Semi-Decentralized Network Slicing for Reliable V2V Service Provisioning: A Model-free Deep Reinforcement Learning Approach

Jie Mei, *Member, IEEE,* Xianbin Wang*, *Fellow, IEEE*, Kan Zheng, *Senior Member, IEEE*

Abstract—Applying of network slicing in vehicular networks becomes a promising paradigm to support emerging Vehicle-to-Vehicle (V2V) applications with diverse quality of service (QoS) requirements. However, achieving effective network slicing in dynamic vehicular communications still faces many challenges, particularly time-varying traffic of Vehicle-to-Vehicle (V2V) services and the fast-changing network topology. By leveraging the widely deployed LTE infrastructures, we propose a semi-decentralized network slicing framework in this paper based on the C-V2X Mode-4 standard to provide customized network slices for diverse V2V services. With only the long-term and partial information of vehicular networks, eNodeB (eNB) can infer the underlying network situation and then intelligently adjust the configuration for each slice to ensure the long-term QoS performance. Under the coordination of eNB, each vehicle can autonomously select radio resources for its V2V transmission in a decentralized manner. Specifically, the slicing control at the eNB is realized by a model-free deep reinforcement learning (DRL) algorithm, which is a convergence of Long Short Term Memory (LSTM) and actor-critic DRL. Compared to the existing DRL algorithms, the proposed DRL neither requires any prior knowledge nor assumes any statistical model of vehicular networks. Furthermore, simulation results show the effectiveness of our proposed intelligent network slicing scheme.

Index Terms—V2V communication, C-V2X Mode-4, network slicing, deep reinforcement learning.

I. INTRODUCTION

In recent years, vehicle-to-vehicle (V2V) communication has become one critical enabler for the rapidly growing connected vehicle and intelligent transportation industries. The global connected vehicle market is expected to grow from $ 42.25 billion in 2018 to $ 142.49 billion by 2026, expanding at a Compound Annual Growth Rate (CAGR) of 16.4 % [1]. Meanwhile, emerging V2V applications, such as cooperative collision avoidance, autonomous driving, and platooning control, have led to a broad spectrum of Quality of Service (QoS) requirements on vehicular networks [2], [3]. However, conventional vehicular networks supporting human-centric applications cannot fully meet the highly diverse QoS

requirements of future V2V applications. This limitation requires the new generation of vehicular networks to enable diverse QoS provisioning by intelligent and efficient utilization of limited radio resources [4].

One promising solution for diverse QoS provisioning in vehicular networks is network slicing, which provides a multipurpose platform to enable a wide range of applications and services. Network slicing creates multiple virtual customized networks, referred to as network slices, on top of a common substrate infrastructure. Therefore, the main goal of this paper is to implement the network slicing paradigm in vehicular networks, where the operator can flexibly compose network slices for meeting specific QsS demands of various V2V applications.

However, the detailed design of the network slicing scheme for vehicular networks is still very challenging due to the following two open issues,

- How to integrate the network slicing paradigm with state-of-art vehicular communication techniques in a cost-effective and scalable manner?
- How to realize proactive and situation-aware network slicing that can ensure diverse QoS requirements of V2V services in time-varying vehicular networks?

This paper addresses these two issues by combining the Cellular Vehicle-to-Everything (C-V2X) system with the concept of Artificial Intelligence (AI) empowered network slicing.

Firstly, the C-V2X standard has been proposed to replace the existing IEEE 802.11p protocol, which cannot fully support today's V2V services. Currently, the C-V2X standard includes two modes of operation, i.e., C-V2X Mode-3 and C-V2X Mode-4. In Mode-3, eNodeB (eNB) directly allocates radio resources to vehicles for their V2V transmissions in a centralized way. In Mode 4, vehicles perform distributed radio resource scheduling to autonomously select radio resources from a radio resource pool without a centralized scheduler [5]. However, compared with Mode 4, Mode 3 could cause unbearable control signaling overhead and processing delay in the dense and dynamic V2V scenarios. Since the poor scalability of Mode 3, exploration of Mode 4 is a potential direction to enable diverse QoS provisioning in realistic transportation environments.

Secondly, although network slicing can customize network slices according to specific QoS demands, this performance gain comes at the cost of introducing much more complexity into the communication system. This high complexity makes traditional mathematical model-based approaches to awareness

of network situation and network operation no longer adequate since the model-based approaches either lack explicit models or do not have the processing time to calculate heuristic solutions [6], [7]. This challenge motivates us to propose an AI-empowered network slicing architecture for vehicular networks to support vehicular applications [8]. It shows great potential in developing intelligent network slicing schemes for supporting V2V applications with diverse QoS requirements.

With the observed considerations, this paper proposes a semi-decentralized network slicing framework based on C-V2X Mode 4 to maximize the long-term QoS performances of V2V services. In principle, the operation of network slices is under the supervision of eNB. Based on deep reinforcement learning (DRL), eNB extracts the underlying network situations and adjusts slice configuration accordingly. Under the coordination of eNB, vehicles of each slice are automatically performing radio resource scheduling procedures. The following briefly summarizes the main technical contributions of this work:

- Design of a semi-decentralized network slicing framework based on C-V2X Mode 4. It has two layers. First, eNB executes adaption of slice configuration, i.e., inter-slice radio resource allocation and tuning of parameters of Mode 4 protocol, according to the network dynamics at a large timescale. Then, conditioned by the slice configuration determined by the eNB, the vehicle in each slice performs autonomous radio resource selection for V2V transmission based on Mode 4. In this framework, the eNB performs slicing control with coarse resource and time granularity and does not require frequent inter-action between eNB and vehicles, significantly reducing the signaling overhead and allowing sufficient time for intelligent processing.
- Model-free DRL to realize the situation-aware slicing control with only partial information of vehicular networks. The adaption of slice configuration at eNB is functioned by a DRL agent. However, the eNB only has partial observation information of vehicular networks. This makes conventional DRL methods inefficient, which are relying on the prior knowledge of systems. There-fore, based on the partially observed Markov decision process (PoMDP), we propose an actor-critic structured DRL algorithm by exploring the long short-term memory (LSTM). Specifically, LSTM enables the eNB to extract the underlying network situation from historical, partial information of the vehicular network. With the proposed DRL algorithm, eNB can perform slicing control with self-configuration and self-optimization capabilities.

The remainder of this paper is structured as follows. Section II presents related works. Section III describe the considered system model of network slicing in Mode 4 based vehicular networks. In Section IV, we propose a semi-decentralized net-work slicing framework based on C-V2X Mode 4. In Section III, we formulate the optimization of slicing configuration policy at the eNB as a PoMDP problem. In Section IV, we propose an actor-critic structured DRL algorithm to solve the formulated problem. In Section V, we present numerical experiments to compare the performance of the proposed algorithm against state-of-the-art baseline schemes. Finally, Section VI draws the conclusions.

II. RELATED WORKS

Since the C-V2X standard is relatively new, which is introduced in 3GPP Releases 14 and 15 and will be further enhanced in Release 16 [9], [10]. Current works mainly focus on three research topics: performance analysis of C-V2X network, radio resource management, and network slicing in C-V2X networks.

As mentioned above, C-V2X Mode-4 employs the distributed radio scheduling scheme, referred to as sensing based Semi-Persistent Scheduling (SPS) scheme, to enable autonomous radio resource management of each vehicle. In the sense-based SPS scheme, vehicles sense and keep a history of the channel status and utilize it to select suitable radio resources for V2V transmissions. Since the autonomy nature of C-V2X Mode 4, it faces radio resource sharing conflicts, i.e., packet collisions when two or more vehicles simultaneously utilize the same radio resources. This issue will affect the performance of vehicular networks. Thus, based on probabilistic theory, performance analytical models of C-V2X Mode 4 network are proposed for quantifying the collision probability, and throughput as a function of vehicle density and the distance between transmitting and receiving vehicle [11], [12]. Besides, based on network-level simulations, the authors in [13]–[15] analyze the impact of the main parameters of Mode 4 on the network performance, which shows that Mode 4 is robust and scalable for highly dynamic vehicular scenarios.

To further improve the performance of C-V2X networks, different studies have proposed options to enhance the radio resource management schemes. The authors in [16] propose a distributed radio resource management scheme for C-V2X Mode 4, which exploits geography information of vehicles to improve V2V communication reliability. Likewise, in [17], a spatial reuse-based radio resource management scheme is proposed to improve the spectrum utilization of vehicular networks. Instead of simply improving spectrum utilization, in the context of C-V2X assisted autonomous driving, the authors in [18] jointly optimize radio resource allocation, cooperative driving perception, and vehicle controls to improve driving safety and transportation efficiency.

However, the effectiveness of C-V2X Mode 4 still needs to be improved due to the following two limitations:
- Low operation efficiency of C-V2X Mode 4 network. In this decentralized network, each vehicle performs the distributed radio resource scheduling independently based on its local knowledge and needs, leading to selfish deci-sions from different vehicles. Thus, to avoid unreasonable decisions from each vehicle, it is critical to establish coordination between eNB and vehicles.
- Extreme difficult for network situational awareness. Firstly, due to the high mobility of vehicles, vehicular network status could change rapidly. Meanwhile, real-time sensing of network situations will consume exces-sive overheads to exchange sensing information between

eNB and vehicles. These make the real-time and precise awareness of network situations challenging at the eNB.

Recently, network slicing has been introduced into vehicular networks to meet diverse QoS requirements for V2X services. By leveraging Lyapunov optimization, [19] proposes a RAN slicing scheduling strategy for the joint radio resource allocation and power control, aiming to maximize long-term network capacity while guaranteeing the strict QoS requirements of V2V services. In [20], a hierarchical RAN slicing framework is developed for the heterogeneous vehicular networks, where other slices opportunistically reuse the idle radio resources of one network slice to improve the spectrum efficiency.

