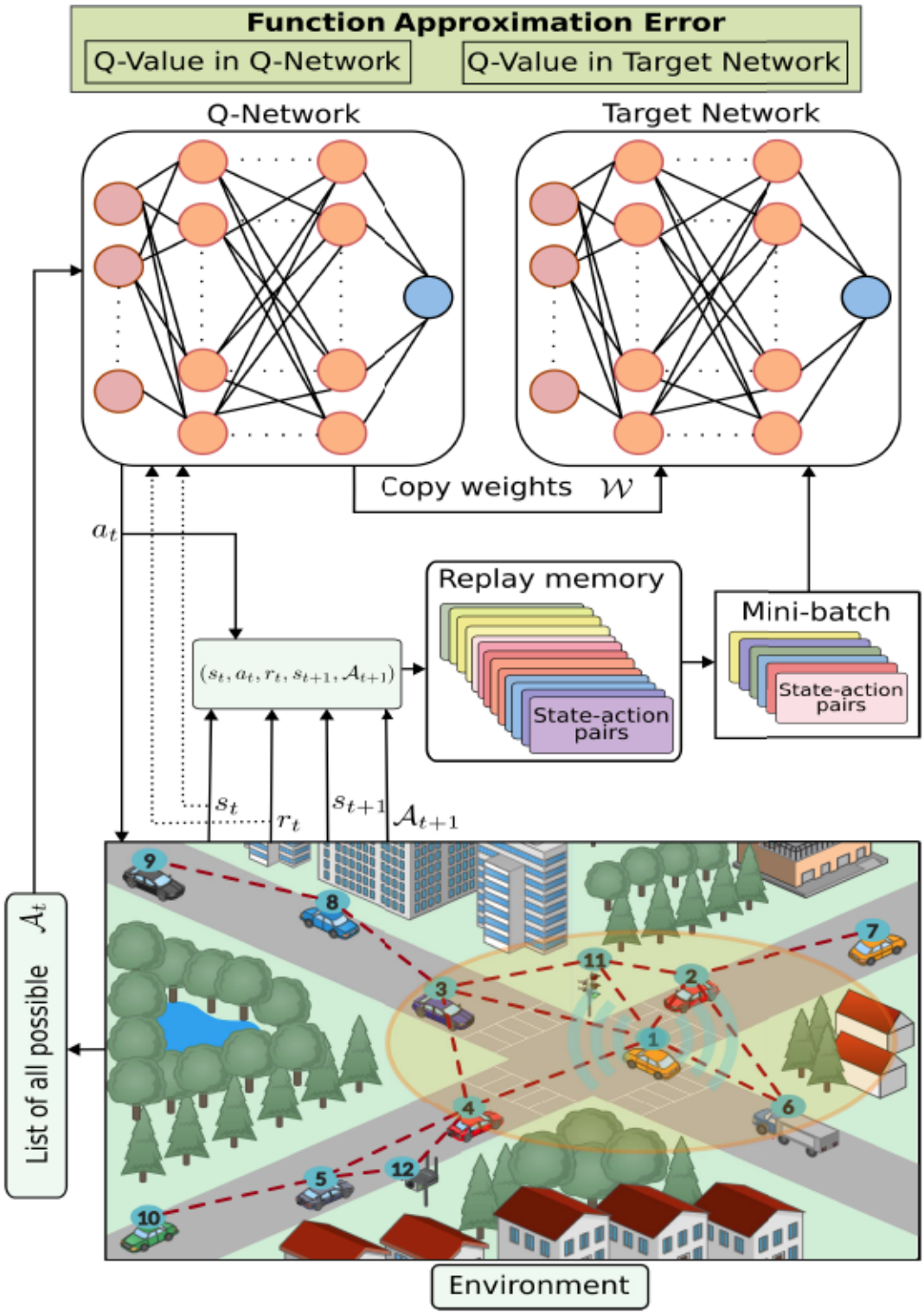


Figure 4.3: The proposed DRL approach to obtain the reward policy.

(RM). In line 10, a random minibatch of transitions $(s_j, a_j, r_j, s'_j, \mathcal{A}_j)$ is sampled from the

Table 4.2: SIMULATION PARAMETERS

Parameters	Values
Activation Functions	ReLU
Number of Neurons	256, 128, 64, 32, 16
Number of Hidden Layers	5
Learning Rate	0.0001
Optimizer	Adam
Total Number of Episodes	255
Decay Rate	1/Episodes * 2
Discounted Reward	0.99

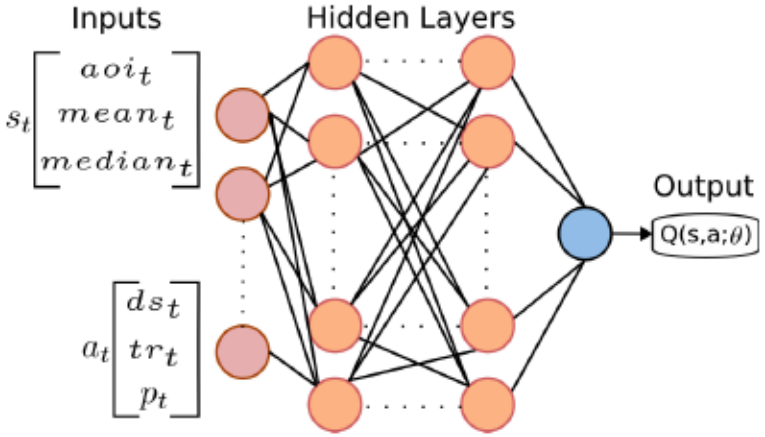


Figure 4.4: The proposed Q-Network.

replay memory RM. For each transition in the minibatch (line 11), the target Q-value $Q_t(s_j, a_j)$ is computed using the target network $\hat{Q}$. The $Q_t(s_j, a_j)$ is calculated as the sum of the immediate reward r_j and the discounted maximum Q-value $\hat{Q}(s'_j, \mathcal{A}_j)$ from the next state s'_j using the primary network Q (line 13). We then perform gradient descent on the difference between the target Q-value $Q_t(s_j, a_j)$ and the primary network Q (line 14). Next, target Q-network weights ($\hat{\theta}$) are updated. The update process involves adjusting the target network parameter θ' towards the primary network parameter θ, where the rate of averaging value τ is typically set to 0.01 (line 15). In this way, target network weights ($\hat{\theta}$) are updated by copying the weights from the primary network. Once the termination condition is met, the algorithm proceeds to the next episode until all episodes are completed. In the end, similar to Algorithm 1, the final agent is executed in the environment (line 16), and the total number of successful communications TNC and fairness F are calculated (line 17).

