Multi-agent DRL-based Task Offloading in Multiple RIS-aided IoV Networks

Bishmita Hazarika, *Student Member, IEEE*, Keshav Singh, *Member, IEEE*, Sudip Biswas, *Member, IEEE*, Shahid Mumtaz, *Senior Member, IEEE*, and Chih-Peng Li, *Fellow, IEEE*

***Abstract*—This paper considers an internet of vehicles (IoV) network, where multi-access edge computing (MAEC) servers are deployed at base stations (BSs) aided by multiple reconfigurable intelligent surfaces (RISs) for both uplink and downlink transmission. An intelligent task offloading methodology is designed to optimize the resource allocation scheme in the vehicular network which is based on the state of criticality of the network and the priority and size of tasks. We then develop a multi-agent deep reinforcement learning (MA-DRL) framework using the Markov game for optimizing the task offloading decision strategy. The proposed algorithm maximizes the mean utility of the IoV network and improves communication quality. Extensive numerical results were performed that demonstrate that the RIS-assisted IoV network using the proposed MA-DRL algorithm achieves higher utility than current state-of-the art networks (not aided by RISs) and other baseline DRL algorithms, namely soft actor-critic (SAC), deep deterministic policy gradient (DDPG), twin delayed DDPG (TD3). The proposed method improves the offloading data rate of the tasks, reduces the mean delay and ensures that a higher percentage of offloaded tasks are completed compared to that of other DRL-based and non-RIS-assisted IoV frameworks.**

***Index Terms*—Internet of vehicles (IoV), multi-access edge computing (MAEC), reconfigurable intelligent surface (RIS), multi-agent deep reinforcement learning (MA-DRL).**

I. Introduction

With the advent of 5G, the past decade has witnessed an enormous proliferation in the fields of the Internet of Things (IoT) and artificial intelligence (AI). This has led to immense inflation in the amount of data being generated from these applications on a daily basis. The wide application of vehicular networks based on AI has drawn extensive attention to internet of vehicles (IoV) networks which involve heterogeneous computation-intensive and delay-intolerant tasks. Some of the potential applications of the IoV networks include (but not limited to) tasks related to (i) Intelligent transportation system (ITS): enhancement of traffic management, congestion reduction, improved road safety, real-time information on traffic conditions, road conditions, and vehicle locations, (ii) Connected and autonomous vehicles (CAVs): V2V communication, V2I communication, etc., (iii) Telematics: remote diagnostics, predictive maintenance, vehicle tracking, etc., (iv) Navigation and location-based services: real-time navigation, traffic, and weather update, route optimization, etc., (v) Smart charging: optimized charging times, load balancing during peak hours, (vi) Infotainment services: entertainment and information services such as movie, music, news, current affairs, etc. Until recently, such tasks were mostly handled by the cloud, which has been a promising computing paradigm to provide adequate resources. Due to the centralized nature of the data centers, the cloud infrastructure faces specific challenges while dealing with large-scale and complex IoT applications involving mobility and geographical distribution that require very low latency communication. Such applications have strict delay constraints and cannot afford transmission latency or network congestion. Low latency communication is crucial for IoV networks since it directly affects the safety and efficiency of the vehicular and transportation system since vehicular tasks involve several highly integral and delay-intolerant applications related to road safety, traffic information, safety information, real-time decision-making such as navigation, lane changing, speed adjustment, surrounding information, etc. These tasks are critical tasks which, if not completed in time might compromise the safety and efficiency of the vehicle as well as the driver. For example, if a vehicle fail to detect an accident or object in front of the vehicle and fail to take lane-changing decision in time, it might lead to major accidents which can also prove to be life-threatening. Thus, ensuring fast, low latency, and reliable communication improve efficiency while ensuring the safety and sustainability of an IoV framework. Fog computing is an advanced decentralized architecture that uses edge devices to carry out a substantial amount of computation and communication locally and routed over the core network, thus solving the issues that plague cloud computing [1] [2]. However, the onboard resources of a low-cost consumer vehicle may not be able to accommodate the computing needs of a vehicular fog computing (VFC) network, thereby affecting the quality of experience (QoE) of the fog vehicular network. Accordingly, to meet the requirement of

The work of K. Singh and C.-P. Li was supported by the National Science and Technology Council of Taiwan, under grants NSTC 112-2218-E-110 -003 and MOST 109-2221-E-110 -050 -MY3. (*Corresponding author: Keshav Singh.*)

The work of S. Biswas was supported by the Science and Engineering Research Board (DST), India under the grant MTR/2022/000648.

This work is a result of the MoU (IIITG/NSYSU 20200728) signed between IIIT Guwahati and National Sun Yat-sen University, Taiwan.

Bishmita Hazarika, Keshav Singh, and Chih-Peng Li are with Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan (E-mail: me.bishmita@gmail.com, {keshav.singh, cpli}@mail.nsysu.edu.tw).

Sudip Biswas is with the Department of ECE, Indian Institute of Information Technology Guwahati, Assam 781015, India (E-mail:sudip.biswas@iiitg.ac.in).

Shahid Mumtaz is with Department of Applied Informatics Silesian University of Technology Akademicka 16 44-100 Gliwice, Poland and Nottingham Trent University, Departement of Engineering, Nottingham NG1 4FQ, UK (E-mail: shahid.mumtaz@ntu.ac.uk).

such networks researchers have introduced multi-access edge computing (MAEC) which can extend the computation power of a vehicular network by allowing the vehicles in an IoV network to offload part of their task to the MAEC server located on the roadside or at the BS [3]–[5].

Irrespective of the task offloading schemes in the physical layer, to ensure low delay and achieve high QoE, IoV networks require ubiquitous ultra-reliable low-latency high-rate wireless communications. Furthermore, long-range communications from vehicle to BS encounter several obstacles in the line-of-sight link such as trees, buildings, etc., which significantly affects the communication quality by blocking the line-of-sight (LoS) signal. In this regard, reconfigurable intelligent surface (RIS) has been envisioned as a revolutionary technology in the future 6G wireless communication systems, owing to its potential to actively customize the wireless propagation environments. Recently, the concept of RIS has generated a lot of attention in both the industry and academic communities as an exquisite way to improve the quality of wireless communication by adjusting the wireless propagation paths incident on the surface. The main advantage of RIS is that it can improve the quality of communication at the cost of very low energy consumption [6]. RIS, unlike traditional relay technologies, operates as a passive device for managing incoming signals without relying on radio frequency chains or complicated signal processing methods. This satisfies the demand of MAEC systems [7]. Thus, RIS combined with MAEC can further enhance the performance of these systems since end-users in such networks can offload tasks to the MAEC server via RIS, thereby enhancing the quality of the network. Accordingly, authors in [8] explored the concept of RIS-aided MAEC and proposed an optimal offloading decision strategy using DRL. Similarly, in [9] the authors studied the RIS-aided MAEC system and optimized the task scheduling strategy in a vehicular network. In particular, RIS is a panel that integrates a vast quantity of passive reflection elements, capable of precise reflect beamforming to amplify the signal strength at authorized receivers and prevent information leakage to unauthorized eavesdroppers through adjusting the phase of incoming signals [10]. The authors in [11] put forward a comprehensive survey of different RIS applications and their advantages such as enhancement in signal strength, physical layer security, and accuracy in deployment position. On the same note, the authors in [12] surveyed the optimality of RIS-assisted resource allocation techniques in vehicular networks while some address future research scope and direction for the same. In [13], the authors proposed an alternating optimization algorithm combined with a genetic algorithm to jointly optimize the active and passive beamforming at the RIS, the authors in [14] investigated the max-min computation efficiency problem for enhancing the security in task offloading. Explicitly, RIS technology has the potential to enhance the offloading rate and improve the physical layer security (PLS) of the devices/vehicles by optimizing its reflection coefficients [15]. Thus, RIS combined with MAEC can achieve substantial improvement in the overall performance of a task-offloading system.

Recently, the use of reinforcement learning (RL) in vehicular communication networks has seen a steep insurgence. In particular, in a VFC framework, the aim is to solve a sequential decision process, which can be formalized under the classical settings of RL, where the agent is required to learn and represent its environment as well as act optimally at a given instant. The optimal action is referred to as the policy. Since vehicular networks involve large-scale dynamic and complex environments, RL algorithms such as Q-learning might not be able to handle massive state spaces in some complex environments. For such environments, deep RL (DRL) algorithms can be an ideal alternative that uses neural networks (NNs) to increase the RL algorithm's scalability and adapt to the VFC environment [16], [17]. Accordingly, the authors in [18] minimize the energy consumption and formulate a DRL-based approach to offload the tasks to roadside units (RSU) or to other vehicles. Similarly, in [16], [19]–[21], the authors used a DRL-based approach for task offloading, while the authors in [22] proposed a dynamic pricing system for offloading tasks to an unmanned aerial vehicle (UAV) mounted in the MAEC server. The authors in [23] designed a dynamic task offloading strategy to achieve optimal task offloading using the twin delayed deep deterministic policy gradient algorithm (also known as TD3), which is one of the state-of-the-art DRL algorithms. Likewise, in [24], the authors formulated a task allocation problem in a UAV-assisted MAEC system and solved it using the TD3 algorithm.

While the use of DRL techniques has proven to achieve promising results, recently the concept of multi-agent DRL is being explored by researchers, particularly for complex environments. Accordingly, while in [25], the authors proposed a multi-agent DRL-based approach for a multi-tier framework, the authors in [26] proposed a multi-agent DRL approach to provide an optimal task offloading solution and minimize the task processing delay. Further, the authors in [27] proposed a multi-agent DRL-based approach for device-to-device communication, and likewise in [5], the authors formulated a multi-agent reinforcement learning-based algorithm for optimal offloading by using the improved Kuhn-Munkres (KM) algorithm for task scheduling. Although several DRL-based algorithms for MAEC and RIS-aided networks have been studied, researchers have not yet used multi-agent DRL-based algorithms in the domains of MAEC and RIS systems.

Accordingly, in this article, we develop a priority-aware resource allocation and task offloading strategy based on a multi-agent DRL algorithm that jointly maximizes the utility of a RIS-aided IoV network and improves the quality of communication between vehicles and the BS. We compare the proposed algorithm with other baseline DRL algorithms, namely soft actor-critic (SAC), deep deterministic policy gradient (DDPG) and twin delayed DDPG (TD3). The primary contributions are summarized below.