Based on our literature review and analysis, the above works on the C-V2X based vehicular networks have the following two limitations,

- Most of the literature assumes that network infrastructures can fully observe the status of the vehicular network. However, this assumption is too optimistic for the real scenarios of vehicular networks due to the high mobility of vehicles [21]. This characteristic hinders the direct sensing of network situations since it needs frequent interaction between eNB and vehicles, which will consume a large amount of signaling overheads. Thus, one effective solution is to enable the network slicing with only the long-term and partial information of vehicular networks.
- Most works are regulated by the conventional mathematical model-based approaches. The fundamental premise of these model-based approaches is to obtain a precise mathematical model to describe the system. Then, based on the accurate system model, we can further analyze or optimize the system performance. However, the dynamics and variation pattern of vehicular networks are difficult to be modeled accurately. Thus, it is reasonable for us to enable the operation of C-V2X based networks through a model-free AI technology, such as model-free DRL technologies.

Notations: In the following, italic boldface lower-case and upper-case characters denote vectors and matrices, respectively. Sets are denoted by calligraphic letters, i.e., $\mathcal{U}$. The operator $|\mathcal{U}|$ represents the cardinality of set $\mathcal{U}$. To ease readability, we list the major notations in Table I.

III. SYSTEM MODEL

Consider a freeway scenario with one eNB, where total available bandwidth is B. The time dimension is partitioned into slots of duration δ, indexed by $t \in \{1, 2, ...\}$. Assume the physical vehicular network is split into N network slices, denoted by $\mathcal{N} = \{1, 2, ..., N\}$, each of which has a specific V2V application it provides. Vehicular user equipment (VUE), counted in terms of transmitter, associated with slice $n \in \mathcal{N}$ are denoted as $\mathcal{V}_n$. Assumed that all VUEs can successfully receive the information of slice configuration from the eNB.

A. Traffic Model of V2V Services

In slice $n \in \mathcal{N}$, each VUE needs to periodically transmit packet to its receiving vehicle with a period of T_n slots.

TABLE I
MAJOR NOTATIONS USED IN THIS PAPER

Notation	Definition
$\mathcal{N}$	Set of network slices: $\{1, ..., N\}$
$\mathcal{V}_n$	Set of VUEs belong to network slice $n \in \mathcal{N}$
B	Total available bandwidth
δ	Duration of each time slot
F_n	Number of subchannel in slice $n \in \mathcal{N}$
B_n	Bandwidth of each subchannel in slice $n \in \mathcal{N}$
T_n^{sw}	Length of the selection window in slice $n \in \mathcal{N}$
$s_{i,m,t}$	$s_{i,m,t} = 1$ if VUE i selects the m-th sub-channel at slot t
$r_{i,t}$	Data rate of VUE i at slot t
Δt	Each epoch is composed by Δt consecutive slots
$\bar{d}_{n,k}$	Average packet delay of VUEs in slice n
$\bar{\beta}_{n,k}$	Average Packet Drop Ratio (PDR) of slice n
$\bar{x}_{n,k}$	Average subchannel occupancy ratio in slice n
$\boldsymbol{O}_{n,k}$	Observation of slice n at epoch k: $\{|\mathcal{V}_{n,k}|, \bar{d}_{n,k}, \bar{\beta}_{n,k}, \bar{x}_{n,k}\}$
$\boldsymbol{O}_k$	Observation of network at epoch k: $\{\boldsymbol{O}_{n,k} | n \in \mathcal{N}\}$
$\boldsymbol{H}_k$	Observation history of network at epoch k: $(\boldsymbol{O}_1, ..., \boldsymbol{O}_{k-1})$
$\boldsymbol{C}_k$	Configuration of slice n at epoch k: $\{F_n, B_n, T_n^{\text{sw}}, \forall n \in \mathcal{N}\}$
$\pi_{\boldsymbol{\theta}}$	Slice Configuration Policy: Mapping from $\boldsymbol{H}_k$ to $\boldsymbol{C}_k$
$J(\boldsymbol{O}_k)$	QoS-related reward function of all slices

Assume the packet in slice $n \in \mathcal{N}$ have a fixed data size Z_n (in bits). It is assumed that each VUE has a queue buffer to store the packet to be delivered, and the packet is delivered based on the first-come-first-serve (FCFS) criteria. Let $l(= 1, 2, ...)$ denotes the index of the packet arriving at VUE i's buffer in slice n. Furthermore, the inter-packet arrival time instant of l-th packet of VUE i is denoted as $t_{i,l}^{\text{a}} = l \cdot T_n$.

B. V2V communication based on C-V2X Mode 4

The total bandwidth B are sliced and assigned to each slice by the eNB. In slice $n \in \mathcal{N}$, the assigned bandwidth are re-organized as F_n sub-channels, indexed by $m \in \{1, 2, ..., F_n\}$, and the bandwidth of each subchannel is B_n ($F_n B_n < B$).

In Mode 4, VUE autonomously selects and reserves sub-channel for V2V transmission with Sensing based Semi-Persistent Scheduling (SPS) scheme [5]. As shown in Figure 1, at slot $t_{i,l}^{\text{a}}$, supposing that VUE $i \in \mathcal{V}_n$ needs to select a sub-channel to transmit l-th packet ($l = 1, 2, ...$), then this procedure can be divided into the following three steps, **Step 1 (Sensing)**: VUE i continuously senses the signal strength in each subchannel during the last T_{sense} slots before slot $t_{i,l}^{\text{a}}$ (referred to as sensing window), and calculate the average signal strength of subchannel.

Step 2 (Subchannel Selection and Transmission): VUE i can select a subchannel within a Selection Window (SW). SW is a time window that includes the slots in the range $[t_{i,l}^{\text{a}}, t_{i,l}^{\text{a}} + T_n^{\text{sw}}]$, where T_n^{sw} is the length of selection window. VUE i sorts the candidate sub-channels in terms of the average received signal strength during the last sensing window, and then reserves the sub-channel with the lowest average received signal strength, which can be stated as:

- VUE i ranks all sub-channels in the selection window by their average signal strength in a descending order and selects the bottom 20% of them to compose the list of candidate sub-channels, denoted as $\mathcal{S}$;
- VUE i will randomly choose one of the candidate sub-channel in list $\mathcal{S}$. Assume VUE i chooses m-th ($1 \le m \le$

Fig. 1. An illustration of Sensing based Semi-persistent scheduling (SPS) process of VUE $i \in \mathcal{V}_n$ in network slice n: At slot $t_{i,l}^{\mathrm{a}}$, VUE i needs to select a sub-channel to transmit its packet. VUE i can select a subchannel within a Selection Window (SW). SW is a time window that ranging from slot $t_{i,l}^{\mathrm{a}}$ to $t_{i,l}^{\mathrm{a}} + T_n^{\mathrm{sw}}$, where T_n^{sw} is the length of SW. Then, VUE i sorts the idle sub-channels in terms of the average signal strength, which are sensed during the last sensing window that ranging from slot $t_{i,l}^{\mathrm{a}} - T_{\mathrm{sense}}$ to slot $t_{i,l}^{\mathrm{a}}$. Finally, VUE i selects a sub-channel with the lowest average signal strength.

F_n) sub-channel at t-th slot for transmitting l-th packet. Let $s_{i,m,t}$ denote sub-channel selection indicator for VUE i, where $s_{i,m,t} = 1$ means m-th sub-channel at the t-th slot is chosen by VUE i; otherwise, $s_{i,m,t} = 0$. However, m-th sub-channel at t-th slot may be used by other VUEs in slice n. For instance, supposing that VUE i and $i' \in \mathcal{V}_n$ simultaneously selects m-th sub-channel at t-th slot, i.e., $s_{i,m,t} = s_{i',m,t} = 1$. This condition will induce the intra-slice interference, which deteriorates the reliability of V2V communication. Then, Signal-to-Interference-plus-Noise Ratio (SINR) of the receiving vehicle of VUE i in m-th subchannel at t-th slot is given by

$$\gamma_{i,m,t} = \frac{P|g_{i,m,t}|^2}{\sum_{i' \in \mathcal{V}_n \setminus \{i\}} s_{i',m,t} P|g_{i',m,t}|^2 + N_0}, \text{ if } s_{i,m,t} = 1, \quad (1)$$

where $g_{i,m,t}$ is the channel gain, which contains path loss, shadowing effect and small-scale fading, from VUE i to its receiving vehicle in m-th subchannel, $g_{i',m,t}$ is the interference channel gain from VUE i' to the receiving vehicle of VUE i, P is the transmitted power of VUE, and N_0 is the power of additive white Gaussian noise (AWGN) in each sub-channel.

Therefore, the achievable data rate of receiver of VUE i at the t-th slot can be approximated by Shannon theory,

$$r_{i,t} = \sum_{m=1}^{F_n} s_{i,m,t} \cdot [B_n \delta \cdot \log(1 + \gamma_{i,m,t})], \; i \in \mathcal{V}_n, \quad (2)$$

where B_n is the bandwidth of each subchannel in slice n and δ is the time duration of slot.

The selected sub-channel is used to transmit a full packet. Then, delay of l-th packet at VUE i's buffer can be expressed as,

$$d_{i,l} = t - t_{i,l}^{\mathrm{a}}, \; i \in \mathcal{V}_n. \quad (3)$$

Specifically, packet latency $d_{i,l}$ approximately follows discrete uniform distribution $\mathrm{unif}\{1, T_n^{\mathrm{sw}}\}$. Thus, in slice $n \in \mathcal{N}$, the latency of VUE is impacted by T_n^{sw} (selection window length).

