Network Slicing with MEC and Deep Reinforcement Learning for the Internet of Vehicles

Zoubeir Mlika, *Member, IEEE*, and Soumaya Cherkaoui, *Senior Member, IEEE*

Abstract—The interconnection of vehicles in the future fifth generation (5G) wireless ecosystem forms the so-called Internet of vehicles (IoV). IoV offers new kinds of applications requiring delay-sensitive, compute-intensive and bandwidth-hungry services. Mobile edge computing (MEC) and network slicing (NS) are two of the key enabler technologies in 5G networks that can be used to optimize the allocation of the network resources and guarantee the diverse requirements of IoV applications.

As traditional model-based optimization techniques generally end up with NP-hard and strongly non-convex and non-linear mathematical programming formulations, in this paper, we introduce a model-free approach based on deep reinforcement learning (DRL) to solve the resource allocation problem in MEC-enabled IoV network based on network slicing. Furthermore, the solution uses non-orthogonal multiple access (NOMA) to enable a better exploitation of the scarce channel resources. The considered problem addresses jointly the channel and power allocation, the slice selection and the vehicles selection (vehicles grouping). We model the problem as a single-agent Markov decision process. Then, we solve it using DRL using the well-known DQL algorithm. We show that our approach is robust and effective under different network conditions compared to benchmark solutions.

I. Introduction

The Internet of vehicles (IoV) is an emerging concept that enhances the existing capabilities of vehicular communication by integrating with the Internet of things (IoT). IoV is a key use-case in the upcoming beyond fifth generation (5G) wireless networks [1, 2]. IoV creates diverse new applications with extremely diverse service requirements including ultra-high reliable and delay-sensitive, bandwidth-hungry as well as compute-intensive applications [3]. For example, accident reports require ultra-reliable and extremely low latency whereas high definition map sharing require high bandwidth. An important open question in today's IoV networks is "how to support, using a unified air interface, future IoV services while guaranteeing their extremely diverse performance requirements?" Network slicing (NS) is a potential solution to respond to this question [4–6]. NS is a tool that enables network operators to support virtualized end-to-end networks that belongs to the principle of software defined networking [7]. It mainly allows creating different logical networks on the top of a common and programmable physical infrastructure. Another technology, namely mobile edge computing, or better known as multi-access edge computing (MEC), is considered as an

important building block in the future IoV ecosystem. The joint implementation of NS and MEC is a key enabler for IoV networks. These two technologies can be used not only to guarantee the diverse requirements of IoV applications but also to deploy the diverse vehicular services at the appropriate locations [3].

Optimal resource allocation in IoV would go through traditional model-based optimization techniques. Due to the complex and highly dynamic nature of IoV, such a model-based approach is not very appealing. In fact, such approach ends up with strongly non-convex optimization problems that are generally NP-hard [8]. Thus, a model-free machine learning approach is crucial.

Reinforcement learning (RL) is a useful technique in solving NP-hard optimization problems. It has been applied successfully to solve very hard problems in different research areas including wireless networks [9]. It is based on Markov decision process (MDP) modeling where agents learn to select the best actions through repeated interactions with an unknown environment by receiving numerical reward signals [8]. Deep RL (DRL) uses the strong ability of neural networks to generalize across enormous state spaces and reduce the complexity of a solution, thus improving the learning process.

In this paper, using DRL, we propose a new solution framework to solve the challenging problem of resource allocation in a MEC-enabled IoV network. More specifically, we focus on the in-coverage scenario of 5G-new radio (5G-NR) in which vehicles communicate with each other through a base station, e.g., NodeB (gNB), that performs MEC-based tasks [10]. We focus on the broadcast communication technique. Due to the scarce spectrum resources, non-orthogonal multiple access (NOMA) is also used in our proposed framework. NOMA is a promising technique to increase the spectral efficiency in vehicular networks [11].

In more detail, the considered resource allocation problem, called IoV resource allocation (IoVRA), involves the allocation of four resources: the slice (deciding which packet to send), the coverage of the broadcast (deciding the range of the broadcast), the resource blocks (RBs), and the power. By carefully allocating these four resources, and by applying the successive interference cancellation (SIC) at the corresponding destination vehicles, NOMA can help in boosting the capacity of the IoV network. The use of NOMA in broadcast communications is different from the usual uplink and downlink NOMA techniques, which is due from the broadcast nature in IoV networks, i.e., two source vehicles broadcast with two distinct transmission powers to the same group of destination vehicles.