- We consider a RIS-aided IoV network involving a MAEC framework that involves vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P), and vehicle-to-infrastructure (V2I) computation offloading. The network consists of vehicles on the move, parked, and pedestrians carrying computation resources to assist task vehicles.

- We formulate the utility functions based on the state of the criticality of the network and the priority and size of tasks and then develop an analytical framework for the RIS-assisted communication network comprising of uplink and downlink.
- We propose an intelligent task offloading methodology for the IoV network to optimize the resource allocation scheme. In particular, we develop a novel MA-DRL solution using the Markov game (MG) for optimizing the task offloading decision strategy. The proposed model maximizes the mean utility of the IoV network and improves the communication quality between vehicles and BSs.

Extensive numerical simulations were performed to evaluate the performance of the developed solution and the impact of key network parameters. Compared to other baseline DRL algorithms, namely soft actor-critic (SAC), DDPG, and TD3 the proposed MA-DRL algorithm achieves higher utility. The proposed method also improves the offloading rate of the tasks as well as ensures that a higher percentage of offloaded tasks are completed compared to that of other DRL-based frameworks. Further, the results also verified the usefulness of deploying RISs in an IoV network towards improving the ratio of completed tasks, average offloading data rate, and reducing the mean delay in the network.

Structure of the paper: The flow of the paper is organized as follows. Section II provides a detailed explanation of the considered system model along with the network architecture, communication model, and task model and formulates the utility functions. In Section III, the proposed multi-agent DRL algorithm modeled by markov game is discussed. The simulation setup is discussed in Section IV and the simulation results are discussed in Section V. Finally, the conclusions are drawn in Section VI.

II. System Model

Table I shows the list of variables used in this paper.

A. Network Architecture

We consider a RIS-aided IoV network that involves vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P) and vehicle-to-infrastructure (V2I) computation offloading using vehicular fog computing (VFC) as illustrated in Fig. 1. The V2P scenario includes pedestrians on the roadside with resources within close proximity of the task vehicles as service providers and the V2I scenario allows the task vehicles to communicate with base stations (BSs) equipped with MAEC servers that aid in the computation of tasks. Both RIS-assisted as well as direct links are considered. We assume that there are Q=$\{BS_1, BS_2, \ldots, BS_q, \ldots, BS_Q\}$ BSs such that BS_q has M RISs within its coverage area denoted as $RIS_1, RIS_2, \ldots, RIS_m, \ldots, RIS_M$. The M RISs are attached to a central controller which dynamically tunes the best-reflected signal towards the intended node. It is to be noted that a RIS can be within the range of several BSs. With regards to the VFC scenarios, the V2V scenario consists of multiple task vehicles (V_t), service vehicles (V_s), and pedestrians with resources. The mobility of task vehicles are bi-directional while the service vehicles can either be on the move (bi-directional) or parked. BSs are deployed which provide uniform coverage throughout the system during the period of consideration. All service vehicles within the communication range of a task vehicle qualify to provide computation service. Moreover, each task vehicle can be simultaneously involved with multiple service vehicles and likewise, each service vehicle can aid multiple task vehicles at the same time.

We consider that the network has T time periods, where every period is divided into multiple time frames. A free-flow traffic model is considered where all the vehicles are within the range of a BS and the velocity of the vehicles remains constant in one time frame but may differ over different time frames. Accordingly, we assume that during period t, V_t has K resource providers (V_s or pedestrian) within its scope such that K=$\{V_1, V_2, \ldots, V_k, \ldots, V_K\}$. The bandwidth is split into several orthogonal spectrum bands where the available frequency range is divided into multiple non-overlapping channels, and each channel is used for a specific transmission. The vehicles and pedestrian exchange information with the BS and offloads tasks using these channels. However, in the case of large-sized tasks, we consider that only one task can be offloaded at one-time frame. We assume that each vehicle or pedestrian entering the coverage area of a base station (BS) transmits a vector message packet containing information about its position, computation capacity, speed, available resources, etc. The task vehicles also notify the BS if it has any offload requests at the beginning of each time slot. Next, the BS sends back a message packet to the task vehicle which contains information such as the unique id of the most eligible V_s where the task is to be offloaded and simultaneously inform the V_s regarding the same. The task vehicles receive this message packet via downlink communication and finally offloads the task to the allocated resource unit via uplink transmission. We consider the message packet containing the information of the

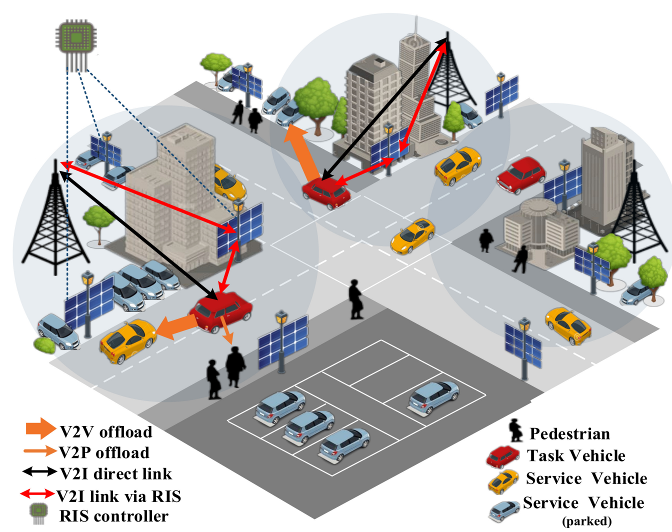


Fig. 1: An illustration of the considered RIS-aided IoV network for the VFC framework.

TABLE I: List of variables

Variable	Description	Variable	Description
V_t	task vehicle	$\mathbf{\Theta}_m$	phase shift matrix of the m^{th} RIS
V_s	service vehicle	$h_{k,m}^{LoS}, h_{m,b}^{LoS}, h_{b,m}^{LoS}, h_{m,k}^{LoS}$	line of sight components
t	time period	$h_{k,m}^{nLoS}, h_{m,b}^{nLoS}, h_{b,m}^{nLoS}, h_{m,k}^{nLoS}$	non-line of sight components
K	resource provider	ξ	rician factor
Z	number of tasks	Y_b	received signal at b^{th} BS
ρ_z	priority of task ϕ_z	P_v	transmit power of vehicles
D_z	data size of task ϕ_z	n_b	Gaussian noise at BS
C_z	computation size of task ϕ_z	$O_{k,t} \in \{0,1\}$	binary, indicating if communication is aided by RIS or not
τ_z	maximum tolerable delay task ϕ_z	$w_{k,t} \in \{0,1\}$	binary, indicating if a vehicle is offloading tasks or not
$h_{k,b}$	channel gain from k^{th} vehicle to BS	P_b	power of transmitter at the BS
$h_{b,k}$	channel gain from BS to k^{th} vehicle	$\lambda_{V_{s,t}}$	distance between V_t and V_s
l	pathloss	$h_{V_{s,t}}$	channel gain from V_t to V_s
Δ	path-loss exponent	$r_{V_{s,t}}$	transmission between V_t to V_s
$\lambda_{k,b}$	distance between the k^{th} vehicle and b^{th} BS	$r_{k,t}$	transmission between V_t to BS
$\lambda_{k,m}$	distance between the k^{th} vehicle and m^{th} RIS	b_{w_1}, b_{w_2}	allocated bandwidths
$\lambda_{m,b}$	distance between the m^{th} RIS and b^{th} BS	Υ_h, Υ_c, Υ_l	high, common, low priority classes
$\widetilde{h}_{k,b}, \widetilde{h}_{b,k}$	random scattering components	S_s, S_m, S_l	small, medium, large size categories
N_x, N_y	number of RIS elements along the x-axis and y-axis	α,β	network criticality parameters
t_z	task completion time	α,β	network criticality parameters
τ_z	maximum tolerable delay	act_t^i	actor module
$-\mathcal{C}_{(\Upsilon_h)}$	task failure penalty	f_{t-1}	previous state information vector data packet
η_p	number of small-sized high priority tasks	a_t^i, s_t^i, θ_i	action, state and parameters at t
$\mathcal{C}_{(\Upsilon_c)}$	positive constant for successful task	in_t^i	environment information packet
η_c	number of medium and large high priority tasks and all common tasks	r_t^i	reward at t
$\mathcal{C}_{(\Upsilon_l)}$	positive utility constant for low priority tasks	$\mathcal{U}_t^k$	overall utility
η_l	number of low priority tasks or sub-tasks	Ω_t^k	binary, task is offloaded or not
$\mathcal{G}$	set of g agents	$\mathcal{D}$	replay buffer
χ_z	energy required for task ϕ_z	ϱ	discount factor
η_{k_s}	number of local high priority tasks	$p^{\varkappa}$	distribution of s_t
ϕ_{τ_s}	delay in computation of tasks	ω	cumulative observation of all agents
ν_i	set of continuous policies with parameters θ_i	a_t^g	action of g^{th} agent at time t
ν'	set of target policies with parameter θ_i'	ϕ_{τ_s}	delay in computation of tasks

allocated resource unit to be constant in size for all cases. The eligible V_s is determined based on the properties of the task that needs to be offloaded such as maximum tolerable delay of the task, expected contact time between V_t and V_s which can be calculated from the relative velocity and distance between the two vehicles, expected delay and computation resource availability in the V_s. For the purpose of this study, we ignore the transmission delay involved in offloading service and assignment of V_s. Furthermore, we assume that V_t has Z tasks in a time period T denoted by Z={ϕ_1,ϕ_2,...,ϕ_z,...,ϕ_Z} where task ϕ_z is characterized by data size (D_z), computation size or number of CPU cycles needed (C_z), delay constraint (τ_z) and priority of task (ρ_z).