In our system model, packet is lost when Z_n error-free bits (i.e., packet size) cannot be correctly decoded by the receiving vehicle of VUE i. Thus, let binary variable $L_{i,l}$ denotes the packet loss indicator at VUE i's buffer, which can be represented as

$$L_{i,l} = \begin{cases} 1, & \text{if } r_{i,t} < Z_n, \\ 0, & \text{otherwise} \end{cases}$$

Furthermore, we define subchannel occupancy ratio, $x_{n,t}$, to characterize the level of subchannel congestion in slice n,

$$x_{n,t} = \sum_{m=1}^{F_n} \mathbb{1}\left\{\sum_{i \in \mathcal{V}_n} s_{i,m,t} \geq 1\right\} \Big/ F_n, \quad (4)$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. **Step 3 (Reservation and Re-selection)**: Once a sub-channel is reserved, the same sub-channel will be used for several consecutive V2V transmissions. After a random number of V2V transmissions VUE i will reselect its reserved sub-channel with probability p_{res}, and repeat **Step 1** and **2**.

Remark 1: Specifically, all VUEs associated with network slice $n \in \mathcal{N}$ have the same parameters (i.e., F_n, B_n and T_n^{sw}) of C-V2X Mode 4. The V2V communication performance is determined by these parameters. In our proposed network slicing framework (Section III-A), the eNB will determine how to adjust the parameters of C-V2X mode 4 for each slice according to the network situations.

IV. SEMI-DECENTRALIZED NETWORK SLICING AND PROBLEM FORMULATION

A. Proposed Network Slicing Framework

In this study, as an extension of our original work [8], a semi-decentralized network slicing framework for vehicular networks is proposed. It consists of an upper-level and a lower-level. As shown in Figure 2, at the upper-level, eNB is responsible for adjusting the slice configuration according to the dynamics of V2V service traffic at a large timescale. Specifically, the time resolution of upper-level is defined as an epoch, indexed by $k \in \{1, 2, ...\}$, the k-th epoch is ranging from slot $k \cdot \Delta t$ to slot $(k+1) \cdot \Delta t - 1$. At the lower-level, vehicles in each slice autonomously select their sub-channel using the sensing-based SPS scheme, which is configured by the upper-level. It is noteworthy that the upper-level is not involved in the real-time radio resource scheduling for V2V communications.

The upper-level is in charge of tuning the slice configuration to improve the QoS performance of services according to the partial observation history of vehicular networks. First, the

Fig. 2. The flowchart of the proposed semi-decentralized network slicing scheme for the C-V2X Mode 4 based vehicular networks.

specific definition of partial observation of vehicular networks is given as follow:

Definition 1 (Partial Observation of Vehicular Networks): The observation of vehicular network at k-th epoch is defined as

$$\boldsymbol{O}_k = \{\boldsymbol{O}_{n,k} \,|\, n \in \mathcal{N}\} \in \mathcal{O},$$

where the observation of slice $n \in \mathcal{N}$ at k-th epoch is

$$\boldsymbol{O}_{n,k} = \{|\mathcal{V}_{n,k}|, \bar{x}_{n,k}\},$$

a). *Number of VUEs in slice n, $|\mathcal{V}_{n,k}|$:* It is defined as the number of vehicles associated with slice n in k-th epoch, i.e. $|\mathcal{V}_{n,k}|$, where $\mathcal{V}_{n,k}$ is the VUE set in slice n during k-th epoch. Because epoch is in the level of hundreds of milliseconds, which is smaller than the vehicle inter-arrival time. Therefore, we assume that set $\mathcal{V}_{n,k}$ does not change within k-th epoch.

b). *Average subchannel occupancy ratio in slice n, $\bar{x}_{n,k}$:* It is defined as

$$\bar{x}_{n,k} = \frac{1}{\Delta t} \sum_{t=k\Delta t}^{(k+1)\Delta t - 1} x_{n,t}.$$

Besides, $\mathcal{O}$ is the set of all possible observations of vehicular networks.

Remark 2: Within each epoch, the eNB gets these two kinds of information through sensing subchannels at each slot, which barely needs information exchange between eNB and vehicles in its service area. At the end of each epoch, eNB aggregates the temporal dynamics of these raw information within epoch and converted into the observation of vehicular network $\boldsymbol{O}_k$. However, due to the high complexity and time varying nature of vehicular networks as well as the limited sensing ability, the observation of vehicular networks $\boldsymbol{O}_k$ is an *partial information* of network situation, which cannot be regarded as a full status of vehicular network.

Therefore, the eNB (i.e., the upper-level controller) can exploit the observation history of vehicular networks and infer the full information of network situation. Specifically,

Definition 2 (Observation History of Vehicular Networks): At the beginning of k-th epoch, the eNB obtains the partial observation of vehicular networks during the previous epoch (i.e., $\boldsymbol{O}_{k-1}$) and add it to the observation history. Herein, the observation history of vehicular networks is defined as

$$\boldsymbol{H}_k = (\boldsymbol{O}_1, ..., \boldsymbol{O}_{k-1}).$$

With the observation history, as shown in Figure 2, the upper level controlling policy are described as follows,

Definition 3 (Network Slicing Configuration Policy): The slicing configuration policy π is defined as a stochastic policy,

$$\pi_\theta (\boldsymbol{C}_k \,|\, \boldsymbol{H}_k) : \boldsymbol{H}_k \to \Pr[\boldsymbol{C}_k \,|\, \boldsymbol{H}_k], \ \boldsymbol{C}_k \in \mathcal{C},$$

which is a mapping from observation history of vehicular networks $\boldsymbol{H}_k$ to a probability distribution over the candidate slice configuration $\mathcal{C}$, which is the collection of all candidate slice configurations. Furthermore, $\boldsymbol{C}_k$ is the slice configuration at k-th epoch,

$$\boldsymbol{C}_k = \left\{ F_n, B_n, T_n^{\mathrm{sw}} \left| \sum_{n \in \mathcal{N}} \mathcal{F}_n \cdot B_n \leq B, n \in \mathcal{N} \right. \right\},$$

a). F_n is the number of subchannels in slice n;
b). B_n is the number of subchannels in slice n;
c). T_n^{sw} is the length of selection window of slice n.

Remark 3: In this paper, the network slicing configuration policy is represented by an Artificial Neural Network (ANN) π_θ, where $\boldsymbol{\theta}$ is the weight vector associated with this ANN. The output of π_θ is a probability distribution over candidate slice configurations. For instance, $\pi_\theta (\boldsymbol{C}_k \,|\, \boldsymbol{H}_k)$ is the probability of selecting slice configuration $\boldsymbol{C}_k \in \mathcal{C}$ under the condition of observation history $\boldsymbol{H}_k$.

B. Problem Formulation

In the following, we formulate the optimization of the RAN slicing policy in the proposed scheme as a stochastic optimization problem, whose goal is to maximize the long-term QoS performance of V2V network slices. Specifically, the average packet delay of UE, should be considered as one important metric of QoS for each slice, which is defined as,

$$\bar{d}_{n,k} = \frac{1}{|\mathcal{V}_{n,k}|} \sum_{i \in \mathcal{V}_{n,k}} \mathbb{E}\{d_{i,l} \,|\, k\Delta t \leq t_{i,l}^{\mathrm{a}} < (k+1)\Delta t\}, n \in \mathcal{N}.$$

Meanwhile, the average Packet Drop Ratio (PDR) of UE quantifies the communication reliability, which should be considered as another metric of QoS. It is defined as,

$$\bar{\beta}_{n,k} = \frac{1}{|\mathcal{V}_{n,k}|} \sum_{i \in \mathcal{V}_{n,k}} \mathbb{E}\{L_{i,l} \,|\, k\Delta t \leq t_{i,l}^{\mathrm{a}} < (k+1)\Delta t\}, n \in \mathcal{N}.$$

Then, we define the reward function for slice n at the k-th epoch as

$$J_{n,k} = \underbrace{\alpha_{n,1} \cdot U_{\mathrm{QoS}}^{\mathrm{PDR}}\left(\bar{\beta}_{n,k}, \bar{\beta}_n^{\max}, \bar{\beta}_n^{\min}\right)}_{\mathrm{PDR\ related\ reward}}$$
$$+ \underbrace{\alpha_{n,2} \cdot U_{\mathrm{QoS}}^{\mathrm{Lat}}\left(\bar{d}_{n,k}, \bar{d}_n^{\max}, \bar{d}_n^{\min}\right)}_{\mathrm{Packet\ Delay\ related\ reward}}, \quad (5)$$

$$U_{\text{QoS}}^{\text{PDR}}\left(\bar{\beta}_{n,k}, \bar{\beta}_n^{\max}, \bar{\beta}_n^{\min}\right) = \begin{cases} 1, & \bar{\beta}_n^{\min} > \bar{\beta}_{n,k} \geq 0, \\ \left(\bar{\beta}_n^{\max} - \bar{\beta}_{n,k}\right) \big/ \left(\bar{\beta}_n^{\max} - \bar{\beta}_n^{\min}\right), & \bar{\beta}_n^{\max} > \bar{\beta}_{n,k} \geq \bar{\beta}^{\min}, \\ 0, & \bar{\beta}_{n,k} > \bar{\beta}_n^{\max}, \end{cases} \tag{6}$$

$$U_{\text{QoS}}^{\text{Lat}}\left(\bar{d}_{n,k}, \bar{d}_n^{\max}, \bar{d}_n^{\min}\right) = \begin{cases} 1, & \bar{d}_n^{\min} > \bar{d}_{n,k} > 0, \\ \left(\bar{d}_n^{\max} - \bar{d}_{n,k}\right) \big/ \left(\bar{d}_n^{\max} - \bar{d}_n^{\min}\right), & \bar{d}_n^{\max} > \bar{d}_{n,k} \geq \bar{d}_n^{\min}, \\ 0, & \bar{d}_{n,k} \geq \bar{d}_n^{\max}, \end{cases} \tag{7}$$

where $U_{\text{QoS}}^{\text{PDR}}(\cdot)$ is a normalized reward function of average PDR $\bar{\beta}_{n,k}$ of slice n. To stabilize the learning procedure of the proposed DRL algorithm developed in Section IV, $U_{\text{QoS}}^{\text{PDR}}(\cdot)$ is designed as a piecewise-linear concave function in (6), where $\bar{\beta}_n^{\min}$ and $\bar{\beta}_n^{\max}$ denotes the min (target) and maximum tolerant PDR for V2V service in for slice n. $\alpha_{n,1}$ is the maximum revenue, when $\bar{\beta}_{n,k}$ is less than target PDR value $\bar{\beta}_n^{\min}$.