OAMM-DDQN

When the conventional DDQN has very large state s_t and action a_t spaces (refer to equations (4.32) and (4.33)), exploring the entire state and action spaces becomes challenging and

Algorithm 10: Proposed Double Deep Q networks-based solution

Data: Environment Interface
Result: TNC, F, RB
Initialize: $Q \leftarrow$ RandomWeights(), $\hat{Q} \leftarrow Q$
for $k \leftarrow 1 : K$ **do**
  for $t \leftarrow 1 : T$ *and* $s_t \neq s_T$ **do**
    $T_{ran} = \varnothing$, $R_{ec} = \varnothing$
    Observe s_t $(aoi_t, mean_t, median_t)$ from the environment
    Choose a_t (ds_t, tr_t, p_t) from a list of possible actions $\mathcal{A}_t$ using an ϵ-Greedy policy and allocate a_t into RB_t
    Obtain action a_t, reward r_t, next state s_{t+1}, and next action $\mathcal{A}_{t+1}$
    Store $(s_t, a_t, r_t, s'_{t+1}, \mathcal{A}_{t+1})$ as one transition in RM
    Sample random minibatch of transitions $(s, a, r, s', \mathcal{A})$ from RM
    for *Each transition* $(s_j, a_j, r_j, s'_j, \mathcal{A}_j)$ *in minibatch* **do**
      Compute target Q value using $\hat{Q}$ network:
      $Q_t(s_j, a_j) \leftarrow r_j + \gamma \cdot Q(s'_j, \arg\max_{\mathcal{A}} \hat{Q}(s'_j, \mathcal{A}))$
      Perform gradient descent step on: $(Q_t(s_j, a_j) - Q(s_j, a_j))^2$
      Update target network weights: $\hat{\theta} \leftarrow \tau \cdot \theta + (1 - \tau) \cdot \hat{\theta}$
Execute trained $DDQN$ agent on environment interface
Given terminal state $s_T, TotalAoI = \sum_{t=1}^{T} A^t$

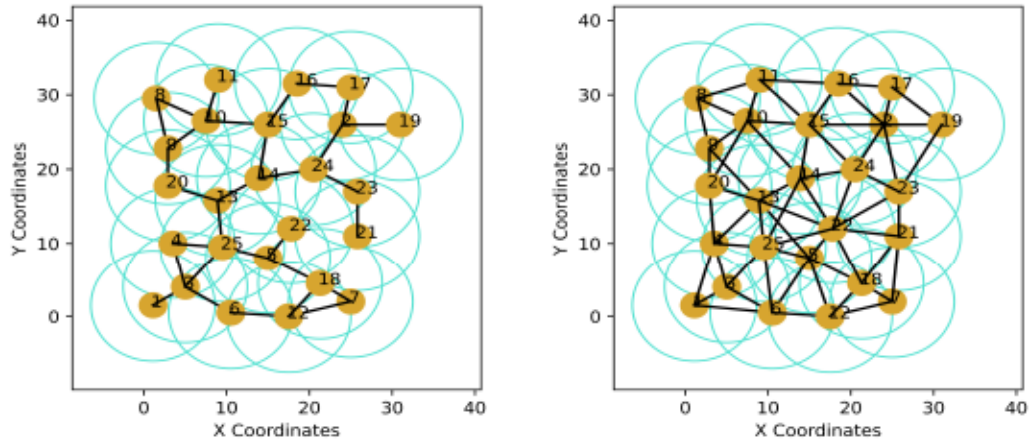
computationally expensive, which will lead to poor performance and slow learning convergence. To address such an issue, we introduce another heuristic-based DRL method called OAMM-DDQN. We initially find solutions by using the heuristic method for an episode and then copy the obtained solution of $OAMM$ method for 500 times. By leveraging the heuristic OAMM method, we sequentially populate the replay memory (RM) with a subset of promising states, actions, rewards, next states, and next actions. This targeted approach reduces the amount of time spent on exploration, allowing the agent to focus on learning from these high-quality states. Hence, the OAMM-DDQN agent explores and exploits the state-action space more effectively, since it understands which actions in specific states yield the best rewards, and accelerates learning convergence by leveraging prior knowledge.

4.6 Performance Evaluation

To evaluate the performance of the proposed methods, in this section, we first compare the Random (the explanations of the agent are explained below), the heuristic OAMM, Qlearning (Qlearn.), DDQN, the hybrid OAMM-Qlearning (OAMM-Qlearn.), and the hybrid OAMM-DDQN with the Optimum solutions obtained from the MILP-based optimization model on small networks. Then, we consider all of them except the optimization model

on both medium and large instances. The total sum of AoI of all data streams is taken as the performance metric, and we vary several parameters such as the number of nodes in the network, the number of time slots, the probability rate of packet generation, and the network density.

Fig. 4.5(a) and Fig. 4.5(b) illustrate an example of V2V networks, where the number of V2V nodes is considered as 25, and the V2V network density is set to 40% while varying the packet generation probability. In Fig. 4.5(a), we consider the probability of packet generation to be 0.25, which means only 25% data streams generate packets at any time slot, and more than half of the packets shown in Fig. 4.5(b) are generated by all data streams in the network. Here, each data stream broadcasts packets to its nearest neighbors within the communication range; otherwise, each acts as a relay node to broadcast or transmit the received packets to its destination node.



(a) (b)

Figure 4.5: Examples of medium networks while considering the V2V Nodes to 25 and network density to 40% by varying packet generation probability to: (a) 0.25, and (b) 0.50.

Random Method

In the Random method, the inputs, outputs, and all variables are considered the same as in the OAMM method, however, instead of sorting each data stream according to their sum of AoI and giving priority to the largest length of the broadcast/transmitter set with the newest packet ID, the data streams, transmitters, and packets are randomly stored.

For wireless communications, the threshold is considered as $\beta = 5dB$, the background noise as $\eta = -111dBm/Hz$, the power loss decay as $\alpha = 2.5$, the transmission power as $P = 20dB$ [68]. We use Python3 to simulate the operation of our algorithms and run on Intel(R) Xeon(R) CPU E5-2637 v4 @ 3.50GHz (2 processors) and 64.0 GB memory. The learning rate α and the discount factor γ required for the Q-value updates in the Qlearn.

and OAMM-Qlearn. are set to 0.0001 and 0.99 respectively. The number of episodes K for the Qlearn. and OAMM-Qlearn. was set to 15000 and 5000 respectively. The results are then averaged over five runs.

4.6.1 Evaluation Over Small Networks

In this subsection, the performance of different methods (Random, OAMM, Qlearn., DDQN, OAMM-Qlearn., OAMM-DDQN, and the optimal solution obtained from the optimization model: Optimum) is evaluated over the total sum of AoI by varying parameters, such as the number of V2V nodes, size of the time frame (the number of time slots), packet generation rate, and the network density (the number of links). For all evaluations, we fix the parameters unless otherwise stated: we set the number of nodes to 10, the number of time slots to 4, the probability of packet generation to 0.5 (to be noted that the probability of 1.0 means all data streams generate packets at any time slot), and the network density to 80% (to be noted that a 100% density means a fully connected network).