B. Communication Model

The channel between k^{th} vehicle and BS is denoted by $h_0 \in \mathbb{C}^{1\times 1}$ while the channel gain of the vehicle follows Rayleigh fading, respectively given as

$$h_{k,b} = \sqrt{l\lambda_{k,b}^{-\Delta}}\,\widetilde{h}_{k,b}\,, \tag{1}$$

$$h_{b,k} = \sqrt{l\lambda_{k,b}^{-\Delta}}\,\widetilde{h}_{b,k}\,, \tag{2}$$

where l denotes path-loss and Δ denotes the path-loss exponent. $\lambda_{k,b}$ denotes the distance between the k^{th} vehicle and BS and $\widetilde{h}_{k,b}$ and $\widetilde{h}_{b,k}$ are random scattering components following Gaussian distribution [9], [28]. In order to avoid penetration loss due to obstacles such as buildings, RISs with N=$N_x \times N_y$ elements are deployed on the roadside units to reflect the signal and control the propagation paths. Here N_x and N_y are the number of RIS elements along the x-axis and y-axis of the metasurface panel, respectively [13]. Accordingly, the phase shift matrix of the m^{th} RIS can be expressed as $\mathbf{\Theta}_m = diag[e^{j\psi_{1,m}}, e^{j\psi_{2,m}}, \ldots, e^{j\psi_{N,m}}]$. The channel vector between k^{th} task vehicle and RIS is denoted by $\mathbf{h}_{k,m} \in \mathbb{C}^{1\times N}$ and between RIS and BS is denoted by $\mathbf{h}_{m,b} \in \mathbb{C}^{N\times 1}$. Thus the channel gains from vehicle to RIS and RIS to BS are modeled using Rician distribution and are given by the following equations, respectively

$$\mathbf{h}_{k,m} = \sqrt{l\lambda_{k,m}^{-\Delta}}\Big(\sqrt{\frac{\xi}{1+\xi}}\mathbf{h}_{k,m}^{LoS} + \sqrt{\frac{1}{1+\xi}}\mathbf{h}_{k,m}^{nLoS}\Big)\,, \tag{3}$$

$$\mathbf{h}_{m,b} = \sqrt{l\lambda_{m,b}^{-\Delta}}\Big(\sqrt{\frac{\xi}{1+\xi}}\mathbf{h}_{m,b}^{LoS} + \sqrt{\frac{1}{1+\xi}}\mathbf{h}_{m,b}^{nLoS}\Big)\,. \tag{4}$$

Here, equation (3) and equation (4) are the channel gains from the k^{th} vehicle to m^{th} RIS and from m^{th} RIS to b^{th} BS, respectively. $\lambda_{k,m}$ and $\lambda_{m,b}$ denote the distance from the k^{th} vehicle to m^{th} RIS and from m^{th} RIS to b^{th} BS, respectively, while ξ represents the Rician factor. $\mathbf{h}_{k,m}^{LoS}$ and $\mathbf{h}_{m,b}^{LoS}$ are the line of sight (LoS) components and $\mathbf{h}_{k,m}^{nLoS}$ and $\mathbf{h}_{m,b}^{nLoS}$ are the non-line of sight (nLoS) components [29]. Similarly, the channel gains from b^{th} BS to m^{th} RIS and from m^{th} RIS to the k^{th} vehicle are respectively given as

$$\mathbf{h}_{b,m} = \sqrt{l\lambda_{m,b}^{-\Delta}}\Big(\sqrt{\frac{\xi}{1+\xi}}\mathbf{h}_{b,m}^{LoS} + \sqrt{\frac{1}{1+\xi}}\mathbf{h}_{b,m}^{nLoS}\Big), \quad (5)$$

$$\mathbf{h}_{m,k} = \sqrt{l\lambda_{k,m}^{-\Delta}}\Big(\sqrt{\frac{\xi}{1+\xi}}\mathbf{h}_{m,k}^{LoS} + \sqrt{\frac{1}{1+\xi}}\mathbf{h}_{m,k}^{nLoS}\Big). \quad (6)$$

Thus, the received signal at b^{th} BS can be expressed as

$$Y_b = \sqrt{P_v}\left(\sum_{k=1}^{K}[h_{k,b}q_k + \sum_{m=1}^{M}\mathbf{h}_{k,b}^{H}\mathbf{\Theta}_m\mathbf{h}_{m,b}q_k]\right) + n_b\,, \quad (7)$$

where P_v denotes the transmit power for communications of all vehicles, $\mathbb{E}[q_k^2] = 1$ is the transmitted signals and n_b is the additive white Gaussian noise at BS. The channel gain from k_{th} vehicle to BS can be given as $|O_{k,t}\mathbf{h}_{m,b}^{H}\mathbf{\Theta}_m\mathbf{h}_{k,m} + h_{k,b}|^2$, where $O_{k,t} \in \{0,1\}$ denotes a binary value such that if the communication is aided by RIS, the value will be $O_{k,t} = 1$. On the contrary, if the communication is not aided by RIS, $O_{k,t} = 0$. Thus, if the communication is not aided by RIS, the equation of SINR from vehicle to base station will only consider the direct channel between vehicle to BS denoted by $h_{k,b}$. $\mathbf{h}_{m,b}^{H}\mathbf{\Theta}_m\mathbf{h}_{k,m}$ is the cascaded channel and $h_{k,b}$ is the direct channel. Thus the uplink signal to interference plus noise ratio (SINR)from vehicle to BS at time t can be expressed as

$$\gamma_{k,t} = \frac{w_{k,t}P_v|O_{k,t}\sum_{m=1}^{M}\mathbf{h}_{m,b}^{H}\mathbf{\Theta}_m\mathbf{h}_{k,m} + h_{k,b}|^2}{\sum_{i=1,i\neq k}^{Y} w_{i,t}P_v|O_{i,t}\sum_{m=1}^{M}\mathbf{h}_{m,b}^{H}\mathbf{\Theta}_m\mathbf{h}_{i,m} + h_{i,b}|^2 + \sigma^2} \quad (8)$$

Here, $w_{k,t} \in \{0,1\}$ indicates if a vehicle is offloading tasks or not, if the vehicle is not offloading task then $w_{k,t} = 0$, otherwise $w_{k,t} = 1$. P_v is the transmit power for communications of the vehicle, σ^2 is the additive Gaussian noise and $\sum_{i=1,i\neq k}^{Y} w_{i,t}P_v|O_{i,t}\sum_{m=1}^{M}\mathbf{h}_{m,b}^{H}\mathbf{\Theta}_m\mathbf{h}_{i,m} + h_{i,b}|^2$ symbolizes the aggregate interference where Y is the number of vehicles using a sub-channel. Similarly, the channel gain from BS to k^{th} user can be given as $|O_{k,t}\mathbf{h}_{r,k}^{H}\mathbf{h}\mathbf{\Theta}_m\mathbf{h}_{b,m} + h_{b,k}|^2$. Thereby, the SINR from BS to vehicles can be expressed as

$$\gamma_{b,t} = \frac{P_b|O_{k,t}\sum_{m=1}^{M}\mathbf{h}_{m,k}^{H}\mathbf{\Theta}_m\mathbf{h}_{b,m} + h_{b,k}|^2}{\sum_{i=1,i\neq k}^{Y} P_b|O_{i,t}\sum_{m=1}^{M}\mathbf{h}_{m,i}^{H}\mathbf{\Theta}_m\mathbf{h}_{b,m} + h_{b,i}|^2 + \sigma^2}\,, \quad (9)$$

where P_b is the power of the transmitter at the BS. Alternatively, the SINR for V2V offloading can be expressed as

$$\gamma_{V_{s,t}} = \frac{P_v\lambda_{V_{s,t}}^{-\Delta}|h_{V_{s,t}}|^2}{\sum_{i\in K,i\neq k} P_v\lambda_{V_{i,r}}^{-\Delta}|h_{V_{i,r}}|^2 + \sigma^2}\,, \quad (10)$$

where $\lambda_{V_{s,t}}$ is the distance between V_t and V_s, $h_{V_{s,t}}$ denotes the desired channel gain and $\sum_{i\in K,i\neq k} P_v\lambda_{V_{i,r}}^{-\Delta}|h_{V_{i,r}}|^2$ represents the aggregate interference [3]. We assume that the wireless channel remains constant when a task is being executed. Accordingly, the rate of transmission between V_t to V_s and V_t to BS can be respectively given as

$$r_{V_{s,t}} = b_{w_1}\log_2(1+\gamma_{V_{s,t}})\,, \quad (11)$$

$$r_{k,t} = b_{w_2}\log_2(1+\gamma_{k,t})\,, \quad (12)$$

where b_{w_1} and b_{w_2} denote the allocated bandwidths for the two scenarios.

C. Task Model

In general, tasks from the vehicles can be classified on the basis of three main factors: task priority, size, and criticality of the network which can further be categorized into three hierarchical priority classes namely high priority (Υ_h), common (Υ_c) and low priority tasks (Υ_l). Υ_h tasks are highly delay-intolerant compared to that of Υ_c and generally include fundamental tasks such as navigation, security, sensing, etc. On the contrary, Υ_l tasks are delay-insensitive and generally involve tasks such as entertainment services, etc. Furthermore, the computational tasks can also be categorized as small (S_s), medium (S_m), and large (S_l) sized tasks based on the computation bits involved. Network conditions such as congestion, collision, lower throughput, etc., affect the quality of the network. Hence in order to improve the quality, we consider the criticality condition of the network as one of the main parameters. For measuring the criticality we introduce a pair of tunable parameters [α,β] such that $\alpha = 1-\beta$, $0 < \alpha, \beta < 1$. If $\alpha > 0, \alpha < \beta$ the network is less critical, while in case of $\beta > 0, \alpha > \beta$, the network is critical. Hence, α is directly proportional to the state of network criticality. The value of α or β can never be zero because, in a real-time atmosphere, a network can neither be too critical that all operations fail nor can it be unrealistically smooth with zero adverse impact from the environment [3]. In the proposed model, tasks are executed in three ways: local execution, V2V offloading, and V2I offloading. The first-hand tasks of a vehicle are generally small in size. Hence, we consider the high-priority small tasks ($\Upsilon_h + S_s$) as crucial tasks (e.g., sensing, navigation, etc.), which need continuous computation for the smooth operation of the vehicle. Hence these tasks are executed locally by the in-vehicle processor. Conversely, Υ_l tasks are delay tolerant, and $\Upsilon_l + S_l$ needs more processing power. Therefore, such tasks are offloaded directly to the BS equipped with a MAEC server so as to conserve the processing capability of the vehicles for future tasks of the V_ts. If there are no V_s within the range of a V_t, then all the tasks except the crucial ones are directly offloaded to the BS. If the size of tasks is larger compared to available computation resources, tasks are divided into multiple sub-tasks and offloaded to different locations.