Zoubeir Mlika and Soumaya Cherkaoui are with the research laboratory on intelligent engineering for communications and networking (INTERLAB), Faculty of Engineering, Department of Electrical and Computer Science Engineering, University of Sherbrooke, Sherbrooke J1K 2R1, Quebec, Canada, (e-mail: zoubeir.mlika@usherbrooke.ca, soumaya.cherkaoui@usherbrooke.ca).
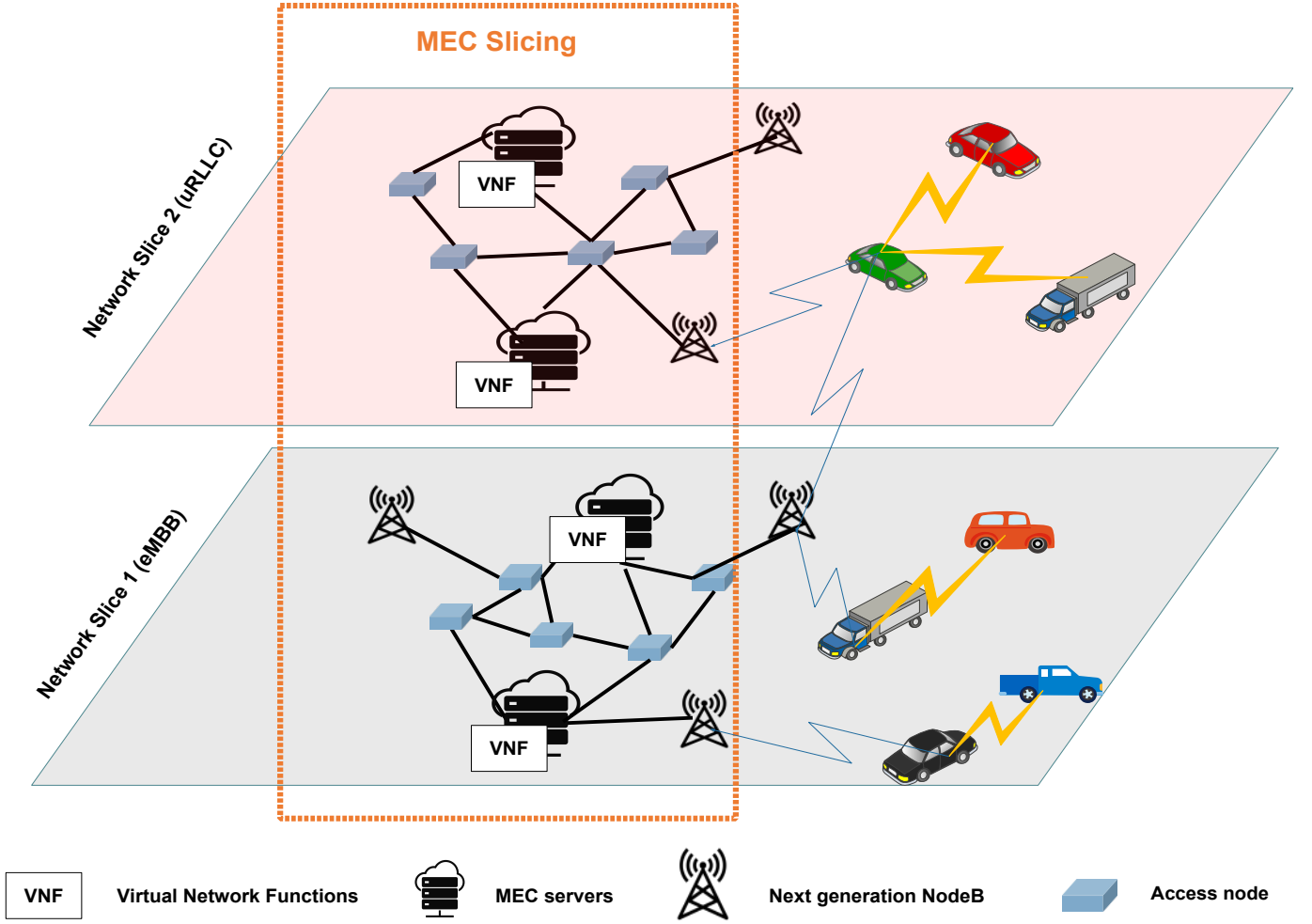
Fig. 1: Two network slices in an IoV-based MEC network.

Even though we propose a MEC-based IoV solution for the case of vehicle-to-vehicle (V2V) communications, our proposed system model is valid for vehicle-to-infrastructure (V2I) communications as well. Indeed, in V2I communications, a vehicle communicates with a gNB-type road side unit (RSU) or a user-type RSU through the cellular Uu or the sidelink (SL) connectivity [12]. For the case of user-type RSU communications, the coverage range selection decision will simply include the RSU. For the case of gNB-type RSU communications, the broadcast coverage range selection could be ignored and replaced by RSU association. Thus, our proposed solution framework is still valid for both V2V and V2I communications.

To the best of our knowledge, this is the first work that proposes a model-free DRL framework to solve IoVRA in MEC-enabled IoV networks based on broadcast, NS and NOMA. The contributions of our work are the following. We model IoVRA as a single agent MDP. Next, we propose a deep-Q-learning (DQL) algorithm to solve it. Finally, we show that our proposed DQL algorithm outperforms benchmark algorithms.

A. Organization

The article is organized as follows. Section II presents the system model, the single agent MDP, and describes the proposed DQL algorithm. Section III presents benchmark algorithmic solutions and gives the simulation results. Finally, section IV draws some conclusions and discusses interesting open research questions.