Meanwhile, $U_{\text{QoS}}^{\text{Lat}}(\cdot)$ in (7) is a normalized reward function of average packet delay $\bar{d}_{n,k}$ of slice n, where $\bar{d}_n^{\max}$ and $\bar{d}_n^{\min}$ denote the maximum tolerant value and minimum (target) of packet delay for V2V service in slice n. $\alpha_{n,2}$ is the maximum revenue, when $\bar{d}_{n,k}$ is less than minimum packet delay $\bar{d}_n^{\max}$.

Therefore, at each epoch k, the reward function of all slices can be defined as

$$J_k\left(\boldsymbol{O}_k\right) = \sum_{n \in \mathcal{N}} J_{n,k}. \tag{8}$$

The goal of this paper is to find the optimal network slicing configuration policy with weights $\boldsymbol{\theta}^*$ that can maximize the long-term reward of all slices, which can be formulated as

$$\max_{\boldsymbol{\theta}} \left\{ J\left(\pi_{\boldsymbol{\theta}}\right) = \mathbb{E}\left[\sum_{k=1}^{\infty} \lambda^{k-1} J_k\left(\boldsymbol{O}_k\right) \Big| \pi_{\boldsymbol{\theta}}\right]\right\}, \tag{9}$$

where λ is the discount factor.

C. PoMDP

Since the network slicing configuration policy π, defined in **Definition 3**, is based on the partial observation of network status, $\boldsymbol{O}_k$, instead of the complete network status. Therefore, the formulated problem (9) can be treated as a PoMDP with an infinite horizon discounted reward. PoMDP is an extension of MDP by adding a set of observations and the corresponding observation model [22], which is defined as follows,

- **System State**: The complete network status at k-th epoch is denoted as $\boldsymbol{X}_k$, which follows Markovian, but cannot be directly observed by eNB;
- **Observation**: At each epoch, the eNB (agent) indirectly observes the complete network status $\boldsymbol{X}_k$ through observation $\boldsymbol{O}_k$ in **Definition 1**, which can be seen as a stochastic function of $\boldsymbol{X}_k$;
- **Action**: The action of the controller is configuration of slices $\boldsymbol{C}_k$ in **Definition 3** and the discrete action space is $\mathcal{C}$ (the collection of candidate slice configuration);
- **Observation History**: The observation history at the k-th epoch $\boldsymbol{H}_k$, in **Definition 2**;
- **Reward Function**: It is the revenue of all slice at each epoch $J_k(\boldsymbol{O}_k)$ in formula (9);
- **Q Function**: It is expected long-term revenue from taking action $\boldsymbol{C}_k$ under observation history $\boldsymbol{H}_k$, which is

$$Q\left(\boldsymbol{H}_k, \boldsymbol{C}_k\right) = \mathbb{E}_{\tau > k}\left[\sum_{k'=k}^{\infty} \lambda^{k'-k} J_{k'}\left(\boldsymbol{O}_{k'}\right) | \boldsymbol{H}_k, \boldsymbol{C}_k\right],$$

where $\tau > k$ refers to the sampling trajectory of observations and actions after epoch k,

$$\tau > k = \left(\boldsymbol{O}_k, \boldsymbol{C}_k, \boldsymbol{O}_{k+1}, \boldsymbol{C}_{k+1}, \cdots\right).$$

- **Value Function**: It represents the expected long-term revenue starting from observation history $\boldsymbol{H}_k$ and the relationship between Q function and value function is,

$$V\left(\boldsymbol{H}_k\right) = \sum_{\boldsymbol{C} \in \mathcal{C}} \pi_{\theta}\left(\boldsymbol{C} | \boldsymbol{H}_k\right) \cdot Q\left(\boldsymbol{H}_k, \boldsymbol{C}\right),$$

where $\pi_{\boldsymbol{\theta}}\left(\boldsymbol{C} | \boldsymbol{H}_k\right)$ is the probability of choosing configuration $\boldsymbol{C} \in \mathcal{C}$ at observation history $\boldsymbol{H}_k$, under stochastic policy $\pi_{\boldsymbol{\theta}}$.

Unfortunately, PoMDP is a very difficult to solve in general and directly solving it suffers the high computational complexity. However, we need to emphasize that utilizing existing RL methods for the problem (9) will raise the following challenge.

Challenge 1: A key assumption underlying majority of RL algorithms is the full observability of system status. However, in this paper, only a partial observation of vehicular network status is available for the eNB, which makes the existing RL methods inadequate.

V. SOLUTION BASED ON ACTOR-CRITIC DEEP REINFORCEMENT LEARNING

Deep Reinforcement Learning (DRL) can apply to a wide range of control problems, since ANN can extract high-level features from raw input data and provide a good approximation of objective functions. Therefore, in this section, we develop a DRL algorithm that can deal with PoMDP problem (9). Firstly, we briefly explain the principles of Actor-Critic based RL and show its potential for solving problem (9). Then, to deal with **Challenge 1**, we propose an Actor-Critic DRL algorithm, which can obtain the optimal slicing configuration policy $\pi_{\boldsymbol{\theta}^*}$ from the observation history of vehicular networks, without requiring prior expert knowledge of networks.

A. Actor-Critic based RL for Solving the Formulated Problem

Generally, there are two categories of RL methods: 1) (e.g. Q-learning) and 2) RL based on policy search (e.g. policy gradient) [22]. Under the basic assumption of Markovian property, the value-based RL methods construct a value/Q function model for estimating how good each state, or state-action pair is, and then search for the optimal policy implicitly by optimizing the value/Q function. The RL based on value function have a good sampling efficiency and stable performance, but at cost of introducing bias in estimating of the value/Q function. On the other hand, without maintaining

a value function model, the policy search methods directly search for the optimal policy by the approximated gradient with respect to the parameters of policy. Compared to the value-based RL methods, policy search methods can obtain a good policy with a faster convergence rate, which can be extended to the non-Markovian scenarios (e.g. PoMDP problems). However, this category usually tends to converge to a local optimal and suffer from higher variance and lower sample efficiency.

To deal with these disadvantages, the Actor-Critic method, a hybrid of both policy-based and value-based method, is proposed. Particularly, as comparison of the value-based methods and the policy-based methods, we highlight two key advantages of the actor-critic methods in the following:

- The actor-critic methods can be applied for non-Markovian scenario, such as the PoMDP problem (9), where only the observation of vehicular networks is available at the controller;
- It can balance the trade-off between the variance of policy gradient and bias of value function estimation, as well as the satisfactory convergence property.

Thus, we utilize the actor-critic method to solve problem (9). The "actor part" updates the policy in the direction given by the "critic part", that is,

a). **The Actor Part**: It uses the policy gradient method to search the best performing policy over a set of parametrized policies π_θ, where vector θ is the parameters of RAN slicing policy π defined in ***Definition 3***. It is assumed that the policy π_θ is differentiable with respect to parameter vector θ, and the gradient of the objective function $J(\theta)$ in the problem (9) is denoted as $\nabla_\theta J(\pi_\theta)$. Then, the maximum of the objective function $J(\theta)$ can be obtained by ascending the gradient of the objective function $\nabla_\theta J(\pi_\theta)$. The policy gradient update for the parameter vector θ is given by

$$\theta \leftarrow \theta + \eta \cdot \nabla_\theta J(\pi_\theta) \quad (10)$$

where $\eta > 0$ is the learning rate for the policy update.

b). **The Critic Part**: According a value function estimation model, the goal of the critic part is to evaluate the performance of the policy π_θ and use it to calculate $\nabla_\theta J(\pi_\theta)$. For problem (9), we can design an ANN to approximate the value function and update the weights of ANN utilizing the observation data set of vehicular networks.

Therefore, we aim at designing a mode-free DRL algorithm with the actor-critic structure to solve problem (9). However, two main technical challenges arise as follows:

- **Challenge 2**: How to deduce the policy gradient $\nabla_\theta J(\pi_\theta)$ for the PoMDP problem (9)? Since existing policy gradient methods can only apply to the Markovian scenario.
- **Challenge 3**: The input of the RAN slicing policy π_θ is observation history H_k, which is a time sequence. How to perform temporal abstraction of H_k in a saleable way?

B. Proposed DRL Algorithm with Actor-Critic Structure

Firstly, we start with the policy gradient $\nabla_\theta J(\pi_\theta)$ to deal with ***Challenge 2***. It is noteworthy that the detailed design of DRL algorithm will be discussed later. Here, we propose the customized policy gradient for PoMDP.