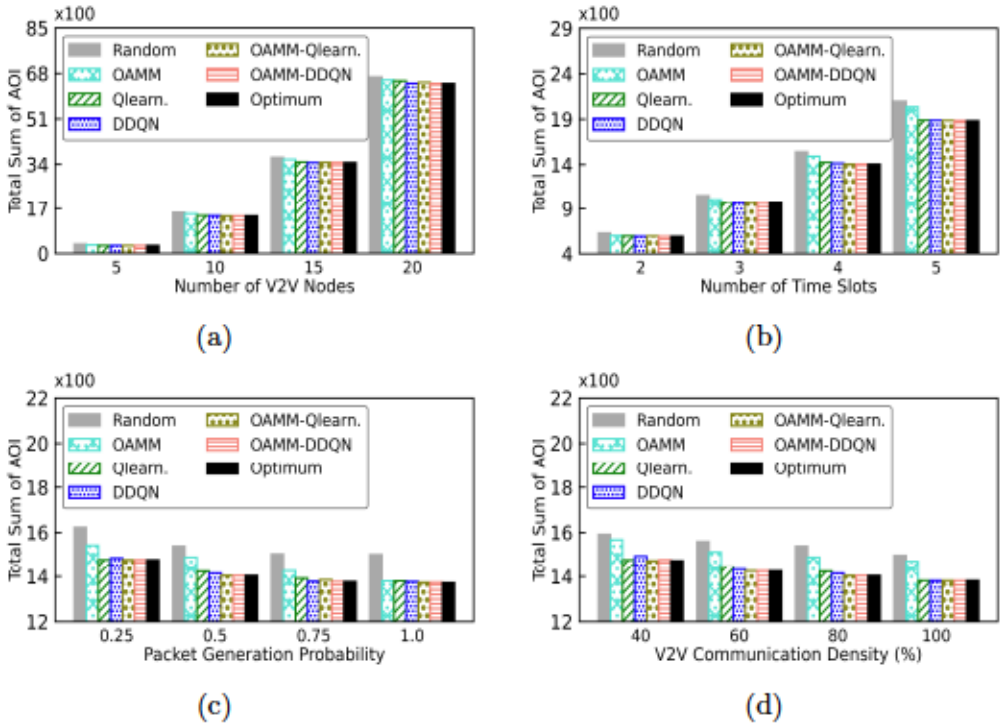


Figure 4.6: Total sum of AoI on small networks for all data streams as a comparison metric for different methods (Random, OAMM, Qlearn., DDQN, OAMM-Qlearn., OAMM-DDQN, and Optimum) by varying (a) number of V2V nodes, (b) number of time slots, (c) packet generation probability, and (d) V2V communication density.

Fig. 4.6(a) provides a visual representation of the total sum of AoI by varying the number of nodes. As depicted in the figure, the Qlearn., DDQN, and OAMM-Qlearn., and OAMM-

DDQN methods perform similarly to the Optimum in the very small network size of 5 nodes and achieve a total sum of AoI of 291. Whereas, the OAMM and Random methods achieve a comparatively higher sum of AoI due to not scheduling all packets in a sequence between the source and destination pairs. Later, as the size of the network increases, an upward trend in the total sum of AoI is observed for all, where with a network of size 20 nodes, only the OAMM-DDQN method has obtained similarly to the optimal solution which is a total sum of AoI of 6415. It is observed that the Random method performs the worse among all other methods because it poses challenges in maintaining the sequential scheduling of packets to reduce the total sum of AoI. Moreover, the hybrid OAMM-Qlearn. method gives optimal solutions for networks of size less than 20 nodes.

The obtained results shown in Fig. 4.6(b) illustrate the impact of varying the number of time slots on the total sum of AoI in small networks. As expected, it is observed that the total sum of AoI rises with a gradual increase in the number of time slots, as more time becomes available for the allocation of packets into RBs. Notably, when the number of time slots is set to 2, all methods except the Random method achieve an optimal solution of 590 total sums of AoI. However, with larger networks, the Qlearn., DDQN, OAMM-Qlearn., and OAMM-DDQN methods almost perform similarly to the Optimal solutions. Whereas, the OAMM method performs very close to the Optimum, and the Random method stands last.

Fig. 4.6(c) describes the variation in the total sum of AoI while varying the packet generation probability in small networks. As depicted, it is noticed that the total sum of AoI decreases with an increment in the total number of packet generation rates, as there are more available packets to be allocated into RBs. Notably, for different probabilities starting from 0.25 to 1.0, the Qlearn., the DDQN, the OAMM-Qlearn., and the OAMM-DDQN methods result in optimal solutions and perform equally to the Optimum. Whereas, the Random method performs worse than others with higher gaps to the Optimum, and the OAMM method results in lower gaps to the optimal solutions.

Fig. 4.6(d) offers meaningful insights into the total sum of AoI as the network density is varied. As depicted in the figure, the total sum of AoI exhibits a downward trend with increasing network density, which ensures the presence of more available links to be allocated into RBs. Notably, the OAMM-Qlearn., the DDQN and Optimum methods demonstrate their superior performance in terms of the total sums of AoI, achieving 1474 when the density of the V2V network is 40%. Whereas, the Random and OAMM methods start with a total sum of AoI scores of around 1591 and 1567, respectively. The simulation result eventually shows a decrement in the sum of AoI scores of 1498 and 1471, respectively when the network density becomes 100%. However, despite this accomplishment, the Random and OAMM methods pale in comparison to Qlearn., DDQN, OAMM-Qlearn., OAMM-DDQN, and Optimum methods in terms of the total sum of AoI.

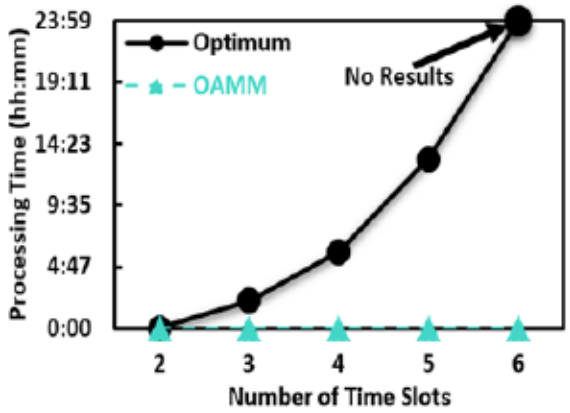


Figure 4.7: Computation time of optimization model (Optimum) vs our proposed heuristic method (OAMM).