II. Proposed DRL for Internet of Vehicles

A. Internet of Vehicles Model

We consider an IoV network composed of a set of source vehicles that generate packets, and a set of destination vehicles that receive packets. All vehicles operate in the in-coverage scenario of 5G-NR [10] and thus they are covered by some gNB that performs edge computing. A source vehicle uses broadcast communications to transmit to a subset of the destination vehicles. The time is slotted into a set of slots. The total bandwidth is divided into a set of frequency slots. A resource block (RB) is given by the pair (frequency, slot).

The proposed system model supports several use cases, including advanced driving with trajectory sharing, extended sensors [13] and is valid for both V2V and V2I communications. To provide guaranteed quality of service requirements

to the different use cases, NS is used, which is an efficient solution in IoV networks [6]. It mainly creates logical networks on the top of a common and programmable MEC-enabled IoV infrastructure. We create two network slices. The first slice (slice 1) is designed for non-safety applications such as video streaming. The second slice (slice 2) is designed for safety applications such as emergency warnings. An example of the MEC-enabled NS system model is given in Fig. 1, where vehicles communicate with gNBs that are connected to MEC servers. On top of this network infrastructure, two network slices are created to support IoV applications. Slice 1 is designated for high throughput or enhanced mobile broadband communication (eMBB) and slice 2 is designated for ultra-reliable and low latency communication (uRLLC).

Each source vehicle has two different packets for each slice, where slice 1's packet (pkt^n) requires high throughput whereas slice 2's packet (pkt^s) has stringent latency requirements. For any packet to be delivered successfully, the corresponding source vehicle requires a set of RBs such that the achievable data rates are above the minimum requirements. Packet pkt^n can be transmitted using any RBs from the frequency-slot resource pool with a carefully chosen transmission power per each RB. However, pkt^s, having an arrival time and a deadline, can be transmitted using any frequency slot but only using slots between its arrival time and deadline with a carefully chosen transmission power per each RB. The wireless channel gain between two vehicles includes fast and slow fading.

A source vehicle has to decide which packet to send, at what range to broadcast, what RBs to use, and what transmission powers to allocate. The range broadcasting optimzation is smilar to the classical vehicle clustering [14–17]. To improve the spectral efficiency of the IoV network, we use NOMA to superimpose the transmissions of the source vehicles transmitting to some destination vehicle, which uses SIC to decode the superimposed transmissions.

B. Proposed Deep-Q-Learning Algorithm

Vehicles operate in the coverage of gNB with MEC, that collects information about vehicles and performs pilot estimation to obtain the channel statistics. Based on the obtained feedback information, gNB observes the IoV environment and makes decisions. It plays the role of an intelligent entity in a single agent MDP. With the help of DRL, gNB learns to solve efficiently the complicated IoVRA problem. Specifically, gNB implements the well-known DQL approach [18]. DQL has mainly two parts: *training* and *inference*. In training, gNB trains a deep-Q-network (DQN), whereas in inference, it takes actions according to its trained DQN. DQL is an improvement of the so-called QL algorithm that is based on a tabular method which creates a table of state-action pairs. QL explores the action space using an exploration policy, e.g., ϵ-greedy. Despite the proven effectiveness of QL, it generally fails when the state and action spaces become large as in IoVRA.

DQL is a promising technique that is proposed to solve the curse of dimensionality in RL by approximating the Q action-value function using deep learning. One way to solve IoVRA is through multi-agent DRL by combining independent QL for each agent. That is, each agent tries to learn its own policy based on its own observations and actions while treating all other agents as part of the environment. This badly influences the result of the training as it creates a non-stationary environment that changes as other agents take decisions. For this reason, a MEC-enabled IoV network facilitates the training in such situation by modeling IoVRA as a single agent who performs the training at the edge of the IoV network. The system architecture of the proposed DQN approach is given in Fig. 2, in which gNB and MEC server interact with the IoV environment and take decisions accordingly.

Before describing in detail DQL, first, IoVRA is modeled as a single agent MDP given by the quadruple: state space, action space, reward function and transition probability. The agent in this MDP is the gNB, which takes an action, receives a reward and moves to the next state based on its interaction with the unknown IoV environment. This interaction helps gNB gain more experiences and improves its accumulated reward.

1) The State Space: At any slot, any state of the IoV environment is unknown directly to gNB. Instead, gNB receives an observation from the IoV environment. In our model, an observation includes local channel state information (CSI) and the transmission behavior of the source vehicles. More precisely, an observation includes the large and small-scale fading values between vehicles. These values can be accurately estimated by the destination vehicles and fed back to gNB without significant delay [19]. The observation also includes a decision variable that indicates whether the source vehicles transmitted in previous slots and if so which packet did they transmit. The third observation indicates the number of leftover bits of packets that each source vehicle needs to send (e.g., initially, the number of leftover bits correspond to the packets sizes). The fourth observation element includes the arrival time and the deadline of slice 2 packets.