Theorem 1 (Proposed Advantage Actor-Critic DRL algorithm for PoMDP Problem): Following the idea of Advantage Actor-Critic (A2C) methods for the MDP problems, the policy gradient $\nabla_\theta J(\pi_\theta)$ for the formulated PoMDP problem (9) is given by,

$$\nabla_\theta J(\pi_\theta) =$$
$$\mathbb{E}_\tau \left[\sum_{k=1}^\infty \lambda^{k-1} \cdot \nabla_\theta \log \pi_\theta (C_k | H_k) \cdot A(H_k, C_k) \right], \quad (11)$$

where $\tau = (O_1, C_1, O_2, ...)$ is a trajectory of network status, and function $A(H_k, C_k)$ is the advantage function, which is

$$A(H_k, C_k) = Q(H_k, C_k) - V(H_k). \quad (11a)$$

Proof: Detailed derivations are given in **Appendix A**. ∎

Remark 4: **Theorem 1** acquires an explicit form of policy gradient $\nabla_\theta J(\pi_\theta)$, where the policy gradient (11) is an expectation over entire trajectory τ of vehicular networks. We can compute gradient (11) approximately by using Monte-Carlo estimation. Specifically, the paradigm of policy gradient in (11) is like the concept to the maximum likelihood (ML) approaches in supervised learning, except that the policy gradient is weighted by sums of advantage functions over the trajectory. In fact, these sums of advantage functions may be positive or negative, thus the policy gradient will try to decrease the likelihood of samples with "negative sums of advantage functions" and increase the likelihood of others.

Based on **Theorem 1**, we propose a DRL algorithm with actor-critic structure. In the this framework, the critic part utilizes an ANN to approximate the advantage function $A(H_k, C_k)$, which is defined in formula (11a), while the actor part utilizes the gradient formula (11) to estimate the policy gradient $\nabla_\theta J(\pi_\theta)$ and then updates the parameter vector θ of the RAN slicing policy π_θ according to the formula (10) in Section IV-A.

C. Implementation of DRL with Actor-Critic Structure

1) Critic Part: Compared to the Q-function $Q(H_k, C_k)$ and the advantage function $A(H_k, C_k)$, the value function $V(H_k)$ is the simplest one since it only depends on the observation history H_k and thus is hoped to be easier for the critic part to learn. With the value function $V(H_k)$, the Q-function $Q(H_k, C_k)$ can be approximated by sample value. Then, the advantage function $A(H_k, C_k)$ can be approximated as follows

$$\hat{A}(H_k, C_k) = \sum_{k'=k}^\infty \lambda^{k'-k} J_{k'}(O_{k'}) - V(H_k), \quad (12)$$

where O_k and H_k are sampled through Monte-Carlo method.

Therefore, the critic part approximates the value function $V(H_k)$, and then estimate the advantage function $A(H_k, C_k)$. Like the idea of utilizing the Deep Q-Network (DQN) to fit the Q function, we introduce a critic neural network $\hat{V}$ to approximate the real value function $V(H_k)$. However, as mentioned in **Challenge 2**, the critic neural network $\hat{V}$ is a function in terms of the observation history at the k-th epoch H_k. To perform the temporal abstraction of observation history H_k, we modify $\hat{V}$ by leveraging recent advances in recurrent neural networks (RNN), that is, replacing the first fully-connected layer with a Long Short-Term Memory (LSTM) layer, which is regarded

as a memory cell with three different gates, which regulating the information and thus allowing to keep the past information [23]. Therefore, the LSTM layer can capture the longer-term temporal dependencies of observation history $\boldsymbol{H}_k$ as compared to the traditional RNNs.

Therefore, the critic neural network $\hat{V}$ is represented as

$$V\left(\boldsymbol{H}_k\right) \approx \hat{V}_w(\hat{\boldsymbol{H}}_k).$$

where vector $\boldsymbol{w}$ is the weights of the critic neural network $\hat{V}$ and term

$$\hat{\boldsymbol{H}}_k = \left(\boldsymbol{O}_{k-K}, ..., \boldsymbol{O}_{k-1}\right).$$

is a finite fixed-length window of past observations, which consists of the $K \in \mathbb{N}^+$ most recent observations of vehicular networks and actions to the k-th epoch.

In the learning process of the critic neural network $\hat{V}$, weights $\boldsymbol{w}$ is learned by minimizing the Mean Square Error (MSE) loss function at each learning step, which is given by

$$L\left(\boldsymbol{w}\right) = \frac{1}{2} \cdot \mathbb{E}\left\{\left[\hat{V}_{\boldsymbol{w}}(\hat{\boldsymbol{H}}_k) - \sum_{k'=k}^{\infty} \lambda^{k'-k} J_{k'}\left(\boldsymbol{O}_{k'}\right)\right]^2\right\}. \quad (13)$$

Then, the weights $\boldsymbol{w}$ of the critic neural network $\hat{V}_{\boldsymbol{w}}(\hat{\boldsymbol{H}}_k)$ are updated as

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \nu \cdot \nabla_{\boldsymbol{w}} L\left(\boldsymbol{w}\right),$$

where ν is the learning rate for weights $\boldsymbol{w}$.

In the following, we describe the detailed structure of the critic neural network $\hat{V}_{\boldsymbol{w}}(\hat{\boldsymbol{H}}_k)$.

a). *Input*: The input $\boldsymbol{O}_k$ (i.e. observation of vehicular networks) to the critic neural network is a vector size $2N$, where the $2n+1$-th to $2(n+1)$-th input entries corresponds to the observation of slice $n \in \mathcal{N}$ at the k-th epoch, $\boldsymbol{O}_{n,k}$.

b). *LSTM layer*: It maintains an internal state and aggregate observation states over time. This gives the critic neural network is responsible of learning how to aggregate observation states over time. We use the Rectified Linear Unit (ReLU) as the activation function for the LSTM layer.

c). *Hidden layers*: The number of neurons of each hidden layer is the same, and ReLU function is used as the activation function.

d). *Output layer*: The output of the DQN is a scalar, which is the estimated value of function $V(\boldsymbol{H}_k)$ under current observation history $\boldsymbol{H}_k$.

2) Actor Part: Based on **Theorem 1** and policy gradient update equation (10), the weights $\boldsymbol{\theta}$ of RAN slicing policy $\pi_{\boldsymbol{\theta}}$ are updated as

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \cdot \nabla_{\boldsymbol{\theta}} J\left(\pi_{\boldsymbol{\theta}}\right)$$

where $\eta \in \mathbb{R}^+$ is the learning rate for the update of parameter $\boldsymbol{w}$ and

$$\nabla_{\boldsymbol{\theta}} J\left(\pi_{\boldsymbol{\theta}}\right) \approx \sum_k \left[\lambda^{k-1} \cdot \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}\left(\boldsymbol{C}_k \mid \boldsymbol{H}_k\right) \cdot \hat{A}\left(\boldsymbol{H}_k, \boldsymbol{C}_k\right)\right],$$

and the advantage function $\hat{A}(\boldsymbol{H}_k, \boldsymbol{C}_k)$ is defined in (12).

Furthermore, to evaluate the training performance of the actor neural network $\pi_{\boldsymbol{\theta}}$, we define the loss function of the actor neural network as

$$L\left(\boldsymbol{\theta}\right) = \sum_k \left[\lambda^{k-1} \cdot \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}\left(\boldsymbol{C}_k \mid \boldsymbol{H}_k\right) \cdot \hat{A}\left(\boldsymbol{H}_k, \boldsymbol{C}_k\right)\right]. \quad (14)$$

Actor neural network $\pi_{\boldsymbol{\theta}}$ has the same structure as the critic neural network $\hat{V}_{\boldsymbol{w}}$. It is noteworthy that the actor neural network and the critic neural network share the same LSTM layer, as illustrated in Figure 3. This setting can make the actor and critic neural network shares the same hidden states of LSTM layer, which makes DRL training more stable. The output layer of policy $\pi_{\boldsymbol{\theta}}$ is described as follows: the output of policy $\pi_{\boldsymbol{\theta}}$ is a vector of size $|\mathcal{C}|$, where each element of the output layer is mapping to the probability of candidate configuration $\boldsymbol{C} \in \mathcal{C}$ under current observation history $\hat{\boldsymbol{H}}_k$.

Fig. 3. The structure of the critic neural network and actor neural network.

The overall learning procedure of our proposed advantage actor-critic DRL algorithm is provided in **Algorithm 1**.

Algorithm 1. Training of the RAN slicing policy $\pi_{\boldsymbol{\theta}}$

Initialization:
  Initialize the critic neural network $\hat{V}$ and the RAN slicing policy (i.e., the actor part) $\pi_{\boldsymbol{\theta}}$ with weights $\boldsymbol{w}$ and $\boldsymbol{\theta}$. Initialize the replay buffer R. Initialize the length of observation history K and the empty observation history.