D. Utility

High-priority tasks are highly delay-sensitive and it is essential for these tasks to complete within time. Thus, the utility function for small sized high priority tasks are designed to gain a positive value upon completion of tasks within the deadline, i.e. task completion time (t_z) < maximum tolerable delay (τ_z). This positive utility gained will be higher if the task is completed sooner. The utility gain will logarithmically decrease with increasing t_z. However, if the task is not completed in time and the task completion time exceeds the maximum tolerable delay, the utility will gain a negative constant as a penalty for task failure. Thus the utility for high-priority tasks of small size can be calculated as

$$\mathcal{U}_n^{\Upsilon_h} = \begin{cases} \alpha * log\left(1+\tau_z - t_z\right)/\eta_p^{\beta}, & t_z \leq \tau_z, \\ -\mathcal{C}_{(\Upsilon_h)}, & t_z > \tau_z\,, \end{cases} \tag{13}$$

where η_p denotes the number of high-priority and small-sized tasks and $-\mathcal{C}_{(\Upsilon_h)}$ is the negative penalty.

Similarly, the second utility function calculates the utility of the medium and large-sized high-priority tasks and all common tasks. In this case, if the task completion time is less than the maximum tolerable delay of the task then the utility will gain a positive score denoted by $\mathcal{C}_{(\Upsilon_c)}$. However, if the task completion time exceeds the maximum tolerable delay, the utility of the tasks decreases exponentially with the increasing time beyond the maximum tolerable delay. Thus, the utility for medium and large sized high priority tasks and all common tasks can be calculated as

$$\mathcal{U}_n^{\Upsilon_c} = \begin{cases} \alpha * \mathcal{C}_{(\Upsilon_c)}/\eta_c^{\beta}, & t_z \leq \tau_z, \\ \mathcal{C}_{(\Upsilon_c)} e^{-u(t_z-\tau_z)}, & t_z > \tau_z\,, \end{cases} \tag{14}$$

where η_c is the number of medium and large sized high priority tasks and all common tasks and $u > 0$ is a constant.

Since Υ_l tasks are delay tolerant, they are offloaded directly to the BS. However, these tasks may still lose their value due to communication failure or other technical errors resulting in $t_z = \infty$, subsequently making the utility zero. Accordingly, the utility of Υ_l tasks can be defined as

$$\mathcal{U}_n^{\Upsilon_l} = \begin{cases} \alpha * \mathcal{C}_{(\Upsilon_l)}/\eta_l^{\beta}, & t_z \leq \tau_z, \\ 0, & t_z > \tau_z\,, \end{cases} \tag{15}$$

where $\mathcal{C}_{(\Upsilon_l)}$ is the positive utility constant for Υ_l tasks and η_l is the number of Υ_l tasks or sub-tasks. If there are no V_s within the range of a V_t at time t, or the available resources in the V_s are insufficient, the tasks are directly offloaded to the BS. Nevertheless, the utility of the task is calculated using the respective utility function according to the task priority. Therefore, the final utility of the network is calculated as

$$\mathcal{U}_n = \mathbb{1}(\Gamma_{(\Upsilon_h)})U_n^{\Upsilon_h} + \mathbb{1}(\Gamma_{(\Upsilon_c)})U_n^{\Upsilon_c} + \mathbb{1}(\Gamma_{(\Upsilon_l)})U_n^{\Upsilon_l} - \chi_z C_z\,, \tag{16}$$

where $\Gamma_{(\Upsilon_h)}$, $\Gamma_{(\Upsilon_c)}$, $\Gamma_{(\Upsilon_l)}$ are priority constants, $\mathbb{1}$ is the indicator function and χ_z is the energy required for task ϕ_z in V_s. Let the service vehicle V_k have η_{k_s} local high-priority tasks and ϕ_{local} is the ratio of the minimum required computation frequency for execution of local tasks to the total available frequency. Then, the utility of local task is given by

$$\mathcal{U}_{local}(\phi) = \sum\nolimits_{s=1}^{\eta_{k_s}} log\left(1+\tau_s - \phi_{\tau_s}\right), \phi \in [\phi_{local}, 1]\,, \tag{17}$$

where ϕ_{τ_s} is the delay in computation of tasks. When a vehicle executes an offloaded task, part of its frequency is assigned to that task. As a result, it changes the utility of its local task from $\mathcal{U}_{local}(\phi_k)$ to $\mathcal{U}_{local}(\phi_k')$ such that $\chi_z C_z = \mathcal{U}_{local}(\phi_k) - \mathcal{U}_{local}(\phi_k')$. Additionally, if more tasks are offloaded to the vehicle then $\mathcal{U}_{local}(\phi_k')$ is updated to $\mathcal{U}_{local}(\phi_k'')$, where ϕ_k'' is the consumed energy for the new task [3], [4].

III. Multi-agent Deep Reinforcement Learning

We assume that at a certain time period, there are multiple V_t within the coverage area of a BS and each of them makes offloading decision based on the dynamic VFC environment statistics. Since the VFC environment is dynamic, these statistics change in every time slot which makes the future states of the environment unpredictable. In the considered framework, we consider an actor-critic-based MA-DRL algorithm where g agents interact with the environment to make an optimal decision for achieving the predefined objective. We aim to maximize the utility of this framework by using multiple agents associated with the BS such that each agent corresponds to a different V_t or a different set of V_t. The dynamics of the state and reward get updated when all the agents mutually join their actions. MA-DRL approach can reduce the instability in a multi-vehicular environment and can be modeled as a Markov game (MG), where a set of agents work towards optimizing a common reward. The details of the architecture of the MA-DRL for the considered framework including components and algorithm are described below.

A. Markov Game and vehicle clustering

In MG, each game can be represented as a matrix, where joint actions can be calculated from the matrices [30], [31]. In general, MG can be represented by the tuple ($\mathcal{G}, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \varrho$), where

- $\mathcal{G}$ is the set of g agents;
- $\mathcal{S}$ denotes a finite state space;
- $\mathcal{A} = A_1 \times A_2 \times A_3 \times \ldots \times A_g$ denotes the joint action space;
- $\mathcal{T} : S \times A \times S \rightarrow [0,1]$ denotes the transition function;
- $\mathcal{R}$ such that $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \rightarrow Reward$;
- ϱ denotes the discount factor.

In the proposed architecture we consider the interaction among the agents to be simultaneous and cooperative. Let $n(V_t)$ and $n(\text{agent})$ represent the number of V_ts and number of agents, respectively. Accordingly, if $n(V_t) \leq n(\text{agent})$, one agent will correspond to one V_t, whereas if $n(V_t) > n(\text{agent})$, V_t with similar parameters will correspond to the same agent. Such V_ts with similar parameters are clustered for the same agent using k-means algorithm considering relative parameters such as speed, location, direction, etc., as measures for feature scaling [5], [26], [31]. **Algorithm 1** illustrates the k-means algorithm used for vehicle clustering.

B. MA-DRL Components

The MA-DRL is modeled as MG as illustrated in Fig. 2. The algorithm consists of four major components as described below.

Algorithm 1 k-means algorithm for vehicle clustering

1: Initialize g agents.
2: Initialize k vehicles.
3: Shuffle vehicles and randomly select g vehicles as centroids.
4: Repeat until no change in centroids.
5: Take parameters as features for feature scaling.
6: Compute the sum of squared distance or similarity among the values of the features among all vehicles and all centroids.
7: Assign each vehicle with the closest centroid vehicle, thus forming a cluster with the centroid vehicle.
8: Assign one agent to each cluster.

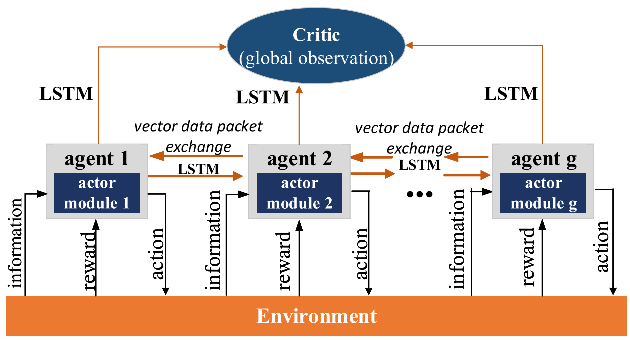


Fig. 2: Block diagram of the MA-DRL modeled as MG.

- *Actor*: Each agent has its individual and independent actor module(s) (act). The module act_t^i accepts information packet in_t^i from the environment at time t. Each module shares the information about its previous states with other actor modules by converting the information as a vector data packet. At time t, an actor module will share the information vector data packet of its previous state at time $t-1$, denoted as f_{t-1}. Every agent chooses its action based on the actor such that the action (a_t^i) corresponds to the mapping function $\mu_i(s_t;\theta_i)$ where s_t and θ_i are the approximate overall state and parameters at time t, respectively and $s_t \approx \{f_{t-1}, in_t^i\}$. An agent gets corresponded to a vehicle when it is in an idle state at the beginning of a time period and does not consider any vehicle whose service is in use or being computed. Each agent follows their own strategy $\mu_i(s_t)$ and for each action a_t^i, it receives a reward $r_t^i = r(s_t, a_t^i)$ from the environment following which the state gets updated to s_{t+1}.
- *Critic*: Multi-agent DRL consists of one global critic, which observes the actions of each actor and evaluates the performance of all the actors centrally. The global critic value function can be given by $\mathrm{Q}(s_t, a_t^1, \ldots, a_t^g)$.
- *Information exchange*: The communication is based on the artificial neural network-based long short-term (LSTM) algorithm [32]. The LSTM algorithm converts the relevant information about observations, states, and actions of all agents and converts it into a vector data packet that is shared among the agents simultaneously. This packet is the previous state information data packet f_{t-1}. Each actor of every agent receives this packet at the beginning of a state and updates the packet by adding their own observed information and actions in the current state. This data packet then gets updated from f_{t-1} to f_t, which contains the information of actor act_t and action a_t of all the agents at time t. Thus, the decision made by every agent is dependent upon the current and previous state of its own as well as other agents, which allow the agents to take global decisions [5].
- *Reward*: According to the observations, the agents take action a_t^i and receives reward r_t^i. Thus the overall reward is represented by

$$R(s_t, a_t) = \frac{1}{T}\sum\nolimits_{t=0}^{T-1}\sum\nolimits_{k=1}^{K}\Omega_t^k \mathcal{U}_t^k\,, \tag{18}$$

where $\Omega \in \{0,1\}$ signifies whether a task is offloaded or not.