2) The Action Space: IoVRA is solved in an online fashion where at each slot, gNB makes a decision that includes (i) the broadcast coverage range selection (ii) the slice selection (iii) the RB allocation, and (iv) the power allocation. For (i), we define a discrete set of coverage distances (including zero). Thus, if gNB chooses a coverage distance (or 0), then it will broadcast (or does not) to all destination vehicles within the chosen coverage circle having as radius the indicated range. For (ii), we define a discrete set of packets (including the empty set) that indicates which packet gNB will decide to transmit. At each slot, each source vehicle has three possible choices: it does not transmit, it transmits a slice 1 packet, or it transmits a slice 2 packet. For (iii), the RB allocation is about choosing the frequency slot to be used in the current slot. For (iv), gNB carefully chooses the transmission power per RB. Note that continuous power allocation makes the implementation of DQL more complex and thus, to keep things simple, we use a discrete set of power levels that gNB can use. Finally, the action space of gNB is given by the Cartesian product of these four discrete sets.

3) The Reward Signal: We mainly focus on maximizing the packet reception ratio (PRR) [20] in IoV broadcast networks. PRR is defined in as follows: for one packet and one source vehicle, the PRR is given by the percentage of vehicles with
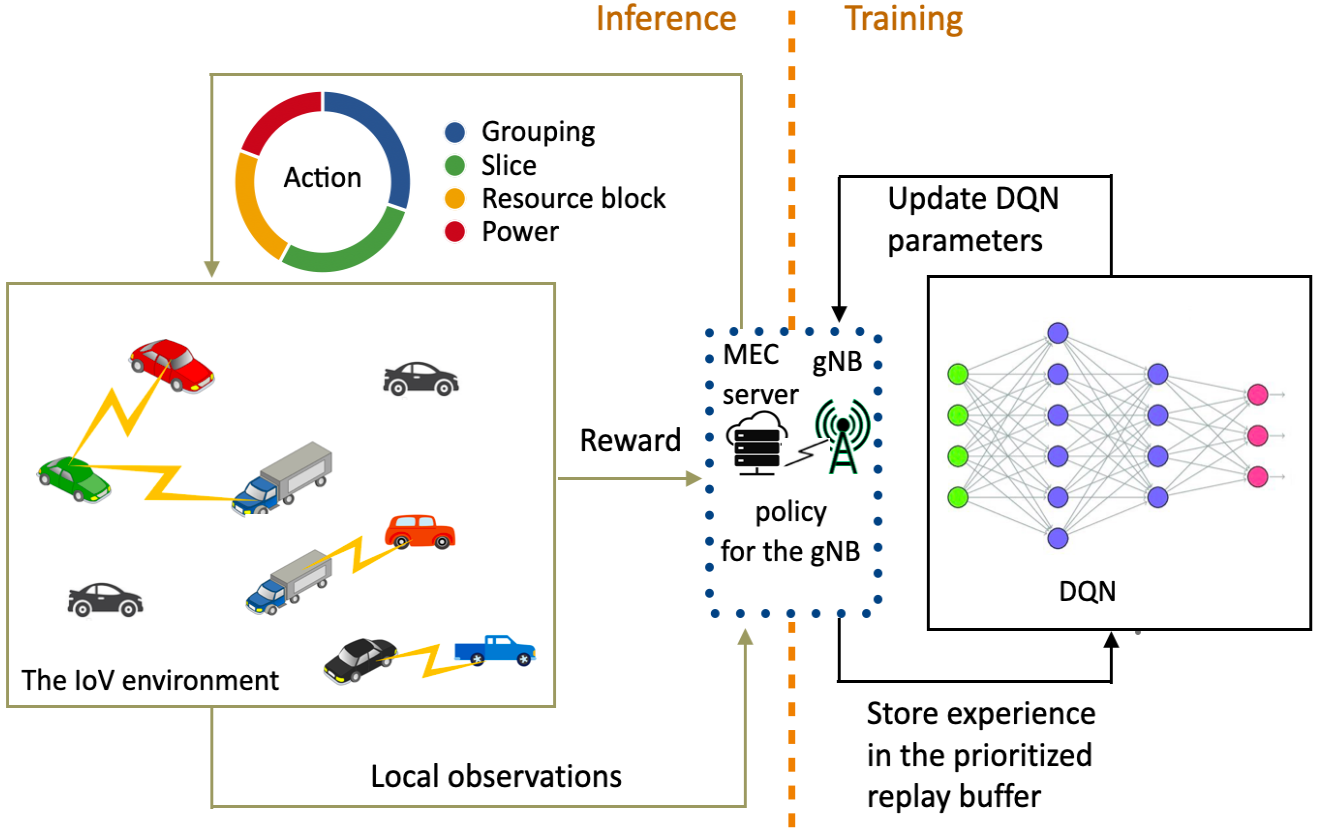
Fig. 2: IoV-based DRL architecture.

successful reception among the total number of receptions. PRR directly relates to the number of successfully received packets. Therefore, our main goal is to maximize the later.