Repeat:
  1) Receive observation of vehicular networks $\boldsymbol{O}_k$.
  2) Append observation and previous slice configuration to history, $\hat{\boldsymbol{H}}_k \leftarrow (\hat{\boldsymbol{H}}_{k-1}, \boldsymbol{O}_{k-1})$.
  3) Select slice configuration, $\boldsymbol{C}_k \leftarrow \pi_{\boldsymbol{\theta}}(\boldsymbol{H}_k)$.
  4) Store the sample trajectory $\{(\boldsymbol{H}_k, \boldsymbol{C}_k) : k = 1, ..., T\}$.
  5) Update the critic neural network $\hat{V}_{\boldsymbol{w}}$ using equation (13),

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \nu \cdot \nabla_{\boldsymbol{w}} L\left(\boldsymbol{w}\right).$$

  6) Update the RAN slicing policy (the actor part) $\pi_{\boldsymbol{\theta}}$ using equation (14)

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \cdot \nabla_{\boldsymbol{\theta}} J\left(\pi_{\boldsymbol{\theta}}\right)$$

Repeat:

---

3) Complexity of the Proposed DRL algorithm : According to analysis method in [24], the computational complexity of learning procedure for the proposed DRL algorithm (i.e., **Algorithm 1**) can be expressed by

$$\mathcal{O}\left(T_{\mathrm{L}} \cdot \left(\sum_{l=0}^{L_{\mathrm{actor}}} n_{\mathrm{actor}}^{(l)} \cdot n_{\mathrm{actor}}^{(l+1)} + \sum_{l=0}^{L_{\mathrm{critic}}} n_{\mathrm{critic}}^{(l)} \cdot n_{\mathrm{critic}}^{(l+1)}\right)\right),$$

where T_{L} is the learning steps of slicing configuration policy training, $n_{\mathrm{actor}}^{(l)}$ is the number of neurons in the l-th layer of the actor part, i.e., neural network $\pi_{\boldsymbol{\theta}}$, $n_{\mathrm{critic}}^{(l)}$ is the number of neurons in the l-th layer of the critic neural network, i.e., neural network $\hat{V}_w$, and L_{actor} (L_{critic}) denotes the number of the hidden layers in the actor part (critic part).

VI. SIMULATION RESULTS AND ANALYSIS

In order to demonstrate the effectiveness of our proposed network slice self-configuration scheme, a system level simulation platform is implemented. Herein, we consider a six-lane freeway and each direction has three lanes, where the

TABLE II
DEFAULT PARAMETER SETTINGS FOR SIMULATION.

Parameter		Assumption	
Carrier frequency/Bandwidth/Number of RBs		5.9 GHz/ 10 MHz/ 50	
Pathloss model/Small-scale fading		WINNER+ B1/ Rician fading	
Total Transmit Power of VUE		20 dBm	
The length of each epoch		400 ms (i.e., 400 slot)	
Absolute vehicle speed		70 km/h	
Average number of VUEs in the vehicular network		100	
Service type		Traffic safety related service	Autonomous driving related service
Weighting factors in utility function (5)		$\boldsymbol{\alpha}_1 = [1,2]$	$\boldsymbol{\alpha}_2 = [1,3]$
Packet size per UE		300 Byte	200 Byte
Packet arrival period		50 ms	25 ms
Candidate slicing configurations	F_n	2 or 3 or 4	2 or 3 or 4
	B_n	1.44 or 2.16 MHz	1.08 or 1.44 MHz
	T_n^{sw}	30 ms or 50 ms	25 ms or 15 ms
The number of candidate slicing configurations		36	

length of the freeway is 3.4 km and the width of lane is set as 4 m (A 1.2, Annex A, 3GPP 36.885 [9]). Software including MATLAB 2019a and Keras 2.2.2 with Python 3.5.2. are utilized for simulations. There are two types of services and two corresponding slices are considered in the simulation: a) network slice for traffic safety related service, which aims at reducing the possibility of traffic accidents and improvement of traffic efficiency; b). network slice for autonomous driving related service, which is utilized for the cooperative awareness and control between autonomous vehicles. Since the critical nature of communication reliability in V2V services, we set higher weighting factor for the PDR related function $U_{\mathrm{QoS}}^{\mathrm{PDR}}$ in the reward function (5). The discount rate λ in the formulated problem (9) for estimating long-term reward function is 0.9.

Based on the observation of vehicular networks $\boldsymbol{O}_k$, the proposed DRL algorithm trains the RAN slicing policy by **Algorithm 1**. Furthermore, in **Algorithm 1**, the critic neural network $\hat{V}$ is a four layers neural network. The LSTM layer contains 256 units and uses Rectified Linear Unit (ReLU) as the activation function. There is one hidden layers in $\hat{V}$. The hidden layer contains 64 units and uses ReLU as the activation function. With linear activation function, the output layer gives the estimated value function. On the other hand, the actor neural network $\pi_{\boldsymbol{\theta}}$ is has same structure to the critic neural network $\hat{V}$, which also has one hidden layer. Besides, the critic neural network $\hat{V}$ and the actor neural network $\pi_{\boldsymbol{\theta}}$ are learned with a learning rate of 10^{-4}.

A. Training Performances

The first experiment aims to examine the convergence of the proposed actor-critic DRL algorithm. Under the default simulation setup, as shown in Figure 4, we plot the variations in the loss functions of the critic neural network $\hat{V}_{\boldsymbol{w}}$ and the actor neural network $\pi_{\boldsymbol{\theta}}$ under different settings of learning rate. This metric can measure the convergence speed of the loss function during the training procedure in Algorithm 1. In Figure 4(a), under default learning rate 10^{-4}, the values of loss function $L(\boldsymbol{w})$ decrease close to 0.02 after the 300 learning steps, which means the value function $\hat{V}_{\boldsymbol{w}}$ has reached to a stable performance. On the other hand, in Figure 4(b),

(a) Training Loss of Critic neural network

(b) Training Loss of Actor neural network

Fig. 4. Illustration of the convergence of the proposed DRL algorithm under different learning rates: 1) the train loss of critic neural network $\hat{V}_{\boldsymbol{w}}$ is measured by loss function $L(\boldsymbol{w})$ defined in (13); 2). the loss function of actor neural network $\pi_{\boldsymbol{\theta}}$ (i.e., the slice self-configuration policy) is estimated by loss function $L(\boldsymbol{\theta})$ defined in (14).

the values of loss function $L(\boldsymbol{\theta})$ decrease quickly during the first 300 learning steps. After the first 300 learning steps, the value curve of the $L(\boldsymbol{\theta})$ is convergent to the value of about 0.04, which indicates the slice self-configuration policy has evolved into convergence condition. The results confirm that the proposed DRL algorithm can avoid the mis-convergence and unstable issues in the training procedure.

Since the learning rate is one crucial hyper-parameter in deep learning, we compare the learning trends under different learning rates. It can be seen that, with a higher value of learning rate, the convergence speed of both actor and critic neural networks is increasing, but at the cost of higher training loss and drastic fluctuation at the convergence condition. For instance, when learning rate equals to 3×10^{-4}, the train loss is "stable" after 100 learning steps, but the value of train loss is dramatically is dramatically fluctuating. Therefore, there is a trade-off between convergence speed and convergence value of training loss.

B. Performance Evaluation

Fig. 5. CDF of the reward function in formula (8) with different schemes under default settings: reward function J can reflect the overall performance of network slicing control–higher value of the function J, the better the QoS performance of network slices.

For the performance comparisons, we consider the Deep Recurrent Q-Network (DRQN), one of state-of-art DRL algorithms, as the baseline scheme [8]. In the DRQN scheme, it replaces the first layer of the DQN with a LSTM layer with 256 units and ReLU activation function. In the simulation, DRQN is a four layers neural network. In the input LSTM layer, there is one input, i.e., the observation of vehicular networks $\boldsymbol{O}_k$. Two hidden layers, each contains 128 units with the ReLU activation function. The output layer has the same size as the number of all candidate slicing configurations (i.e., $I_{\mathcal{C}}$), and each element of the output layer is mapping to an estimated value of $Q(\boldsymbol{O}_k, \boldsymbol{C}_k)$ under the observation $\boldsymbol{O}_k$. In the baseline scheme, the DRQN is learned by using the Adam algorithm with a learning rate of 10^{-4}, and weights of the target DRQN are copied from the weights of DRQN every 200 learning steps. Besides, ϵ-greedy rule is used in the baseline scheme.

(a) The slice for the autonomous driving related service.

(b) The slice for the traffic safety related service.

Fig. 6. QoS performance of each network slice versus different vehicle density (i.e., the average number of VUEs in the network) under different schemes.

The parameters of ϵ-greedy rule are set as $\epsilon_{\min} = 0.01$ and $\epsilon_{\text{decay}} = 0.01$.

From a statistical point of view, we evaluate the reward function defined in (8), which is used to indicate the QoS performance of slicing control. Figure 5 presents the cumulative distribution functions (CDFs) of the reward function with the proposed scheme and state-of-the-art DRQN scheme. It shows that the overall performance of the proposed scheme (mean value is 3.924) is better than the DRQN scheme (mean value is 3.276). Specifically, the proposed DRL scheme can improve around 20% than that of the DRQN scheme. The main reason is that, compared to the proposed DRL algorithm, the DRQN scheme, one kind of value-based DRL approaches, is introducing a higher level of bias in the estimation of the Q-value function, and then making the sub-optimal decisions. Then, the DRQN scheme mis-estimates the real status of network slices, and then make inappropriate adaption of slice configuration, which may deteriorate the PDR performance of each slice. In the following part, more experiment are carried out to show the RAN slicing performance of different schemes.

As shown in the Figure 6, it depicts the QoS performance of two considered network slices versus different vehicle density

(i.e., the average number of VUEs in the network) under the DRQN and the proposed scheme. Figure 6(a) depicts the QoS performance of UE in the slice for the autonomous driving related service. It can be seen that the average packet delay and PDR of UE decreases on the increasing vehicle density. Compared with the DRQN based network slicing, the proposed scheme has both lower average packet delay and PDR of UE. On the other hand, Figure 6(b) depicts the QoS performance of UE in the slice for the traffic safety related service. It can be observed from Figure 6(b) that, in the slice for the traffic safety related service, the proposed scheme has better average packet latency performance than the DRQN scheme. However, the DRQN scheme has slight better average PDR performance than the proposed scheme.

The main reason is that, in the reward function (5), the higher weighting factor for the PDR related to the slice for the autonomous driving related service, the proposed scheme tends to choose the slice configuration with more sub-channels and wider sub-channel bandwidth for the slice for the autonomous driving related service. This slicing strategy can ensure and improve the reward function but at cost of sacrificing the average PDR to a certain extent. The DRQN scheme is prone to choose the slice configuration with longer selection window length, which will decrease the average PDR but increase the average packet delay. Besides, when the vehicle density is increasing, the performance of DRQN is deteriorating rapidly. Overall, the result conforms to our expectations of the proposed DRL scheme for network slicing in C-V2X Mode 4 based networks.