Fig. 4.7 shows the computation time for both Optimum and OAMM methods by varying the number of time slots from 2 to 6. By increasing the number of time slots, the optimization model requires more time to execute, and the processing time grows exponentially, while the execution time of the heuristic method is in milliseconds. The optimization model failed to obtain results for 6 and higher time slots due to a lack of CPU memory. Hence, the optimization model is not scalable or good for large networks. Therefore, in the next subsection, we evaluate the performance of our proposed methods without the optimization model using medium and very large instances.

4.6.2 Evaluation Over Medium Networks

In this subsection, the performance of different methods (Random, OAMM, Qlearn., DDQN, and OAMM-Qlearn.) is evaluated over medium networks. For all evaluations, we fix the parameters unless otherwise stated: we set the number of nodes to 50, the number of time slots to 40, the probability of the packet generation to 0.5, and the network density to 80%.

Fig. 4.8(a) provides insights into the total sum of AoI when the number of nodes is varied from 25 to 100 nodes. As depicted in the figure, a total sum of AoI of 350149 is achieved by the OAMM-Qlearn. method while considering twenty-five nodes. Whereas, with the Qlearn. and OAMM methods, we achieve a comparatively higher score due to not scheduling all packets in a sequence between the source and destination pairs. Afterward, with the gradual increase in the number of network nodes, an upward trend in the total sum of AoI is observed for all, where OAMM-Qlearn. generates the lower total sum of AoI (equal to 7882780) compared to other introduced methods when the number of nodes is 100. It is observed that the Random method generates the worst result among all methods. This can be attributed to the growing number of broadcasted nodes with an increasing number of nodes in the network. Moreover, the OAMM-Qlearn. method minimizes the total sum

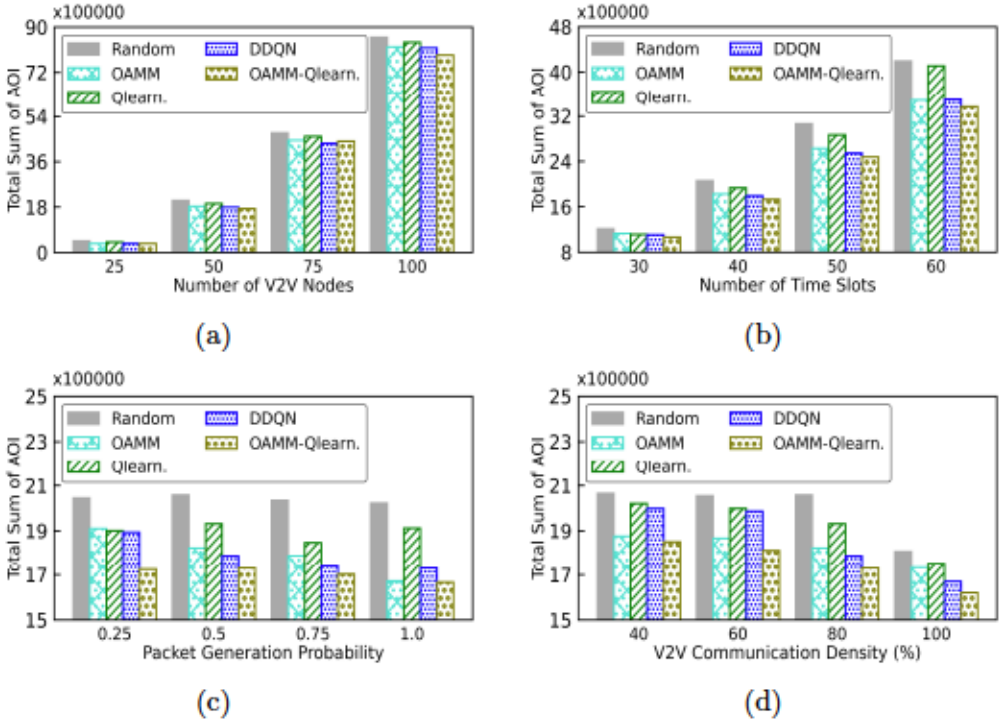


Figure 4.8: Total sum of AoI on medium networks for all data streams as a comparison metric for different methods (Random, OAMM, Qlearn., DDQN, and OAMM-Qlearn.) by varying (a) number of V2V nodes, (b) number of time slots, (c) packet generation probability, and (d) V2V communication density.

of AoI compared to the OAMM and Qlearn. over the experiment.

With a gradual rise in the number of time slots, there is an expected increment in the total sum of AoI among all methods, as shown in Fig. 4.8(b). When there are 30 time slots, the OAMM-Qlearn. method achieves a total sum of AoI of just over 1064058. In contrast, the Qlearn., and OAMM methods achieve a higher total sum AoI around 1111870, and 1128983, respectively. As anticipated, it is observed that the total sum AoI rises with an increase in the number of time slots, as more time becomes available for the allocation of packets into resource blocks. By increasing the number of time slots to 60, both Qlearn. and OAMM fail to minimize the total sum of AoI when compared to the OAMM-Qlearn. method which reach a total sum of 3360078, respectively. Moreover, the Random method performs the worst compared to other proposed methods and ended up with a total sum of 4190766 when dealing with 60 time slots.

Fig. 4.8(c) illustrates the variation in the total sum of AoI while varying the probability of packet generation in large networks. It is noticed that the total sum of AoI decreases with an increment in the total number of packet generation rates, as there are more available packets to be allocated into resource blocks. Notably, when the probability of packet generation is

set to 0.25, the OAMM-Qlearn. method obtains a total sum AoI of around 1724190 and 1720090 respectively, whereas the Random and OAMM methods achieve a comparatively higher sum AoI. By increasing the number of packet generation rates to 1.0, the OAMM-Qlearn. generates a lower total sum AoI, whereas the worst result (i.e., a total sum AoI of 2021438) is observed while considering the Random method.

Fig. 4.8(d) offers insights into the total sum of AoI as the density of the network varies from 40% to 100%. As depicted in the figure, the total sum AoI decreases as the network density becomes denser. The reason is because there are more links available to be allocated into RBs. Notably, the OAMM-Qlearn. method demonstrates superior performance in terms of the total sum AoI, achieving 1850256 when the density of the network is 40%, while the Random and Qlearn. methods start with a total sum AoI of around 2064312 and 2019631 respectively, and it respectively decreases to $TotalAoI$ = 1807890 and 1753598 when the network density becomes 100%. However, despite this accomplishment, the Random and Qlearn. methods pale in comparison to OAMM-Qlearn. in terms of the total sum of AoI.