The reward signal at any slot is the sum of individual rewards of each source vehicle. Hence, the reward signal depends on whether each source vehicle has successfully transmitted its packet or not. Technically, since we aim to maximize the number of successfully received packets, we set the reward to one once a packet is successfully delivered and zero otherwise. However, this leads to poor design since the zero individual reward leads to no useful information for learning. Thus, we build the individual reward design based on the following. When a packet is not successfully delivered or the delivery has not been completed yet, the individual reward is set to the *normalized* achievable rate between the corresponding vehicles. The normalization is used to upper-bound the reward. When the packet is successfully delivered, the individual reward is set to the chosen upper-bound. In the first case, upper-bounding the individual reward helps gNB acquire useful information for future decisions whereas in the second case, choosing the individual reward to be the upper-bound teaches gNB the best possible decisions to take in the future and helps in maximizing the number of successfully delivered packets. The achievable data rate is calculated based on the signal to interference-plus-noise ratio (sinr) according to uplink NOMA. The overall reward signal that gNB receives is thus the sum of individual rewards of each source vehicle. The goal of DQL is to maximize the cumulative reward over

the long-run, given some initial state of the IoV environment. This cumulative reward is the sum over many time steps of the weighted rewards where the weight is proportional to some constant called the discount factor. This discount factor makes future rewards more important for gNB agent as their corresponding weight becomes larger. In IoVRA problem, since the proposed MDP model consists of episodes of finite length, i.e., each episode lasts a finite number of slots, IoVRA belongs to the finite horizon set of problems [21]. Further, since we aim to maximize the number of successfully delivered packets, then the MEC-based gNB agent can simply choose the discount factor to be one or a number that is close to one in order to accumulate higher rewards and thus a higher number of successfully delivered packets.

4) The Probability Transition: The probability of moving to the next state while being in an old state and taking some action depends on the highly dynamic IoV environment and cannot be explicitly calculated. This transition happens due to the channel coefficients variation and vehicles mobility.

5) Training in DQL: The DQL algorithm is composed of two parts: *training* and *inference*. The training is composed of several episodes where each episode spans the number of slots. DQL uses DNNs to approximate the Q function. We leverage DQL with prioritized replay memory and dueling. In general experience replay memory helps to remember and use past experiences. Standard replay memory is used to sample experience transitions uniformly without paying attention to the significance of the sampled experiences. Prioritized expe-

rience replay memory is proposed to pay more attention to important experiences. This indeed makes the learning better. Also, dueling is proposed as a new neural network architecture that represents two estimators for the Q function.

In detail, the training lasts a number of episodes and requires as input the IoV environment which includes the vehicles, the channel coefficients, the packet requirements, the available RBs and any other relevant IoV network parameter. It returns as output the trained DQN. The first step in DQL is to start the simulator which generates the vehicles and all network parameters, then it initializes the DQN hyperparameters. In the beginning of the first slot, the initial state of the IoV environment (initial distances of the vehicles, etc.) is revealed to gNB. Next, DQL iterates the episodes. For each episode, the environment is built by (i) updating the network parameters, e.g., the leftover bits of each source vehicle are updated based on the previous episodes, and (ii) moving the vehicles according to the mobility model. Next, the exploration rate ϵ is annealed based on the episode index. Annealing the exploration rate over time is a technique used in RL to solve the dilemma between exploration and exploitation, i.e., as the time goes by, we decrease ϵ to increase the exploitation probability as the agent starts to learn something useful. After a few episodes, the value of ϵ is no longer decreased. Then, gNB chooses for each source vehicle an action that is a tuple of the coverage distance, the packet, the frequency slot, and the power level. Once gNB agent chooses its action according to the annealed ϵ, it calculates the reward signal. Specifically, a destination vehicle calculates the received sinr, finds the number of bits a source vehicle is transmitting, and communicates this information to gNB using feedback channels. The environment moves to the next state and gNB adds to its prioritized replay memory the actual experience with some associated priority, i.e., the obtained tuple (state, action, reward, next state) is associated some priority. Initially, gNB assigns random priorities to its experiences but the priorities change as it starts to learn and updates its DQN parameters. gNB samples a mini-batch from its prioritized replay memory according to their priorities that forms a dataset used to train the DQN. gNB uses a variant of the well-known stochastic gradient descent to minimize the loss and it updates the priorities of the sampled experiences proportionally to the value of the loss. Finally, once in a while, the trained DQN is copied into the target DQN.

6) Implementing DQL: The inference of DQL is as follows (see Fig. 2). First, the trained DQN is loaded. Also, the annealed ϵ is loaded from the last training episode (the index of the episode is also revealed). Then, for each episode (which represents a new random channel realization), the environment is reset and built—initializing the network parameters and the transmission behaviors of each agent. Next, for each slot, gNB agent, after observing the environment, chooses the best action according to its trained DQN after feedback communication between itself and the destination vehicles. Then, the reward signal is obtained, and the next episode starts with a new random channel realization.

The inference in DQL is done in an online fashion. That is, it is executed in each slot without knowing the future observations. The training in DQL is the most computationally intensive task. It is executed for a large number of episodes and can be done in an offline manner with different channel conditions and IoV network topologies. Note that training in DQL needs to be re-executed only when the topology of the IoV network undergoes significant changes, depending on the IoV network dynamics.