VII. Conclusion

In this paper, we propose an intelligent semi-decentralized network slicing framework for the C-V2X Mode 4 networks, which aims at maximizing the long-term QoS performance of V2V services. Specifically, the proposed network slicing framework is implemented by a carefully designed actor-critic structured DRL algorithm. It has following advantages. Firstly, due to the proposed scheme has a semi-decentralize structure, the eNB only operates at a large timescale (in the level of hundreds of milliseconds), which has good scalability and can significantly reduce the signaling overhead. Meanwhile, because the eNB can infer the global view of vehicular networks from observation history, the decision-making process of the slice configuration at the eNB side can better ensure the control performance. Simulation results show that the proposed scheme has stable convergence control performance and achieves higher QoS performance as compared to the state-of-art baseline scheme. However, due to the lack of data generated in real V2V service traffic, the traffic model is assumed as 3GPP model, which has a limitation in characterizing the realistic scenario. Last but not least, the proposed DRL algorithm is essentially an on-policy leaner, which has lower sample efficiency. These issues need to be further studied.

Appendix

A. Proof of Theorem 1

Proof: Firstly, based on the definition of value function in (11), the long-term revenue $J(\pi_{\theta})$ can be re-written as

$$J(\pi_{\theta}) = \mathbb{E}\left[\sum_{k=1}^{\infty} \lambda^{k-1} J(\mathbf{O}_k) \Big| \pi_{\theta}\right] = \sum_{\mathbf{H}_1} \Pr[\mathbf{H}_1] V(\mathbf{H}_1),$$

where observation history $\mathbf{H}_1 = (\mathbf{O}_1)$. Then, gradient of $J(\pi_{\theta})$ can be written as

$$\nabla_{\theta} J(\pi_{\theta}) = \sum_{\mathbf{H}_1} \Pr[\mathbf{H}_1] \nabla_{\theta} V(\mathbf{H}_1).$$

Thus, our focus is to obtain the derivation of the value function, (i.e. $\nabla_{\theta} V(\mathbf{H}_k)$, $k = 1, 2, ...$),

$$
\begin{aligned}
&\nabla_{\theta} V(\mathbf{H}_k) \\
&= \nabla_{\theta}\left(\sum_{\mathbf{C}_k \in \mathcal{C}} \pi_{\theta}(\mathbf{C}_k | \mathbf{H}_k) \cdot Q(\mathbf{H}_k, \mathbf{C}_k)\right) \\
&= \sum_{\mathbf{C}_k \in \mathcal{C}} [\nabla_{\theta}(\pi_{\theta}(\mathbf{C}_k | \mathbf{H}_k)) \cdot Q(\mathbf{H}_k, \mathbf{C}_k) \\
&\qquad\qquad + \pi_{\theta}(\mathbf{C}_k | \mathbf{H}_k) \cdot \nabla_{\theta}(Q(\mathbf{C}_k, \mathbf{H}_k))], \quad (15)
\end{aligned}
$$

where the gradient of Q function can be written as

$$
\begin{aligned}
&\nabla_{\theta}(Q(\mathbf{H}_k, \mathbf{C}_k)) = \\
&\nabla_{\theta}\left\{\sum_{\mathbf{O}_{k+1} \in \mathcal{O}} \Pr[\mathbf{H}_{k+1} | \mathbf{H}_k, \mathbf{C}_k] \cdot (J_k(\mathbf{O}_k) + \lambda \cdot V(\mathbf{H}_{k+1}))\right\} \\
&= \lambda \cdot \sum_{\mathbf{O}_{k+1} \in \mathcal{O}} \Pr[\mathbf{H}_{k+1} | \mathbf{H}_k, \mathbf{C}_k] \cdot \nabla_{\theta}(V(\mathbf{H}_{k+1})). \quad (16)
\end{aligned}
$$

Furthermore, for arbitrary policy π_{θ}, based on the employment of log-derivative trick, we have

$$
\begin{aligned}
&\nabla_{\theta}(\pi_{\theta}(\mathbf{C}_k | \mathbf{H}_k)) = \\
&\quad \pi_{\theta}(\mathbf{C}_k | \mathbf{H}_k) \cdot \log(\nabla_{\theta}(\pi_{\theta}(\mathbf{C}_k | \mathbf{H}_k))). \quad (17)
\end{aligned}
$$

Substituting equations (16) and (17) into gradient (15), then $\nabla_{\theta} V(\mathbf{H}_k)$ can be rewritten as formula (18), where

$$\phi(\mathbf{H}_k, \mathbf{C}_k) = \nabla_{\theta}(\log(\pi_{\theta}(\mathbf{C}_k | \mathbf{H}_k))) \cdot Q(\mathbf{H}_k, \mathbf{C}_k).$$

Equation (18) has a nice recursive form and the future state value function $V(\mathbf{H}'_k)$ ($k' = k + 1, k + 2, ...$) can be repeated unrolled by following the same equation.

Then, we keep on unrolling $V(\mathbf{H}'_k)$ in equation (18), we can obtain equation (19), where

$$
\begin{aligned}
&\Pr[\mathbf{H}_{k'} | \mathbf{H}_k; \pi_{\theta}] = \\
&\prod_{i=0}^{k'-k} \pi(\mathbf{C}_{k+i} | \mathbf{H}_{k+i}) \cdot \Pr[\mathbf{O}_{k+i+1} | \mathbf{H}_{k+i}, \mathbf{C}_{k+i}] \quad (20)
\end{aligned}
$$

is the probability of transitioning from $\mathbf{H}_k$ to $\mathbf{H}'_k$.

Direct use of equation (19) to estimate $\nabla_{\theta} V(\mathbf{H}_k)$ will induce high variance of gradient estimation. To deal with this issue, researchers propose an idea is to add a "baseline" that will not affect the expectation but reduce the variance. One such "baseline" can be derived using following reasoning:

For any policy π_{θ}, it is true that $\sum_{\mathbf{C}_{k'} \in \mathcal{C}} \pi_{\theta}(\mathbf{C}_{k'} | \mathbf{H}_{k'}) = 1$. Then, taking the gradient ∇_{θ} from both sides and utilizing equation (17), we can obtain:

$$
\begin{aligned}
0 &= \sum_{\mathbf{C}_{k'} \in \mathcal{C}} \nabla_{\theta}(\pi_{\theta}(\mathbf{C}_{k'} | \mathbf{H}_{k'})) \\
&= \sum_{\mathbf{C}_{k'} \in \mathcal{C}} \pi_{\theta}(\mathbf{C}_{k'} | \mathbf{H}_{k'}) \cdot \nabla_{\theta}(\log(\pi_{\theta}(\mathbf{C}_{k'} | \mathbf{H}_{k'}))). \quad (21)
\end{aligned}
$$

Multiplying the expression (21) with some value independent of $\mathbf{C}_{k'}$, e.g., $\lambda^{k'-k} V(\mathbf{H}_{k'})$, we have

$$\sum_{\mathbf{C}_{k'} \in \mathcal{C}} \pi_{\theta}(\mathbf{C}_{k'} | \mathbf{H}_{k'}) \nabla_{\theta}(\log(\pi_{\theta}(\mathbf{C}_{k'} | \mathbf{H}_{k'}))) \lambda^{k'-k} V(\mathbf{H}_{k'}) = 0.$$

Adding this equation into gradient formula (19), $\nabla_{\theta} V(\mathbf{H}_k)$ can be rewritten as follows,

$$
\begin{aligned}
&\nabla_{\theta} V(\mathbf{H}_k) = \\
&\mathbb{E}_{\tau > k}\left[\sum_{k'=k}^{\infty} \lambda^{k'-k} \nabla_{\theta}(\log(\pi_{\theta}(\mathbf{C}_{k'} | \mathbf{H}_{k'}))) \cdot A(\mathbf{H}_{k'}, \mathbf{C}_{k'}) \Big| \mathbf{H}_k; \pi_{\theta}\right],
\end{aligned}
$$

$$\nabla_{\boldsymbol{\theta}} V\left(\boldsymbol{H}_k\right)=\sum\nolimits_{\boldsymbol{C}_k \in \mathcal{C}}\left[\pi_{\boldsymbol{\theta}}\left(\boldsymbol{C}_k \mid \boldsymbol{H}_k\right) \cdot \phi\left(\boldsymbol{H}_k, \boldsymbol{C}_k\right)+\lambda \cdot \sum\nolimits_{\boldsymbol{O}_{k+1} \in \mathcal{O}} \pi_{\boldsymbol{\theta}}\left(\boldsymbol{C}_k \mid \boldsymbol{H}_k\right) \cdot \operatorname{Pr}\left[\boldsymbol{H}_{k+1} \mid \boldsymbol{H}_k, \boldsymbol{C}_k\right] \cdot \nabla_{\boldsymbol{\theta}}\left(V\left(\boldsymbol{H}_{k+1}\right)\right)\right] \qquad (18)$$