The main experimental protocol for the MA-DRL algorithm includes the following steps:

1) **Environment definition:** The VFC environment is defined which includes the definitions of the state space, action space, and reward function.
2) **Initialization of the agents:** Next, the agents are defined to make decisions based on the environment. Each of the agents maps the state of the respective section of the network to actions.
3) **Agent training:** During the training process, each of the agents interacts with the environment and collects experiences, and stores them in the replay buffer. Consequently, the updates its policy based on these experiences and repeats the process for a fixed number of episodes or until convergence is reached.
4) **Agent evaluation:** After the training process, the performance of the agent is evaluated by analyzing its ability to optimize the reward or utility function.

C. MA-DRL Algorithm Design

During the training process, the actor and the critic networks are trained centrally, while the target is trained in a decentralized manner. The joint state, action, reward, and next state to replay buffer $\mathcal{D}$. The main critic network is updated for each agent the action-value function for states (s_t^i), actions (a_t^i) at the time (t) for the considered network can be represented by Bellman equation given as

$$Q_t^i(s_t^i, a_t^i) = \mathbb{E}[r_i^t + \varrho\mathbb{E}[Q_t^i(s_t^{\prime i}, a_t^{\prime i})]\,. \tag{19}$$

Here, the Q value is defined as

$$Q = \mathbb{E}[r_i + \varrho\mathbb{E}[Q_i(s', a_t^{\prime 1}, \ldots, a_t^{\prime g})]]\,, \tag{20}$$

where ϱ is the discount factor. Now, let the parameters for the policies and set of agent policies be given by $\theta_i = \{\theta_1, \ldots, \theta_g\}$ and $\pi_i = \{\pi_1, \ldots, \pi_g\}$, respectively. Then, for the i^{th} agent, the gradient can be written as

$$\nabla_{\theta_i} J(\theta_i) = \underset{(s_t \sim p^{\mathtt{x}}, a_t \sim \pi_i)}{\mathbb{E}}[\nabla_{\theta_i} \log \pi_i(a_i|s_i) Q_i^{\pi}(\omega, a_t^1, \ldots, a_t^g)]\,, \tag{21}$$

where $p^{\mathtt{x}}$ is the distribution of s_t, ω denotes the cumulative observation of all agents, and $Q_i^{\pi}(\omega, a_t^1, \ldots, a_t^g)$ is the central action-value function. The first vector is determined by calculating the weighted average of the policy gradient, while, second-moment vector is determined by calculating the weighted average of the squared gradient of the policy with respect to the parameters at each step. Thus the mean and variance are updated as $y_1 * \text{mean}_{t-1} + (1 - y_1) * (\nabla_{\theta_i} J(\theta_i))$ and $y_2 * \text{var}_{t-1} + (1 - y_2) * (\nabla_{\theta_i} J(\theta_i))^2$, respectively, where $y_1 = 0.9$ and $y_2 = 0.999$ denotes the decay rate. mean_t and var_t denotes the updated mean and variance at time t.

We consider a deterministic-based policy approach such that ν_i denotes the set of continuous policies with respect to the set of parameters θ_i. The tuple $(\omega, \omega', a_t^i, \dots, a_t^g, r_t^i, \dots, r_t^g)$ are stored in $\mathcal{D}$ as information of the g agents. Now, Q_i^ν is updated as

$$\mathcal{L}(\theta_i) = \mathop{\mathbb{E}}_{(f_t^i, \omega, a_t^i, r_t^i, \omega')} [(Q_i^\nu(\omega, a_t^i, \dots, a_t^g) - y)^2] , \qquad (22)$$

$$y = r_t^i + \varrho Q_i^{\nu'}(\omega', a_t'^i, \dots, a_t'^g) . \qquad (23)$$

Here, (22) calculates the loss of the critic network, (23) computes the target, and ν' denotes the set of target policies with parameter θ_i'. Thus, the gradient for continuous policy is updated as [33]

$$\nabla_{\nu_i} J(\theta_i) = \mathop{\mathbb{E}}_{(\omega, a_t \sim \mathcal{D})} [\nabla_{\theta_i} \nu_i(a_i|s_i) \nabla_{a_i} Q_i^\nu(\omega, a_t^i, \dots, a_t^g)]. \qquad (24)$$

The policy iteration continues until convergence. The above MA-DRL algorithm is summarized in Algorithm 2.

IV. Simulation Set-up

A two-way roadway set-up with multiple task and service vehicles in the range of a BS is accounted for and the BSs are deployed with a distribution of 1BS/km. The types of tasks within the network are classified based on priority and size while considering the criticality of the network. The simulation parameters are given in **Table II**.

For the selection of the most eligible service vehicle in V2V offloading, we propose a multi-agent DRL algorithm, called MA-DRL as illustrated in **Algorithm 2**. The stopping criteria of the proposed algorithm is based on the convergence of the network policy which is determined based on the average reward obtained per 100 episode i.e. when the average reward does not increase per 100 episode. Additionally, we evaluate the performance of the system with and without RIS as the transmission paths between the BS and vehicles can be direct as well as via RIS. The RISs are assumed to be installed on buildings or roadside units and are equipped with a central microcontroller.

The proposed MA-DRL algorithm is compared against three baseline single-agent DRL (SA-DRL) algorithms. The SA-DRL algorithms are modeled on markov decision process (MDP) framework. The MDP environment includes a set of probable states, actions, and rewards. In MDP, one agent interacts with the environment in a discrete time step and decides the action based on the current state information at each time frame. A brief description of the algorithms are summarized below.

Algorithm 2 MA-DRL algorithm

Input: current environment
Initialize parameters $\theta_i = \{\theta_1, \dots, \theta_g\}$ and δ for actor and critic network, respectively.
Initialize replay buffer $\mathcal{D}$.
Initialize main actor and critic networks with weights θ and δ, respectively.
Initialize target actor and critic networks θ' and δ'.
for all training steps **do**
  Initialize vector data packet f_0^0, $t = 0$.
  for each episode **do**
    Initialize a random process for action exploration.
    Retrieve initial state.
    while $t < T$ **do**
      for $i = 1 : g$ **do**
        Select action a_t^i for all agents at time t w.r.t. current policy.
        Execute actions.
        Receive reward r_t^i and update state s_{t+1}^i.
        Update vector data packet by f_{t-1}^i =LSTM$(f_{t-1}^{i-1}, [s_t^i, a_t^i])$
        Store transition $(f_t^i, \omega, a_t^i, r_t^i, \omega')$ in $\mathcal{D}$
      end for
      Updated $f_t^0 = m_{t-1}^g$
      $t = t + 1$
    end while
    for do agents i=1 to g
      Randomly sample a mini-batch of transition from $\mathcal{D}$.
      Update critic network using equation (22).
      Update actor using equation (24).
      Update LSTM.
      Soft-update target actor and target critic networks:
$$\theta_i' \leftarrow \kappa\theta_i + (1 - \kappa)\theta_i' ,$$
$$\delta_i' \leftarrow \kappa\theta_i + (1 - \kappa)\theta_i' .$$
    end for
  end for
end for
until convergence

A. Deep Deterministic Policy Gradient (DDPG)

DDPG is used for continuous action space which combines the concept of deterministic policy gradient (DPG) and deep Q-network (DQN). The algorithm uses off-policy data and learns the Q function using the Bellman equation. The loss function of the critic and the target is given respectively as

$$L(\theta^Q) = \mathop{\mathbb{E}}_{(s_t \sim \rho^{\mathbb{x}}, a_t \sim \mathbb{x}, R_t \sim \mathbb{x})} [(Q(s_t, a_t|\theta^Q) - \Psi_t)^2] , \qquad (25)$$

$$\Psi_t = R(s_t, a_t) + \gamma Q(s_{t+1}, \varrho(s_{t+1})|\theta^Q) , \qquad (26)$$

where θ^Q denotes the parameter values and $p^{\mathbb{x}}$ denotes the distribution of s_t. The policy updation can be formulated as

$$\nabla_{\theta^\varrho} J^\varrho \approx \mathop{\mathbb{E}}_{(s_t \sim \rho^{\mathbb{x}})} [\nabla_a Q(s, a|\theta^Q)|_{s_t, a=\varrho(s_t)} \nabla_{\theta^\varrho} \varrho(s|\theta^\varrho)|_{s_t}] ,$$

TABLE II: Simulation Parameters

Parameters	Values	Parameters	Values
$[\Upsilon_h, \Upsilon_c, \Upsilon_l]$	$[0.5, 1, 2]$	computation size	$[1, 3, 5]$
max. tolerable delay (Υ_h)	$[0.5, 1, 2, 4]$s	traffic	$[20 - 70]$ vehicles/km
max. tolerable delay (Υ_c, Υ_l)	$[5, 6, 7, 9]$s	max. relative distance	500m
data size	$[3, 5, 10]$MB	max. relative velocity	±60Km/hr
max. tasks/vehicle	55	batch size	256
max. local tasks/vehicle	7	vehicle computation power	$[5 - 12]$GHz
number of RIS	3	number of RIS elements	128GHz
LR_{actor}	0.001	ϱ	0.95
LR_{critic}	0.02	κ	0.01
HL_{actor}	$[512, 128]$	replay buffer	10^6
HL_{critic}	$[512, 256]$	random seeds	10
b_{w_1}	15MHz	b_{w_2}	20MHz
Optimizer	Adam	Adam optimizer tolerance	10^{-6}
actor fully connected layers	4	critic fully connected layers	3
activation function	Softmax	dataset samples	$60,000$
training sample	$42,000$	validating & testing samples	9000
dataset generation library	OpenAI Gym [34]	ML library	Tensorflow

$$\approx \frac{1}{H} \sum_t [\nabla_a Q(s, a|\theta^Q)|_{s_t, a=\varrho(s_t)} \nabla_{\theta^\varrho} \varrho(s|\theta^\varrho)|_{s_t}] \,, \tag{27}$$

where ∇ denotes the parameters of the policy and H denotes the batch of transition in the replay buffer. The policy iteration continues until convergence.