III. Performance Evaluation

In this section, we validate the proposed DQL method. The simulation setup is based on the highway scenario of [20] and most simulation parameters are taken from [22, 23]. We consider a six-lane highway with a total length of 2 km where each lane has a width of 4 m. There are three lanes for the forward direction (vehicles move from right to left) and three lanes for the backward direction. The source and destination vehicles are generated according to spatial Poisson process. Vehicles' speed determine the vehicle density and the average inter-vehicle distance (in the same lane) is $2.5\text{s} \times v$ where v is the vehicle absolute speed. The speed of a vehicle depends on its lane: the ith forward lane (from top to bottom with $i \in \{1, 2, 3\}$) is characterized by the speed of $60 + 2(i - 1) \times 10$ km/h, whereas the ith backward lane (from top to bottom with $i \in \{1, 2, 3\}$) is characterized by the speed of $100 - 2(i - 1) \times 10$ km/h. The number of source vehicles m and destination vehicles n is randomly chosen. The important simulation parameters are given as follows [22, 23]. The carrier frequency is 2 GHz, the per-RB bandwidth is 1 MHz, the vehicle antenna height is 1.5 m, the vehicle antenna gain is 3 dBi, the vehicle receiver noise figure is 9 dB, the shadowing distribution is log-normal, the fast fading is Rayleigh, the pathloss model is LOS in WINNER + B1, the shadowing standard deviation is 3 dB, and the noise power N_0 is −114 dBm.

Unless specified otherwise, the slice 1 packet's size is randomly chosen in $\{0.1..1\}$ Mb. The slice 2 packet's size is 600 bytes. gNB chooses a coverage (in m) from the set $\{100, 400, 1000, 1400\} \cup \{0\}$. The power levels (in dBm) are given by $\{15, 23, 30\} \cup \{-100\}$ where −100 dBm is used to indicate no transmission. We set $m = 3$, $n = 4$, $F = 2$, and $T = 20$; each slot has duration 5 ms. The DQN is trained in the Julia programming language using Flux.jl. The DQN consists of an input and an output layer and of three fully connected hidden layers containing respectively 256, 128, and 120 neurons. The ReLu activation function is used in each layer. The ADAM optimizer with a learning rate of 10^{-5} is used. The training lasts 3000 episodes with an exploration rate starting from 1 and annealed to reach 0.02 for the 80% of the episodes.

To the best of our knowledge, there are no current research works that solve IoVRA while considering the slice selection, the broadcast coverage selection, the RBs and the power allocation. We implement three benchmarks: two are based on NOMA and one is based on OMA. The partial idea of all benchmarks comes from [24] which is based on the swap matching algorithm. All benchmarks are centralized in the edge and offline. They are called OMA-MP, NOMA-MP, and NOMA-RP. In OMA-MP, every RB is used by at most one

vehicle and the maximum transmission power is allocated. In NOMA-MP and NOMA-RP, every RB can be shared, and the maximum transmission power or a random transmission power are allocated, respectively. The coverage and slice selections are decided randomly at the beginning of each slot. The allocation of the RBs to the vehicles is done similarly in all benchmarks. First, an initial RB allocation is executed that gives the highest sum of channel power gain between a source vehicle and its destination vehicle. Once the initial allocation is obtained, a swap matching is performed to improve the number of packets successfully received. If no swap improves the matching, then the algorithm terminates.

In the simulation results, we present two performance metrics: the cumulative rewards for training the DQL and the number of successfully received packets for the inferring DQL. In the training, the reward signal received by gNB is given by the sum of the individual rewards of each source vehicle. The individual reward is equal either to (i) the upper-bounded achievable rate or to (ii) the upper bound. The event (i) happens when a packet is not yet delivered whereas the event (ii) happens when a packet is completely and successfully delivered. In the inference, the reward signal is simply given as the total number of successfully delivered packets.
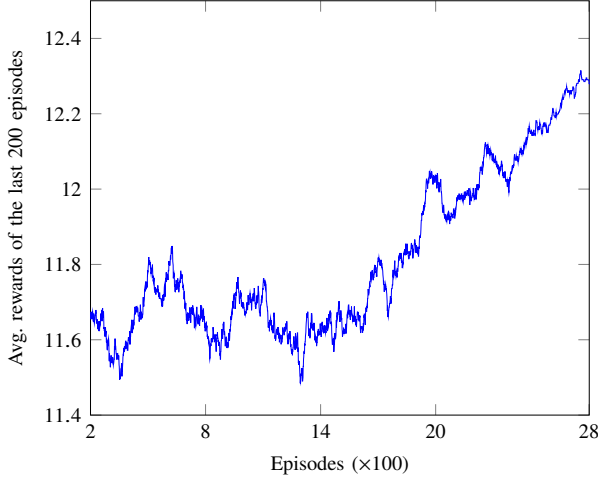

Fig. 3: Training rewards.