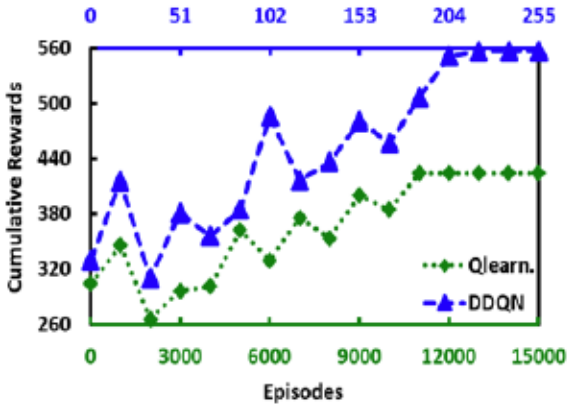


Figure 4.9: Learning curves of Reinforcement Learning algorithms: Qlearn. vs DDQN.

Fig. 4.9 shows the learning curve of the conventional Qlearn. and DDQN. The x-axis represents the episode number, while the y-axis represents the cumulative rewards. The results were obtained using a network with 25 nodes with 80% link connectivity, using a time frame divided into 40 equal time slots, with a packet generation probability of 50%. At the beginning of the episode, the DDQN. method begins with a higher cumulative reward of around 329, while the Qlearn. method starts with a cumulative reward of approximate 303. This inequality arises because the DDQN method uses neural networks to train a batch of samples, which yields a more promising initial result. As both techniques continue to explore the environment, they gradually achieve more rewards over time. However, the DDQN approach shows faster learning compared to the Qlearn. approach; it rapidly adapts to the environment, obtaining higher cumulative rewards at earlier episodes. As the learning curves progress, the performance gap between the two methods becomes more

visible. By the end of training, particularly at episode 255 and 15000 for the DDQN and Qlearn. methods respectively, the DDQN method outperforms the Qlearn. method with a cumulative reward gap of 132; the DDQN method obtains a cumulative reward of 555, while the Qlearn. method reaches a cumulative reward of 423.

4.6.3 Evaluation Over Large Networks

In this subsection, the performance of different methods (Random, and OAMM) is described over the total sum of AOI by varying other parameters, such as number of V2V nodes, time frame (number of time slots), packet generation probability, and the density of links in the network. For all evaluations, we fix the parameters unless otherwise stated: we set the number of nodes to 600, time slots to 100, packet generation probability to 0.5, and the density of links in the network to 80%.

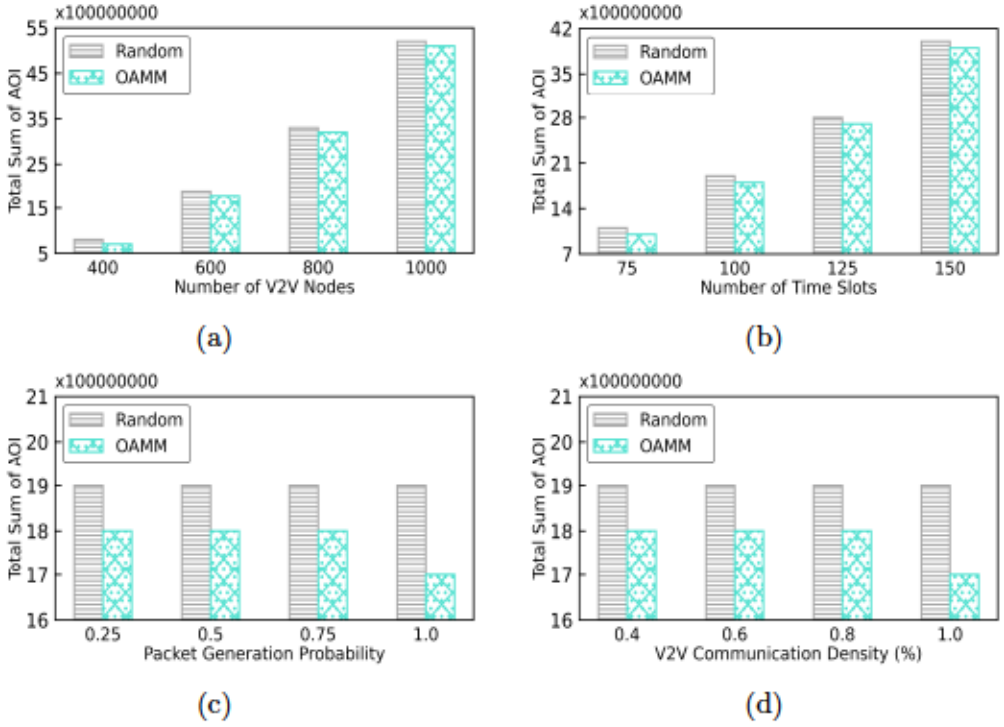


Figure 4.10: Total sum of AoI on Large networks for all data streams as a comparison metric for different methods (Random, and OAMM) by varying (a) number of V2V nodes, (b) number of time slots, (c) packet generation probability, and (d) V2V communication density.

Fig. 4.10(a) respectively provides insights into the total number of sum of AoI when the number of nodes is varied. As depicted in the figure, a total sum of AoI (i.e., 790361201) is achieved by the OAMM method while considering 400 nodes, whereas with the Random, we achieve comparatively higher sum of AoI (i.e., 812522658) due to not scheduling all packets

in a sequence of between source and destination pairs. Afterward, with the gradual increase in growing number of broadcasted nodes while increasing the number of V2V nodes, an upward trend in the number of total sum of AoI is observed for both of them, where the OAMM generates a lower sum of AoI result (i.e., 5130333402) compared to the Random when the number of nodes is 1000. Moreover, the OAMM method minimizes the total sum of AoI compared to the Random method over the experiment.

With a gradual rise in the number of time slots, there is an expected increment in the total number of sum of AoI among all methods, as shown in Fig. 4.10(b). When there are 75 time slots, the OAMM method achieves a total sum of AoI (i.e., 1050069146). In contrast, the Random achieves a higher number of sum of AoI (i.e., 1163902935). As anticipated, it is observed that the total sum of AoI rises with an increase in the number of time slots, as more time becomes available for the allocation of packets into RBs. By increasing the number of time slots to 150, the Random method fails to minimize the total sum of AoI when compared to the OAMM method that reaches a total of 3930687320 sum of AoI. Moreover, the Random method shows a worse result compared to the OAMM method and ended up with a sum of AoI (i.e., 4069855458) when dealing with 150 time slots.

Fig. 4.10(c) illustrates the variation in the total sum of AoI while varying the packet generation probability in large networks. It is noticed that the total sum of AoI decreases with an increment in the total number of packet generation rate, as there are more available packets to be allocated into RBs. Notably, when the number of packet generation probability is set to 0.25, the OAMM method obtains a total sum of AoI (i.e., 1834802846), whereas Random method achieves a comparatively higher number of sum of AoI (i.e., 1959060866). When the number of packet generation rate increases to 1.0, the OAMM method generates a total sum of AoI (i.e., 1782033388), whereas a higher number of total sum of AoI (i.e., 1950648500) is observed while considering the Random method.