B. Twin Delayed DDPG (TD3)

TD3 is a successor to DDPG which overcomes the problem of overestimation of the Q function in DDPG. It was designed to counter drawbacks of other actor-critic type algorithms [35]–[37]. The additional features in TD3 are: it adds delay to policy updation and regularizes the noise. TD3 also uses two critic functions and hence adds two Q functions and both functions are chosen in succession to compute a single target. The Q-function giving minimum target value is used first. The Q-functions are computed by using mean square Bellman error minimization and the target is defined as

$$\Psi_t = R(s_t, a_t) + \gamma \min_{i=1,2} Q_{\theta_{c_i}, \theta_i'} \varrho(s_{t+1}) | \theta_i{}^Q) \,, \tag{28}$$

where $\theta_i{}^Q$ denotes the parameter values. The actions which form the Q-learning target policy are defined as $\mathcal{L}_{\theta_{targ}}$, where the added noise is clipped for noise regularisation. Thus the target action is formed as

$$a(s) = clip(\mathcal{L}_{\theta_{targ}}(s) + clip(\epsilon, c_1, c_2, a_{low}, a_{high})), \tag{29}$$

where $\epsilon \sim \mathcal{N}(0, \nabla)$ is the clipped noise, c_1 and c_2 are the two critics such that the valid actions lie within the range $a_{low} \le a \le a_{high}$. Each of the Q functions are computed using different loss functions defined as

$$\mathcal{L}(\theta_{c_1}) = \mathop{\mathbb{E}}_{(s_t \sim \rho^{\mathbb{x}}, a_t \sim \mathbb{x}, R_t \sim \mathbb{x})} (Q_{c_1}(s_t, a_t|\theta^Q) - \Psi_t)^2] \,. \tag{30}$$

$$\mathcal{L}(\theta_{c_2}) = \mathop{\mathbb{E}}_{(s_t \sim \rho^{\mathbb{x}}, a_t \sim \mathbb{x}, R_t \sim \mathbb{x})} [(Q_{c_2}(s_t, a_t|\theta^Q) - \Psi_t)^2] \,, \tag{31}$$

The policy updation of TD3 is similar to DDPG and the policy iteration continues until convergence.

C. Soft Actor-Critic (SAC)

SAC acts as a link between stochastic policy optimization and DDPG-style-based approaches. This algorithm works on the basis of entropy regularization which is formulated using Bellman equation. The policy of SAC is updated using Kullback-Leibler divergence [38] formulated as

$$\pi_n = \arg\min_{\pi' \in \Pi} I_{KL} \Big(\pi'(.|s_t) || \frac{\exp((1/\Delta) Q^\pi(s_t, .)}{Z^\pi(s_t)} \Big) \,, \tag{32}$$

where Π is set of policies and I_{KL} is the amount of information lost. (32) can be further minimized by parameter updation given as

$$J_\pi(\theta) = \mathop{\mathbb{E}}_{(s_t \sim \mathcal{D})} \Big[\mathop{\mathbb{E}}_{a_t \sim \pi_\theta} [\Delta \, log(\pi_\Theta(a_t|s_t)) - Q_\theta(s_t, a_t)] \Big] \,. \tag{33}$$

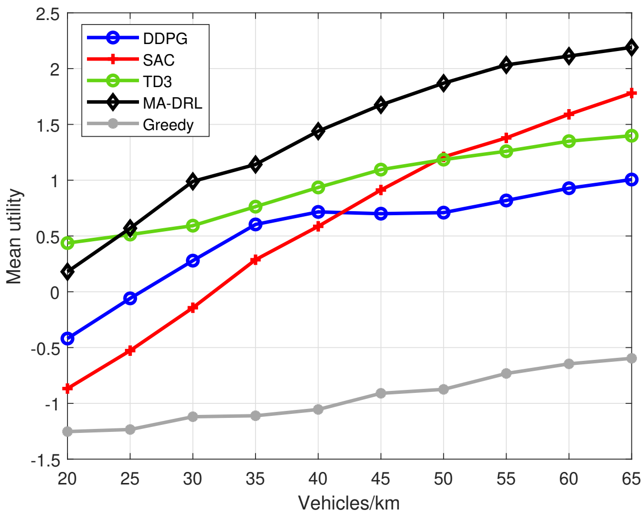
The policy iteration continues until an optimal value is reached.

V. Simulation Results

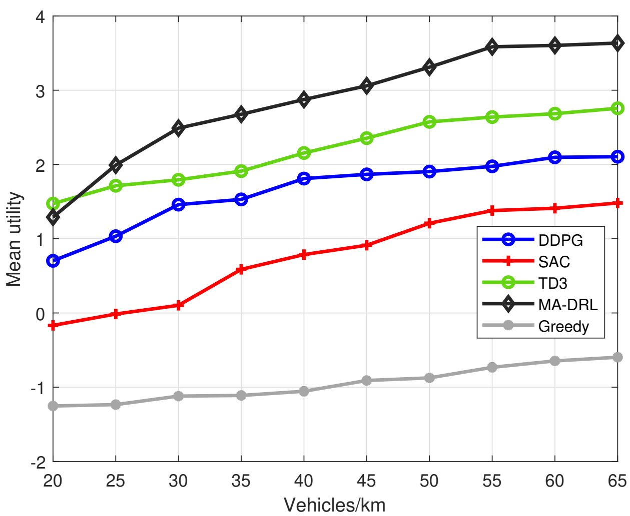
In this section, we simulate the considered VFC framework and numerically analyze the communication performance of the proposed algorithm with and without RIS. We compare the proposed MA-DRL with other benchmark DRL algorithms, namely SAC, DDPG, and TD3, and one baseline greedy algorithm, that chooses the most eligible service vehicle on the basis of maximum remaining computation power.

A. Mean Utility

In the considered VFC system, the speed of the task vehicles are set to $40 Km/hr$ with a maximum relative velocity between two vehicles being $60 Km/hr$. Fig. 3a and Fig. 3b show the mean utility of the network under moderate and critical network conditions, respectively. Moderately critical network is defined by setting the values of α and β as 0.3 and 0.7, respectively, and critical networks are defined by setting the α and β as 0.7 and 0.3, respectively. For both cases,

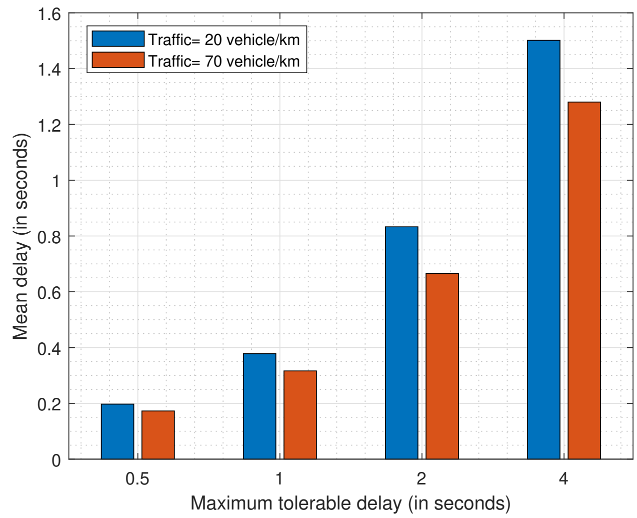


(a) Moderately critical network, [α, β]=[0.3,0.7].

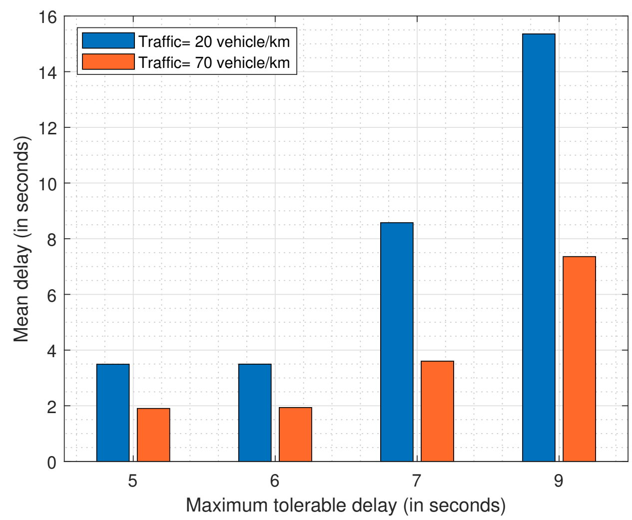


(b) Critical network, [α, β]=[0.7,0.3].

Fig. 3: Mean utility with respect to different traffic conditions.



(a) Mean delay for high priority tasks.



(b) Mean delay for common and low priority tasks.

Fig. 4: Mean delay in task execution for different traffic conditions.

the mean utility of the network for the proposed MA-DRL is compared with benchmark single-agent actor-critic type DRL algorithms such as TD3, DDPG, and SAC. Additionally, the DRL algorithms are also compared with a non-DRL algorithm called the Greedy algorithm which considers the available computation power of the vehicles as the primary parameter for selecting service vehicles. From the figures, it can be observed that in both cases the multi-agent approach is outperforming the single-agent approaches. But we can also see that when the traffic is very less, TD3 tends to have a slightly higher utility compared to MA-DRL. This justifies the fact that multi-agent algorithms perform better than single-agent environments in more complex networks. It is to be noted that in our simulation we have considered vehicles moving in four directions, i.e., we have considered two-way roads with intersections. Moreover, high traffic density and multiple-directional mobility make the network more complex. Further, in Fig. 3a, we can see that the pattern of TD3 and DDPG is almost similar during mid-traffic density and high-traffic density. However, the performance of SAC is seen to be improving from low to high making it an ideal alternative for high traffic but not suitable in low traffic. While in critical condition, both DDPG and TD3 outperform SAC throughout. However, in the overall scenario, if the network layout is much less complex, TD3 can be considered as an ideal alternative for MA-DRL.