Fig. 3 illustrates the convergence of the proposed DQL algorithm versus training episodes. The figure shows the cumulative average rewards per episode where the average is taken over the last 200 episodes. It is clear that the average reward improves as the training episodes increase. This shows the effectiveness of the proposed algorithm. The training in DQL gradually converges starting from the episode number ≈ 2700. Note that the convergence of the algorithm is not smooth and contains some fluctuations which is due mainly to the high mobility nature of the IoV environment. Based on Fig. 3, DQN is trained for 3000 episodes to provide some convergence guarantees.

In the next two figures, we present, as a performance metric, the reward obtained in the inference part of DQL, which is the number of successfully received packets. We show this performance metric as stacked bars where each bar is divided into two parts: the lower part indicates the number

of successfully delivered slice 1 packets and the higher part indicates the number of successfully delivered slice 2 packets.
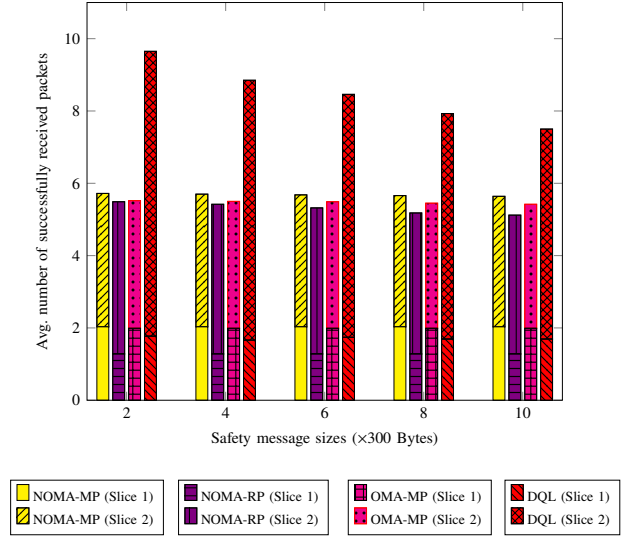

Fig. 4: Impact of safety message sizes

Fig. 4 shows the performance of DQL against the benchmarks when varying the slice 2 packet sizes. We can see that DQL succeeds in delivering more packets without having the full and future CSI as in the benchmarks. For example, DQL can, on average, deliver successfully almost 9 packets. However, other benchmarks can only deliver, on average, almost 6 packets. NOMA-RP achieves the lowest performance as expected. Further, DQL achieves a higher number of successfully delivered slice 2 packets. This is particularly important in IoV communication as slice 2 packets are mainly safety packets and thus must have a higher priority of being delivered.
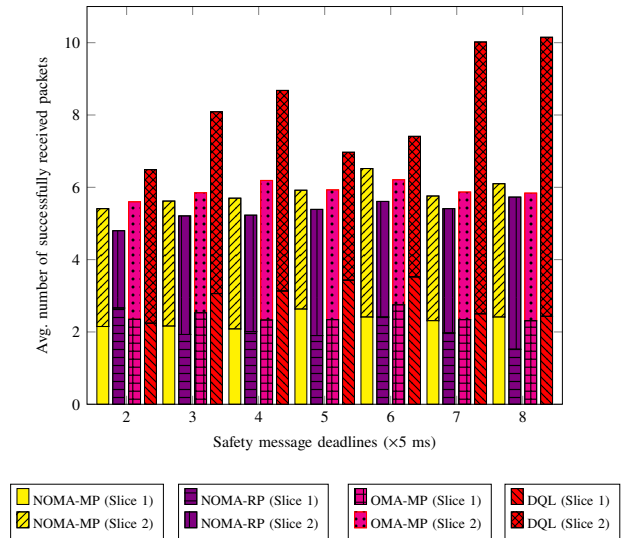

Fig. 5: Impact of safety message deadlines

Fig. 5 shows the performance of DQL against the benchmarks when varying the slice 2 packets deadlines. DQL

still achieves the best performance when the deadline of the safety packets increases. The gap between DQL and other benchmarks widens further as the deadline increases. We further notice that NOMA-RP has the worst performance for all algorithms which shows the need of a suitable power allocation method in IoVRA.

We notice from both Fig. 4 and Fig. 5 that there is an unfair allocation of resources between the packets of the two slices. This is mainly due to highly dynamic nature of the IoV network (e.g., vehicle positions, their speeds, etc.). For example, if a source vehicle is located close to a destination vehicle, then the quality of the wireless link between both vehicles will likely be good. Thus, gNB learns through DQL to equally likely transmit both packets. However, in the case where the source vehicle is located far away from the corresponding destination vehicle, the quality of the wireless link between both parties will probably be poor and thus, gNB will likely learn through DQL to transmit only slice 2 packets to guarantee a successful V2V communication (since slice 2 packets might not require a large number of RBs compared to slice 1 packets). It is thus important to study the fairness among different slices in such IoV network, which will be investigated in our future works.

IV. Conclusions and Future Works

In this paper, we developed an online MEC-based scheme to solve the slice selection, coverage selection, resource block and non-orthogonal multiple access power allocation problem in the Internet of vehicles network. We modelled the problem as a single agent Markov decision process and developed a DQL algorithm. The proposed DQL algorithm is proven robust and effective against various system parameters including the high mobility characteristics of IoV networks. It also outperformed some baseline benchmark algorithms that are based on global and offline decisions. In future works, we will investigate a two-time scale DRL approach that decides for coverage and slice selection on a slower time scale. Further, we will study the fairness of multiple slices. Finally, we will extend our system model to include mmWave communications.