Fig. 4.10(d) offers meaningful insights into the total number of sum of AoI as the density of network is varied. As depicted in the figure, the total sum of AoI exhibits a downward trend with increasing network density. The reason is because there are more links available to be allocated into RBs. Notably, the OAMM method demonstrates superior performance in terms of the total sum of AoI, achieving 1816636976 when the density of V2V network is 40%, while the Random method starts with a total sum of AoI score of around 1960350679, the obtained results eventually show an decrement in the sum of AoI score with $TAoI =$ 1807890 and 1753598, respectively, when the network density becomes 100%. However, despite this accomplishment, the Random method pales in comparison to OAMM in terms of total sum of AoI.

The OAMM method always outperforms the random method for large networks because it sorts each data stream according to their sum of AoI and giving priority to the largest length of the broadcast/transmitter set with the newest packet ID.

4.7 Summary

This chapter has proposed novel and effective approaches to address the critical issue of the age of information (AoI) minimization in AV-assisted vehicular networks. The main objective is to minimize the total or average AoI of all data streams for autonomous vehicles, taking into account resource allocation under the half-duplex constraint, traversal of broadcasted nodes between sources and destinations, link scheduling, and the reuse of resource blocks. The problem was mathematically formulated using Mixed Integer Linear Programming (MILP) to obtain optimal solutions on small networks. However, due to the complexity of the optimization model, a scalable heuristic method named the Online Age of Information Minimization Method (OAMM) was introduced to efficiently solve the problem for large networks. To address the dynamic nature of the environment, the problem was further modeled as a Markov Decision Process (MDP) and solved using Qlearning, a reinforcement learning algorithm. The integration of the OAMM-Qlearning hybrid approach resulted in significant improvements in minimizing the expected weighted total or average AoI. This achievement is crucial as it enables the delivery of time-sensitive and reliable data streams for various autonomous vehicle applications. The performance of the proposed OAMM-Qlearning approach was extensively evaluated through simulations, demonstrating its effectiveness and practicality in real-world scenarios. By contributing innovative solutions for AoI minimization in AV-assisted vehicular networks, this research contributes to the advancement of intelligent transportation systems and smart cities. The proposed approaches hold great potential in enhancing the efficiency and reliability of data transmissions for autonomous vehicles, paving the way for safer and more responsive autonomous driving experiences. As autonomous vehicles continue to play a crucial role in shaping the future of transportation, the findings of this study have valuable implications for the development of efficient and intelligent vehicular communication systems.

Chapter 5

Conclusion

Intelligent Transportation Systems (ITS) have become essential not only for the proliferation of autonomous driving but also for facilitating real-time data exchange among vehicles, utilizing the capabilities of wireless communication. This dynamic exchange of real-time decision-making information through V2V communications ensures efficient, reliable, safe, and comfortable driving experiences, while simultaneously enhancing overall traffic safety and efficiency.

Throughout this thesis, we embarked on the exploration of two distinct projects, each with its unique objectives. The first project focused on the maximization of communication instances within groups of vehicles, all while maintaining equity among V2V communication pairs. This intricate challenge required us to address several critical factors, including multi-hop routing path determination, transmission power control, link scheduling, and resource reuse, all within the constraints of Half-Duplex and Signal-to-Interference-Plus-Noise Ratio (SINR).

The second project delved into the minimization of the Age of Information (AoI) across all data streams within autonomous vehicular-assisted networks, considering a specific time-frame. This problem encompassed the optimization of data relay participation, transmission timings, and data packet selection, all while reusing resources within the context of Half-Duplex and SINR constraints.

To tackle both of these complex challenges, we initiated our approach by mathematically formulating the problems as Mixed Integer Linear Programming (MILP) models, thereby obtaining optimal solutions. However, due to their computational complexity, we introduced scalable heuristic methods tailored to address the needs of larger networks. Furthermore, recognizing the dynamic nature of the environment and the mobility of vehicles, we embraced a Markov Decision Process (MDP) framework. Within this framework, we harnessed the power of two reinforcement learning (RL) algorithms - Q-learning and Double Deep Q-Networks (DDQN) - to provide solutions.

Additionally, we augmented the learning capabilities of the RL agents and overall system performance by proposing innovative hybrid heuristic-based RL methods, seamlessly integrating the strengths of the RL agents with our introduced heuristic strategies.

Through comprehensive numerical simulations, we have conclusively demonstrated the efficacy of these hybrid approaches in solving both problems. Our methods outperformed random agents and introduced heuristic techniques, and conventional RL methods across varied network sizes. Moreover, our hybrid approaches exhibited scalability, achieving a remarkable worst performance gap of less than 5% when compared to optimal solutions on small networks.

The findings of this research have the potential to significantly contribute to the advancement of intelligent transportation systems and smart cities. By enhancing driving experiences through reliable, safer, and more responsive vehicular communications, we pave the way for a future marked by seamless and efficient mobility.

Bibliography

[1] M. Mao, P. Yi, and T. Hu, "Roadside infrastructure deployment scheme based on internet of vehicles information service demand," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 1, p. e4671, 2023.

[2] S. Zhao, B. Gui, G. Chen, and B. Yang, "On rate fairness maximization of vehicular networks: A deep reinforcement learning approach," in *2022 International Conference on Networking and Network Applications (NaNA)*, 2022, pp. 113–118.

[3] X. Bi, X. Sun, Z. Lyu, B. Zhang, and X. Wei, "A back adjustment based dependent task offloading scheduling algorithm with fairness constraints in vec networks," *Computer Networks*, vol. 223, p. 109552, 2023.

[4] B. L. Nguyen, D. T. Ngo, M. N. Dao, V. N. Q. Bao, and H. L. Vu, "Scheduling and power control for connectivity enhancement in multi-hop i2v/v2v networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 322–10 332, 2022.

[5] A. Al-Hilo, D. Ebrahimi, S. Sharafeddine, and C. Assi, "Vehicle-assisted rsu caching using deep reinforcement learning," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2021.

[6] N. Waqar, S. A. Hassan, A. Mahmood, K. Dev, D.-T. Do, and M. Gidlund, "Computation offloading and resource allocation in mec-enabled integrated aerial-terrestrial vehicular networks: A reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 478–21 491, 2022.

[7] A. J. M. Muzahid, S. F. Kamarulzaman, M. A. Rahman, S. Murad, M. A. S. Kamal, and A. H. Alenezi, "Multiple vehicle cooperation and collision avoidance in automated vehicles: survey and an ai-enabled conceptual framework," *Scientific Reports*, 2023.

[8] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2092–2104, 2020.