B. Delay

In Fig. 4, we show the mean delay in task execution of the proposed MA-DRL algorithm for low traffic (20Km/hr) and high traffic (70Km/hr) conditions. In particular, Fig. 4a shows the mean delay for high-priority tasks considering the maximum tolerable delay ranges between $[0.5-4]$seconds since high-priority tasks are strict on delay tolerance. While Fig. 4b shows the mean delay for common and low-priority tasks where the maximum tolerable delay ranges between $[5-9]$seconds, common and low-priority tasks are comparatively more delay tolerant. For delay evaluation, we consider an

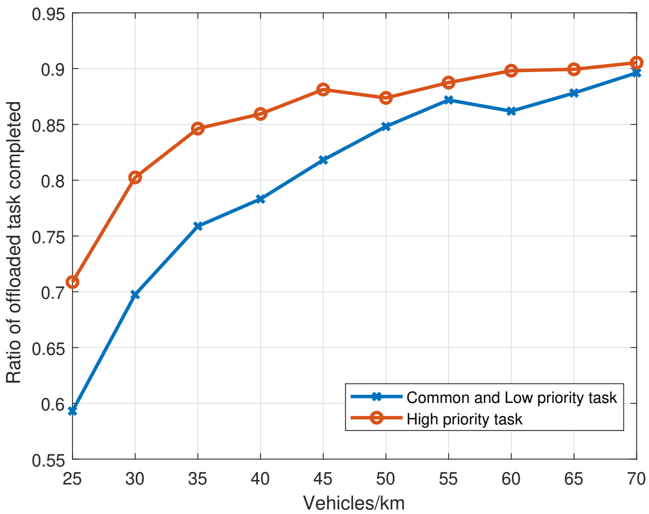


Fig. 5: Ratio of successfully offloaded tasks that are completed for various traffic conditions.

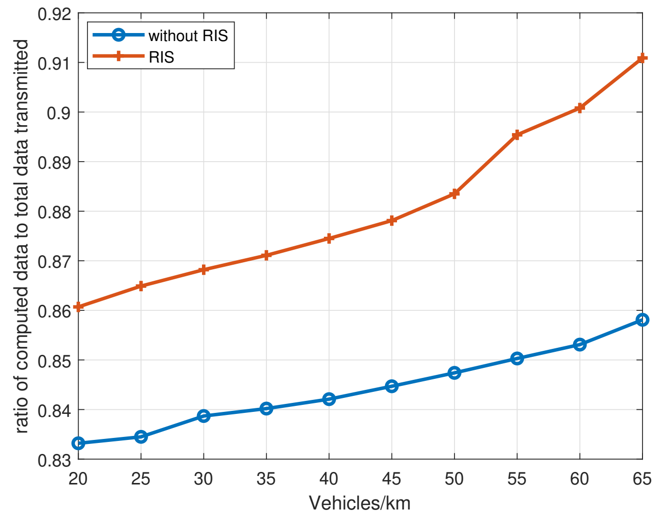


Fig. 6: Ratio of computed data to total data transmitted.

average critical network with the values of $[\alpha, \beta] = [0.5, 0.5]$. In both cases we consider the number of task vehicles to be the same, hence when traffic density is low, it implies that the number of service vehicles are less and when traffic density is high, the number of service vehicles are more. Thus the task vehicles do not have to queue for offloading to other service vehicles until a service vehicle completes the ongoing tasks and needs to offload comparatively fewer tasks to the BS during high traffic which reduces the mean delay. Comparing Fig. 4a and Fig. 4b we can see a significant difference between the mean delay in high-priority tasks compared to the mean delay in common and large tasks which justify that high-priority tasks are more delay sensitive.

C. Successful tasks

In Fig. 5, we compare the ratio of high-priority offloaded tasks successfully completed to that of common and low-priority offloaded tasks successfully completed. High-priority tasks are offloaded only to other service vehicles, however common and low-priority tasks are offloaded partly to vehicles and partly to the BS. Fig. 5 shows that the task completion rate of high-priority tasks is higher compared to common and low-priority tasks. This justifies the objective that high-priority tasks are crucial and that completion of these tasks are more important than other tasks for a vehicle to operate smoothly in an IoV network.

D. RIS assisted task offloading

1) Rate of data successfully computed: Fig. 6 shows the ratio of the amount of total data transmitted by the vehicles to the total data computed. Even though only a fraction of the total data is offloaded to the BS while the rest of the data are computed in the V2V infrastructure, the graphs show a significantly increased amount of data being computed at the BS when the network is being aided by RIS. Moreover, the part of task offloading within the V2V infrastructure is offloaded from one vehicle to another vehicle via a direct

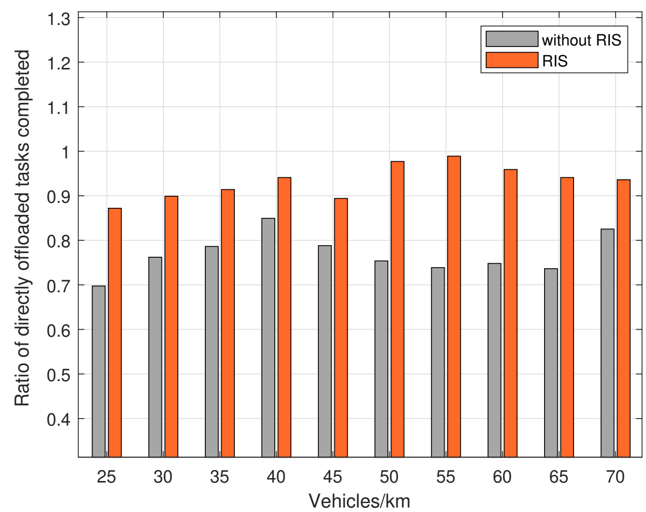


Fig. 7: Ratio of tasks offloaded to BS that are successfully completed.

link where no RIS is involved. Hence the improvement in performance seen in Fig:6 when the network is being aided by RIS is due to the bits involved in the tasks offloaded to the BS via RIS. Consequently, it implies that the incorporation of RIS improves communication and allows more data to be offloaded and computed at the BS.

2) Ratio of tasks offloaded to BS that are successfully completed: In Fig. 7 we show the ratio of tasks that are offloaded to the BS and are successfully completed when assisted by RIS versus the scenario without RIS. This figure shows that when the communication is being assisted by RIS, it improves the quality of communication and allows more data to be transmitted as justified by Fig.6. Thus, it reduces the amount of data lost in transmission compared to that of a communication that is not supported by RIS and reduces the rate of task failure. Hence, the rate of successfully completed tasks is relatively higher with RIS.

3) Average offloading rate of tasks to BS: In Fig. 8, we show the average offloading rate of tasks from vehicles to the BS with and without RIS. From the figure, it can be

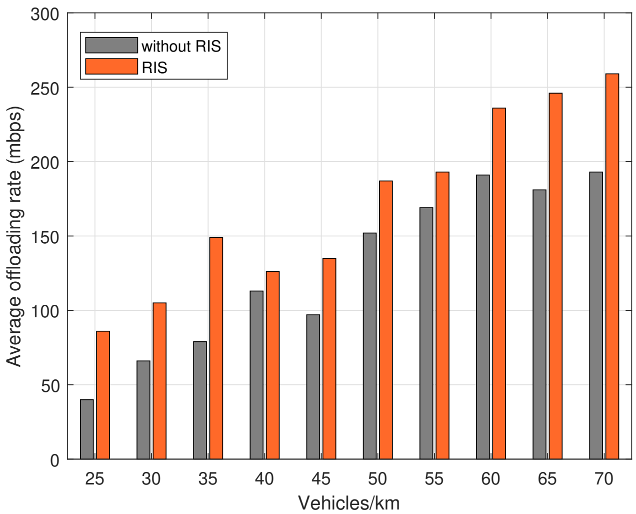


Fig. 8: Average offloading rate of tasks to the BS.

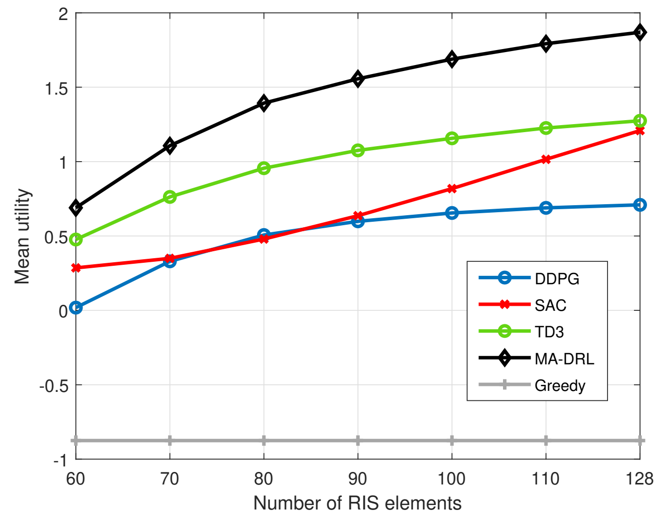


Fig. 10: Mean utility for different number of RIS elements.

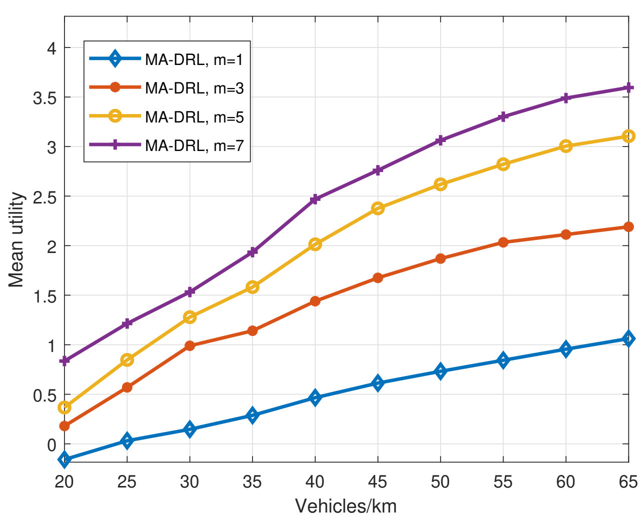


Fig. 9: Mean utility vs traffic for $m = [1-7]$ number of RIS.