$$\nabla_{\boldsymbol{\theta}} V\left(\boldsymbol{H}_k\right)=\sum\nolimits_{\boldsymbol{C}_k \in \mathcal{C}} \pi_{\boldsymbol{\theta}}\left(\boldsymbol{C}_k \mid \boldsymbol{H}_k\right) \cdot \phi\left(\boldsymbol{H}_k, \boldsymbol{C}_k\right)$$
$$+\lambda \cdot \sum\nolimits_{\boldsymbol{O}_{k+1} \in \mathcal{O}} \operatorname{Pr}\left[\boldsymbol{H}_{k+1} \mid \boldsymbol{H}_k ; \pi_{\boldsymbol{\theta}}\right] \cdot\left(\sum\nolimits_{\boldsymbol{C}_{k+1} \in \mathcal{C}} \pi_{\boldsymbol{\theta}}\left(\boldsymbol{C}_{k+1} \mid \boldsymbol{H}_{k+1}\right) \cdot \phi\left(\boldsymbol{H}_{k+1}, \boldsymbol{C}_{k+1}\right)\right)$$
$$+\lambda^2 \cdot \sum\nolimits_{\boldsymbol{O}_{k+1}, \boldsymbol{O}_{k+2} \in \mathcal{O}, \boldsymbol{C}_{k+1} \in \mathcal{C}} \operatorname{Pr}\left[\boldsymbol{H}_{k+2} \mid \boldsymbol{H}_k ; \pi_{\boldsymbol{\theta}}\right] \cdot\left(\sum\nolimits_{\boldsymbol{C}_{k+2} \in \mathcal{C}} \pi_{\boldsymbol{\theta}}\left(\boldsymbol{C}_{k+2} \mid \boldsymbol{H}_{k+2}\right) \cdot \phi\left(\boldsymbol{H}_{k+2}, \boldsymbol{C}_{k+2}\right)\right)+\cdots$$
$$=\mathbb{E}_{\tau>k}\left[\sum\nolimits_{k^{\prime}=k}^{\infty} \lambda^{k^{\prime}-k} \nabla_{\boldsymbol{\theta}}\left(\log \left(\pi_{\boldsymbol{\theta}}\left(\boldsymbol{C}_{k^{\prime}} \mid \boldsymbol{H}_{k^{\prime}}\right)\right)\right) \cdot Q\left(\boldsymbol{H}_{k^{\prime}}, \boldsymbol{C}_{k^{\prime}}\right) \Big| \boldsymbol{H}_k ; \pi_{\boldsymbol{\theta}}\right] \qquad (19)$$

where $A(\boldsymbol{H}_{k^{\prime}}, \boldsymbol{C}_{k^{\prime}})$ is the advantage function defined in (11a). Then the gradient of the objective function $J(\pi_{\boldsymbol{\theta}})$ is

$$\nabla_{\boldsymbol{\theta}} J\left(\pi_{\boldsymbol{\theta}}\right)=$$
$$\mathbb{E}_\tau\left[\sum_{k^{\prime}=1}^{\infty} \lambda^{k^{\prime}-1} \nabla_{\boldsymbol{\theta}}\left(\log \left(\pi_{\boldsymbol{\theta}}\left(\boldsymbol{C}_{k^{\prime}} \mid \boldsymbol{H}_{k^{\prime}}\right)\right)\right) \cdot A\left(\boldsymbol{H}_{k^{\prime}}, \boldsymbol{C}_{k^{\prime}}\right) \Big| \pi_{\boldsymbol{\theta}}\right].$$

Then, we obtain **Theorem 1**. ∎

REFERENCES

[1] "Global connected vehicle market, industry insights by growth, emerging trends and forecast by 2023," Kenneth Research, Tech. Rep., 2019.
[2] G. Fodor *et al.*, "Supporting enhanced vehicle-to-everything services by LTE release 15 systems," *IEEE Communications Standards Magazine*, vol. 3, no. 1, pp. 26–33, 2019.
[3] M. LiWang *et al.*, "A computation offloading incentive mechanism with delay and cost constraints under 5g satellite-ground iov architecture," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 124–132, 2019.
[4] M. Boban *et al.*, "Connected roads of the future: Use cases, requirements, and design considerations for Vehicle-to-Everything communications," *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 110–123, 2018.
[5] S. Chen *et al.*, "Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017.
[6] X. Shen, J. Gao, W. Wu *et al.*, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 45–66, Jan. 2020.
[7] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 7331–7376, June 2019.
[8] J. Mei, X. Wang, and K. Zheng, "Intelligent network slicing for V2X services toward 5G," *IEEE Network*, vol. 33, no. 6, pp. 196–204, Oct. 2019.
[9] *Study on LTE-based V2X services (Release 14)*, 3GPP TR 36.885 Std., July 2016.
[10] *Study on Enhancement of 3GPP Support for 5G V2X Services (Release 15)*, 3GPP TR 22.886 Std., dec. 2016.
[11] M. Gonzalez-Martín *et al.*, "Analytical models of the performance of C-V2X Mode 4 vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1155–1166, 2019.
[12] D. M. Mughal *et al.*, "Performance analysis of V2V communications: A novel scheduling assignment and data transmission scheme," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 7045–7056, 2019.
[13] F. Eckermann *et al.*, "Performance analysis of C-V2X mode 4 communication introducing an open-source C-V2X simulator," in *Proc. IEEE Vehicular Technology Conference (VTC2019-Fall)*, Honolulu, HI, USA, Sep. 2019, pp. 1–5.
[14] B. Toghi *et al.*, "Analysis of distributed congestion control in cellular Vehicle-to-Everything networks," in *Proc. IEEE Vehicular Technology Conference (VTC2019-Fall)*, Honolulu, HI, USA, Sep. 2019, pp. 1–7.
[15] S. Sabeeh *et al.*, "Estimation and reservation for autonomous resource selection in c-v2x mode 4," in *Proc. IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'19)*, Istanbul, Turkey, Sep. 2019, pp. 1–6.

[16] R. Molina-Masegosa, M. Sepulcre, and J. Gozalvez, "Geo-based scheduling for C-V2X networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8397–8407, June 2019.
[17] Q. Chen, H. Jiang, and G. Yu, "Service oriented resource management in spatial reuse-based C-V2X networks," *IEEE Wireless Communications Letters*, vol. 9, no. 1, pp. 91–94, Sep. 2020.
[18] P. Wang *et al.*, "Platoon cooperation in cellular V2X networks for 5G and beyond," *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 3919–3932, June 2019.
[19] Y. Chen, Y. Wang, M. Liu, J. Zhang, and L. Jiao, "Network slicing enabled resource management for service-oriented ultra-reliable and low-latency vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7847–7862, 2020.
[20] S. Zhang, H. Luo, J. Li *et al.*, "Hierarchical soft slicing to meet multi-dimensional qos demand in cache-enabled vehicular networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2150–2162, 2020.
[21] H. Yang *et al.*, "Twin-timescale radio resource management for ultra-reliable and low-latency vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1023–1036, Dec. 2020.
[22] L. Lei *et al.*, "Deep reinforcement learning for autonomous internet of things: Model, applications and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1722–1760, 2020.
[23] N. Heess *et al.*, "Memory-based control with recurrent neural networks," *arXiv preprint arXiv:1512.04455*, Dec. 2015.
[24] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y. Liang, "Intelligent resource scheduling for 5G radio access network slicing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7691–7703, Jun. 2019.

Jie Mei (S'18-M'19) received the B.S. degree from the Nanjing University of Posts and Telecommunications (NJUPT), China, in 2013, and the Ph.D. degree in information and communication engineering from the Beijing University of Posts and Telecommunications (BUPT) in June 2019. He is currently a Post-Doctoral Associate with the Electrical and Computer Engineering, Western University, Canada. His research interests include intelligent communications, multi-dimensional intelligent multiple access, and Vehicle-to-Everything (V2X) communication. He was a TPC member of IEEE Globecom 2020.

Dr. Xianbin Wang (S'98-M'99-SM'06-F'17) is a Professor and Tier-1 Canada Research Chair at Western University, Canada. He received his Ph.D. degree in electrical and computer engineering from the National University of Singapore in 2001.

Prior to joining Western, he was with Communications Research Centre Canada (CRC) as a Research Scientist/Senior Research Scientist between July 2002 and Dec. 2007. From Jan. 2001 to July 2002, he was a system designer at STMicroelectronics. His current research interests include 5G/6G technologies, Internet-of-Things, communications security, machine learning and intelligent communications. Dr. Wang has over 450 highly cited journal and conference papers, in addition to 30 granted and pending patents and several standard contributions.

Dr. Wang is a Fellow of Canadian Academy of Engineering, a Fellow of Engineering Institute of Canada, a Fellow of IEEE and an IEEE Distinguished Lecturer. He has received many awards and recognitions, including Canada Research Chair, CRC President's Excellence Award, Canadian Federal Government Public Service Award, Ontario Early Researcher Award and six IEEE Best Paper Awards. He currently serves/has served as an Editor-in-Chief, Associate Editor-in-Chief, Editor/Associate Editor for over 10 journals. He was involved in many IEEE conferences including GLOBECOM, ICC, VTC, PIMRC, WCNC, CCECE and CWIT, in different roles such as general chair, symposium chair, tutorial instructor, track chair, session chair, TPC co-chair and keynote speaker. He has been nominated as an IEEE Distinguished Lecturer several times during the last ten years. Dr. Wang is currently serving as the Chair of IEEE London Section and the Chair of ComSoc Signal Processing and Computing for Communications (SPCC) Technical Committee.

Kan Zheng (S'02-M'06-SM'09) received the B.S., M.S., and Ph.D. degrees from Beijing University of Posts and Telecommunications (BUPT), China, in 1996, 2000, and 2005, respectively, where he is currently a Professor. He has rich experiences on the research and standardization of the new emerging technologies. He is the author of more than 200 journal articles and conference papers in the field of wireless networks, IoT, vehicular communication, and so on. He holds editorial board positions for several journals and has organized several special issues, including IEEE COMMUNICATIONS SURVEYS & TUTORIALS, IEEE Communication Magazine, and IEEE SYSTEM JOURNAL.