[9] Z. Deng, Z. Cai, and M. Liang, "A multi-hop vanets-assisted offloading strategy in vehicular mobile edge computing," *IEEE Access*, vol. 8, pp. 53 062–53 071, 2020.

[10] C.-M. Huang, S.-Y. Lin, and Z.-Y. Wu, "The k-hop-limited v2v2i vanet data offloading using the mobile edge computing (mec) mechanism," *Vehicular Communications*, vol. 26, p. 100268, 2020.

[11] I. Turcanu, P. Salvo, A. Baiocchi, F. Cuomo, and T. Engel, "A multi-hop broadcast wave approach for floating car data collection in vehicular networks," *Vehicular Communications*, vol. 24, p. 100232, 2020.

[12] Y. Sun, L. Xu, Y. Tang, and W. Zhuang, "Traffic offloading for online video service in vehicular networks: A cooperative approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7630–7642, 2018.

[13] J. Mei, K. Zheng, L. Zhao, Y. Teng, and X. Wang, "A latency and reliability guaranteed resource allocation scheme for lte v2v communication systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3850–3860, 2018.

[14] A. Masmoudi, K. Mnif, and F. Zarai, "A survey on radio resource allocation for v2x communication," *Wireless Communications and Mobile Computing*, vol. 2019, p. 12, 2019.

[15] S. Roger, D. Martín-Sacristán, D. Garcia-Roger, J. F. Monserrat, P. Spapis, A. Kousaridas, S. Ayaz, and A. Kaloxylos, "Low-latency layer-2-based multicast scheme for localized v2x communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2962–2975, 2019.

[16] M. Sadeghi, E. Björnson, E. G. Larsson, C. Yuen, and T. Marzetta, "Joint unicast and multi-group multicast transmission in massive mimo systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6375–6388, 2018.

[17] A. Masmoudi, S. Feki, K. Mnif, and F. Zarai, "Efficient scheduling and resource allocation for d2d-based lte-v2x communications," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2019, pp. 496–501.

[18] A. B. Tambawal, R. M. Noor, R. Salleh, C. Chembe, M. H. Anisi, O. Michael, and J. Lloret, "Time division multiple access scheduling strategies for emerging vehicular ad hoc network medium access control protocols: A survey," *Telecommun. Syst.*, vol. 70, no. 4, p. 595–616, 2019.

[19] Y. Chen, "Application of 5g mobile communication technology integrating robot controller communication method in communication engineering," *Journal of Robotics*, vol. 2023, no. 8, pp. 7630–7642, 2023.

[20] A. K. Kazi, S. M. Khan, and N. G. Haider, "Reliable group of vehicles (rgov) in vanet," *IEEE Access*, vol. 9, pp. 111 407–111 416, 2021.

[21] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, "Age of information aware trajectory planning of uavs in intelligent transportation systems: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 382–12 395, 2020.

[22] E. Fountoulakis, T. Charalambous, A. Ephremides, and N. Pappas, "Scheduling policies for aoi minimization with timely throughput constraints," *IEEE Transactions on Communications*, vol. 71, no. 7, pp. 3905–3917, 2023.

[23] X. Xie, H. Wang, and X. Liu, "Scheduling for minimizing the age of information in multi-sensor multi-server industrial iot systems," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2023.

[24] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, "A reinforcement learning framework for optimizing age of information in rf-powered communication systems," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4747–4760, 2020.

[25] S. Leng and A. Yener, "An actor-critic reinforcement learning approach to minimum age of information scheduling in energy harvesting networks," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8128–8132.

[26] M. Zhang, Y. Dou, P. H. J. Chong, H. C. B. Chan, and B.-C. Seet, "Fuzzy logic-based resource allocation algorithm for v2x communications in 5g cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2501–2513, 2021.

[27] Y. Hou, X. Wu, X. Tang, X. Qin, and M. Zhou, "Radio resource allocation and power control scheme in v2v communications network," *IEEE Access*, vol. 9, pp. 34 529–34 540, 2021.

[28] G. G. Lema, "Performance evaluation of beamforming for network throughput enhancement," *International Journal of Communication Systems*, vol. 33, no. 16, 2020.

[29] M. Sharara, S. Hoteit, V. Vèque, and F. Bassi, "Minimizing power consumption by joint radio and computing resource allocation in cloud-ran," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, 2022, pp. 1–6.

[30] X. Fan, X. Fan, D. Liu, B. Fu, and S. Wen, "Optimal relay selection for uav-assisted v2v communications," *Wireless Networks*, 2021.

[31] D. Ebrahimi, H. Elbiaze, and W. Ajib, "Device-to-device data transfer through multi-hop relay links underlaying cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9669–9680, 2018.

[32] X. Wu, Y. Hou, X. Tao, and X. Tang, "Maximization of con-current links in v2v communications based on belief propagation," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[33] D. Ebrahimi and C. Assi, "On the interaction between scheduling and compressive data gathering in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2845–2858, 2016.

[34] X. Li, L. Ma, R. Shankaran, Y. Xu, and M. A. Orgun, "Joint power control and resource allocation mode selection for safety-related v2x communication," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7970–7986, 2019.

[35] X. Wu, Y. Hou, X. Tao, and X. Tang, "Maximization of con-current links in v2v communications based on belief propagation," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[36] N. Geng, Q. Bai, C. Liu, T. Lan, V. Aggarwal, Y. Yang, and M. Xu, "A reinforcement learning framework for vehicular network routing under peak and average constraints," *IEEE Transactions on Vehicular Technology*, pp. 1–11, 2023.

[37] X. Li, L. Lu, W. Ni, A. Jamalipour, D. Zhang, and H. Du, "Federated multi-agent deep reinforcement learning for resource allocation of vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 8, pp. 8810–8824, 2022.

[38] Y. Yuan, G. Zheng, K.-K. Wong, and K. B. Letaief, "Meta-reinforcement learning based resource allocation for dynamic v2x communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8964–8977, 2021.

[39] C. Guo, C. Wang, L. Cui, Q. Zhou, and J. Li, "Radio resource management for c-v2x: From a hybrid centralized-distributed scheme to a distributed scheme," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 4, pp. 1023–1034, 2023.

[40] J. Tian, Q. Liu, H. Zhang, and D. Wu, "Multiagent deep-reinforcement-learning-based resource allocation for heterogeneous qos guarantees for vehicular networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1683–1695, 2022.

[41] M. Parvini, A. Gonzalez, A. Villamil, P. Schulz, and G. Fettweis, "Joint resource allocation and string-stable cacc design with multi-agent reinforcement learning," in *2023 International Conference on Communications (ICC 2023)*, 2023.

[42] P. Dai, Y. Huang, K. Hu, X. Wu, H. Xing, and Z. Yu, "Meta reinforcement learning for multi-task offloading in vehicular edge computing," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2023.