V. Acknowledgment

References

[1] A. Triwinarko, I. Dayoub, and S. Cherkaoui, "Phy layer enhancements for next generation v2x communication," *Vehicular Communications*, vol. 32, p. 100385, 2021.

[2] A. Alalewi, I. Dayoub, and S. Cherkaoui, "On 5g-v2x use cases and enabling technologies: a comprehensive survey," *IEEE Access*, 2021.

[3] R. Soua, I. Turcanu, F. Adamsky, D. Führer, and T. Engel, "Multi-Access Edge Computing for Vehicular Networks: A Position Paper," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.

[4] Z. Mlika and S. Cherkaoui, "Network slicing for vehicular communications: a multi-agent deep reinforcement learning approach," *Annals of Telecommunications*, vol. 76, no. 9, pp. 665–683, 2021.

[5] C. Campolo, A. Molinaro, A. Iera, R. R. Fontes, and C. E. Rothenberg, "Towards 5G Network Slicing for the V2X Ecosystem," in *Proc. IEEE Conf. on Netw. Softwarization and Workshops (NetSoft)*, 2018, pp. 400–405.

[6] H. Khan, P. Luoto, S. Samarakoon, M. Bennis, and M. Latva-Aho, "Network Slicing for Vehicular Communication," *Transactions on Emerging Telecommunications Technologies*, p. e3652, e3652 ett.3652. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3652

[7] M. Azizian, S. Cherkaoui, and A. S. Hafid, "Vehicle software updates distribution with sdn and cloud computing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 74–79, 2017.

[8] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-Learning-Based Wireless Resource Allocation With Application to Vehicular Networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, 2020.

[9] A. Abouaomar, S. Cherkaoui, Z. Mlika, and A. Kobbane, "Service function chaining in mec: A mean-field game and reinforcement learning approach," *arXiv preprint arXiv:2105.04701*, 2021.

[10] 3GPP, "Study on NR Vehicle-to-Everything (V2X)," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.885, 03 2019, version 16.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3497

[11] B. Di, L. Song, Y. Li, and Z. Han, "V2X Meets NOMA: Non-Orthogonal Multiple Access for 5G-Enabled Vehicular Networks," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 14–21, 2017.

[12] 5GCAR, "Final Design and Evaluation of the 5G V2X System Level Architecture and Security Framework," The 5G Infrastructure Public Private Partnership (5GPPP), Deliverable D4.2, 11 2019, version 1.1. [Online]. Available: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5c9d36fbc&appId=PPGMS

[13] S. A. Ashraf, R. Blasco, H. Do, G. Fodor, C. Zhang, and W. Sun, "Supporting Vehicle-to-Everything Services by 5G New Radio Release-16 Systems," *IEEE Commun. Standards Mag.*, vol. 4, no. 1, pp. 26–32, 2020.

[14] M. Azizian, S. Cherkaoui, and A. S. Hafid, "A distributed d-hop cluster formation for vanet," in *2016 IEEE wireless communications and networking conference*. IEEE, 2016, pp. 1–6.

[15] ——, "A distributed cluster based transmission scheduling in vanet," in *2016 IEEE international conference on communications (ICC)*. IEEE, 2016, pp. 1–6.

[16] ——, "Dcev: A distributed cluster formation for vanet based on end-to-end realtive mobility," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2016, pp. 287–291.

[17] M. Azizian, S. Cherkaoui, and A. Hafid, "An optimized flow allocation in vehicular cloud," *IEEE Access*, vol. 4, pp. 6766–6779, 2016.

[18] V. Mnih *et al.*, "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 02 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[19] Y. S. Nasir and D. Guo, "Multi-Agent Deep Reinforcement Learning for Dynamic Power Allocation in Wireless Networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, 2019.

[20] 3GPP, "Study on Evaluation Methodology of New Vehicle-to-Everything (V2X) Use Cases for LTE and NR," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 37.885, 06 2019, version 15.3.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3209

[21] M. J. Kochenderfer, C. Amato, G. Chowdhary, J. P. How, H. J. D. Reynolds, J. R. Thornton, P. A. Torres-Carrasquillo, N. K. Üre, and J. Vian, *Decision Making Under Uncertainty: Theory and Application*, 1st ed. The MIT Press, 2015.

[22] L. Liang, H. Ye, and G. Y. Li, "Spectrum Sharing in Vehicular Networks Based on Multi-Agent Reinforcement Learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, 2019.

[23] L. Wang, H. Ye, L. Liang, and G. Y. Li, "Learn to Compress CSI and Allocate Resources in Vehicular Networks," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3640–3653, 2020.

[24] M. Zeng, A. Yadav, O. A. Dobre, and H. V. Poor, "Energy-Efficient Joint User-RB Association and Power Allocation for Uplink Hybrid NOMA-OMA," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5119–5131, Jun. 2019.