[43] Y. Ju, H. Wang, Y. Chen, T.-X. Zheng, Q. Pei, J. Yuan, and N. Al-Dhahir, "Deep reinforcement learning based joint beam allocation and relay selection in mmwave vehicular networks," *IEEE Transactions on Communications*, pp. 1–1, 2023.

[44] X. Fan, B. Liu, C. Huang, S. Wen, and B. Fu, "Utility maximization data scheduling in drone-assisted vehicular networks," *Computer Communications*, vol. 175, pp. 68–81, 2021.

[45] B. L. Nguyen, D. T. Ngo, M. N. Dao, Q.-T. Duong, and M. Okada, "A joint scheduling and power control scheme for hybrid i2v/v2v networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 668–15 681, 2020.

[46] J. Huang, Y. Yang, Z. Gao, D. He, and D. W. K. Ng, "Dynamic spectrum access for d2d-enabled internet of things: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17 793–17 807, 2022.

[47] G. G. M. N. Ali, M. A. S. Mollah, S. K. Samantha, P. H. J. Chong, Y. L. Guan, Y. L. Guan, and Y. L. Guan, "On striking the balance between the fairness of service and throughput in roadside units based vehicular ad hoc networks," *International Journal of Vehicle Autonomous Systems*, 2016.

[48] A. Di Maio, R. Soua, M. R. Palattella, and T. Engel, "Roadnet: Fairness- and throughput-enhanced scheduling for content dissemination in vanets," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018, pp. 1–6.

[49] Z. Tariq, H. Z. Khan, U. Fakhar, M. Ali, A. N. Akhtar, M. Naeem, and A. Wakeel, "Fairness-based user association and resource blocks allocation in satellite–terrestrial integrated networks," *Physical Communication*, vol. 55, p. 101934, 2022.

[50] G. Chen, Y. Chen, J. Wang, and J. Song, "Minimizing age of information in downlink wireless networks with time-varying channels and peak power constraint," *IEEE Transactions on Vehicular Technology*, pp. 1–12, 2023.

[51] I. Kadota, M. Rahman, and E. Modiano, "Wifresh: Age-of-information from theory to implementation," in *2021 International Conference on Computer Communications and Networks (ICCCN)*, 12 2020, pp. 1–11.

[52] A. Muhammad, I. Sorkhoh, M. Samir, D. Ebrahimi, and C. Assi, "Minimizing age of information in multiaccess-edge-computing-assisted iot networks," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 052–13 066, 2022.

[53] Q. Chen, S. Guo, W. Xu, Z. Cai, L. Cheng, and H. Gao, "Aoi minimization charging at wireless-powered network edge," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, 2022, pp. 713–723.

[54] Y.-P. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, pp. 2903–2915, 2020.

[55] J. Liu, R. Zhang, A. Gong, and H. Chen, "Optimizing age of information in wireless uplink networks with partial observations," *IEEE Transactions on Communications*, pp. 1–1, 2023.

[56] W. Jin, J. Sun, K. Chi, and S. Zhang, "Deep reinforcement learning based scheduling for minimizing age of information in wireless powered sensor networks," *Computer Communications*, vol. 191, pp. 1–10, 2022.

[57] T. Wu, P. Wen, and S. Tang, "Optimal scheduling strategy of auv based on importance and age of information," *Wireless Networks*, vol. 29, pp. 87–95, 2023.

[58] P. Zou, J. Zhang, and S. Subramaniam, "Performance modeling of scheduling algorithms in a multi-source status update system," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 156–161.

[59] Y. He, G. Chen, Y. Chen, J. Wang, and J. Song, "Scheduling algorithms for minimizing the age of synchronization in wireless networks with random updates under throughput constraints," in *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2022, pp. 1–6.

[60] M. E. Kabir, I. Sorkhoh, B. Moussa, and C. Assi, "Joint routing and scheduling of mobile charging infrastructure for v2v energy transfer," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 736–746, 2021.

[61] Y. Zhu, Y. Deng, and Q. Ji, "A model simulation and analyses for resource allocation scheme in v2v communications with deep q network," in *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2021, pp. 0681–0688.

[62] R. Atallah, M. Khabbaz, and C. Assi, "Multihop v2i communications: A feasibility study, modeling, and performance analysis," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2801–2810, 2017.

[63] Q. Luo, C. Li, T. H. Luan, T. H. Luan, T. H. Luan, and W. Shi, "Collaborative data scheduling for vehicular edge computing via deep reinforcement learning," *IEEE Internet of Things Journal*, 2020.

[64] B. Lee and W. Shin, "Max-min fairness precoder design for rate-splitting multiple access: Impact of imperfect channel knowledge," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1355–1359, 2023.

[65] M. Kadivar and N. Mohammadi, "A maximum clique based approximation algorithm for wireless link scheduling under sinr model," *Journal of Computer and System Sciences*, 2022.

[66] S. K. Mohanty, S. K. Udgata, and S. K. Udgata, "Satpas: Sinr-based adaptive transmission power assignment with scheduling in wireless sensor network," *Engineering Applications of Artificial Intelligence*, 2021.

[67] W. Zaher and et al., "Omnidirectional-wheel conveyor path planning and sorting using reinforcement learning algorithms," *IEEE Access*, vol. 10, pp. 27 945–27 959, 2022.

[68] D. Ebrahimi, P. Ghosh, F. Alzhouri, and T. E. Alves De Oliveira, "Maximizing data communications within groups of vehicles while maintaining fairness," 2023. [Online]. Available: http://dx.doi.org/10.13140/RG.2.2.34779.34081

[69] H. Shokri-Ghadikolaei, C. Fischione, and E. Modiano, "On the accuracy of interference models in wireless communications," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.

[70] T. He, K.-W. Chin, Z. Zhang, T. Liu, and J. Wen, "Optimizing information freshness in rf-powered multi-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 9, pp. 7135–7147, 2022.

[71] M. Li, C. Chen, H. Wu, X. Guan, and X. Shen, "Edge-assisted spectrum sharing for freshness-aware industrial wireless networks: A learning-based approach," *IEEE Transactions on Wireless Communications*, vol. 21, no. 9, pp. 7737–7752, 2022.

[72] H. Wang, X. Xie, and J. Yang, "Optimizing average age of information in industrial iot systems under delay constraint," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2023.

[73] K. Chen, F. Benkhelifa, H. Gao, J. A. McCann, and J. Li, "Minimizing age of information in multihop energy-harvesting wireless sensor network," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 736–25 751, 2022.

[74] Q. Chen, S. Guo, W. Xu, Z. Cai, L. Cheng, and H. Gao, "Aoi minimization charging at wireless-powered network edge," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, 2022, pp. 713–723.