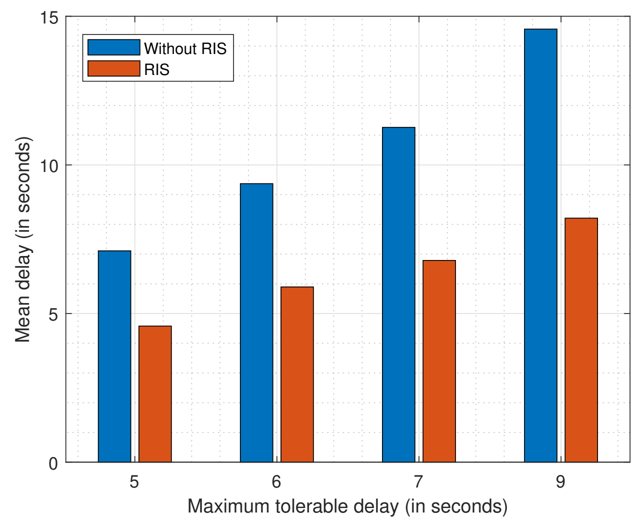


Fig. 11: Mean delay for the tasks offloaded to BS directly vs via RIS.

seen that with the aid of the RIS, the offloading rate is higher as compared to the scenario without the RIS. This is because the RIS mitigates signal attenuation and dispersion, thus improving the signal-to-noise ratio at the vehicles. Thus, during high traffic conditions or when there is a higher number of tasks to be offloaded to the BS, the aid of RIS can help with faster offload and thereby faster processing of data.

4) Mean utility for different number of RIS: Fig. 9 shows the mean utility for the proposed MA-DRL algorithm at different traffic conditions. We compare the mean utility of the framework for different number of RIS m. We can see that the mean utility is less when $m = 1$ and it significantly increases when the value of m increases. However, we can also observe that the difference between the mean utility graphs gradually decreases, i.e. the difference between $m = 1$ and $m = 3$ graph is more than the difference between $m = 3$ and $m = 5$. Additionally, the effect of number of RIS on the mean utility is more in higher traffic.

5) Mean utility vs number of RIS elements: Fig. 10 illustrates the performance comparison of the proposed MA-DRL algorithm with the other three DRL algorithms and the baseline greedy algorithm for different RIS elements. For the comparison in Fig. 10 we have considered $vehicles/km = 50$ and criticality parameters as $[\alpha, \beta] = [0.3, 0.7]$, i.e., the network is moderately critical. The number of RIS elements is within the range $[60-128]$. We can see that except for the greedy algorithm, the mean utility of all the other algorithms increases as the number of RIS elements increases. This is because increasing the number of RIS elements increases the achievable rate of the RIS-assisted channels which as a result improves the task offloading quality for all the algorithms.

6) Mean delay for the tasks offloaded to BS: In Fig. 11, we show the mean delay for only tasks which are offloaded to the BS when the maximum tolerable delay lies within the range [5-9]. Here, we compare the mean delay when with and without the presence of RIS. It can be seen that when the network is aided by RIS, the delay is significantly reduced.

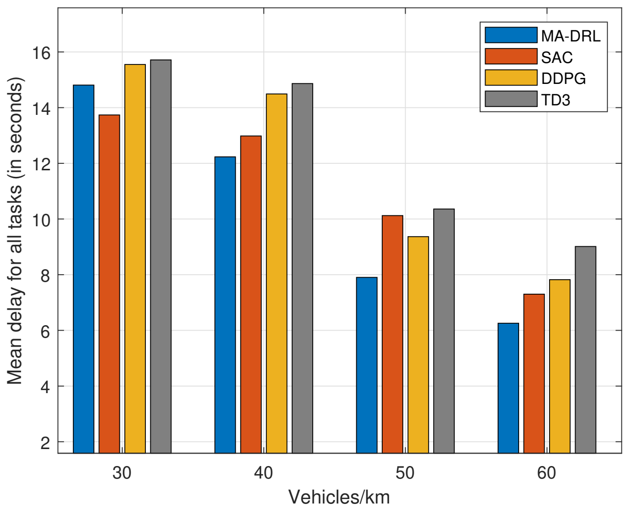


Fig. 12: Mean delay for all tasks combined.

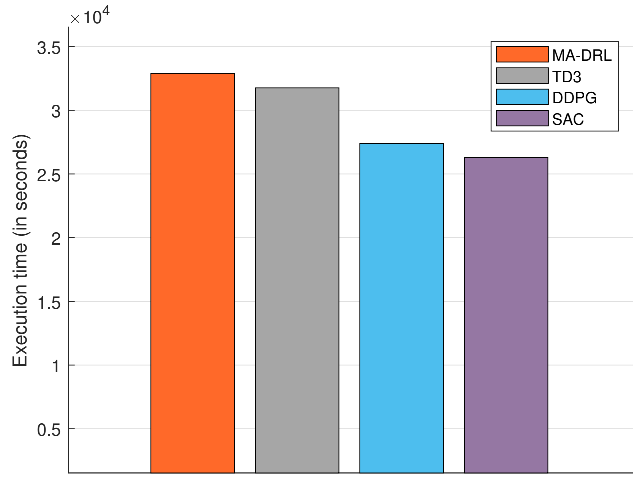


Fig. 13: Execution time for MA-DRL, TD3, DDPG and SAC.

E. Mean delay for all tasks

Fig. 12 shows the overall mean delay for Υ_h, Υ_c and Υ_l tasks cumulatively with traffic density of $30, 40, 50$ and 60 vehicles/km. The figure shows that when vehicle density is lower, the mean delay is higher irrespective of the algorithm. This observation can be justified by considering the fact that when vehicular density is low, task vehicles have fewer options for service providers. Therefore, tasks are either queued to the available service providers or offloaded to the BS. In either situation delay increases which eventually leads to an overall increase in mean delay. Similarly, when the density of vehicle is high, most of the tasks can be offloaded in V2V offloading mode which reduces the mean delay. Additionally, it can be observed that the delay in TD3 is significantly higher compared to other algorithms despite the fact that it performs better in terms of utility compared to SAC and DDPG. Although at 30vehicle/km, SAC performs with the least mean delay, as vehicular density increases, MA-DRL tends to outperform all the other algorithms with minimum mean delay.

F. Execution time for DRL algorithms

Fig. 13 shows the execution time for all four DRL algorithms. The execution time for the multi-agent approach is seen to be higher compared to that of the single agent approaches since this method involves complex calculations and is specially designed for complex networks. From the previous results, we can also draw the conclusion that the results of the multi-agent approach outperform the others since in this framework we have considered a complex urban setup for the vehicular model. However, for simpler and less complex frameworks or more rural networks with less number of vehicles, the multi-agent model would not perform well. For such scenarios, single-agent methods can prove to be more ideal.

G. Complexity Analysis

Table III shows the complexity of all four actor-critic type algorithms. For all the algorithms, we consider that the number of variables in the state space is $\frac{\mathcal{N}_s}{2}$ and the space complexity is S_p. Let us assume that the algorithms have actor networks J fully connected layers and critic layers with E fully connected layers where $\int_{actor,j}$ denotes the size of the actor in the j^{th} layer of actor-network and $\int_{critic,i}$ is the size of the critic in the i^{th} layer of critic network. We assume that each fully connected layer contains $M \times N$ matrix with N bias as a vector. Therefore, $(M+1)N$ can be considered as the space complexity of one layer. The SAC algorithm is stochastic while the other three algorithms, i.e., DDPG, TD3 and MA-DRL are deterministic. Hence SAC has a discrete action space while the other three algorithms have continuous action space which is one of the main differences between SAC and DDPG. Thus, the complexity of the SAC algorithm is given in the first row of Table III, where $\sum_{t=1}^{T}(\mathcal{N}_s)$ is the complexity of the action space for a discrete time period. While $\mathcal{O}(S_p)$ is the complexity of state space. Similarly, the complexity of DDPG and TD3 is represented in the second and third row of Table III, respectively, where the complexity of action space is represented by $\mathcal{N}_s$. Additionally, TD3 is a modified version of DDPG which uses two critics, hence, the complexity of the critic network in TD3 is $2 * \left(\sum_{i=0}^{E-1}(\int_{critic,i})^2 + 1\right)$. On the same note, MA-DRL has multiple agents and each agent performs different actions, hence, the size of joint action space can be defined as $|\mathbb{A}|^g$ where g is the number of agents. Since MA-DRL algorithm uses multiple agents and the global critic monitors the overall actor networks, the complexity of the actor-network is exponential to the number of agents used and given by $(\sum_{j=0}^{J-1}(\int_{actor,j})^2 + 1^g$. The complexity of the critic network also increases since only one critic network is responsible for multiple actors, and is represented as $\left(\sum_{i=0}^{E-1}(\int_{critic,i})^2 + 1\right)\mathcal{C}$, where $\mathcal{C}$ is a constant. Additionally, the complexity of the joint action space can be represented as $\mathcal{O}((\mathcal{N}_s)^g - \mathcal{C})$ where C is a constant. Although, the complexity of the multi-agent algorithm is

TABLE III: Computation complexity of the algorithms

SAC	$\mathcal{O}\left(\left(\sum_{j=0}^{J-1}\left(\int_{actor,j}\right)^2+1\right)+\left(\sum_{i=0}^{E-1}\left(\int_{critic,i}\right)^2+1\right)\right)+\mathcal{O}\left(\sum_{t=1}^{T}\left(\mathcal{N}_s\right)\right)+\mathcal{O}\left(S_p\right)$
DDPG	$\mathcal{O}\left(\left(\sum_{j=0}^{J-1}\left(\int_{actor,j}\right)^2+1\right)+\left(\sum_{i=0}^{E-1}\left(\int_{critic,i}\right)^2+1\right)\right)+\mathcal{O}\left(\mathcal{N}_s\right)+\mathcal{O}\left(S_p\right)$
TD3	$\mathcal{O}\left(\left(\sum_{j=0}^{J-1}\left(\int_{actor,j}\right)^2+1\right)+2*\left(\sum_{i=0}^{E-1}\left(\int_{critic,i}\right)^2+1\right)\right)+\mathcal{O}\left(\mathcal{N}_s\right)+\mathcal{O}\left(S_p\right)$
MA-DRL	$\mathcal{O}\left(\left(\sum_{j=0}^{J-1}\left(\int_{actor,j}\right)^2+1\right)^g+\left(\sum_{i=0}^{E-1}\left(\int_{critic,i}\right)^2+1\right)\mathcal{C}\right)+\mathcal{O}\left(\left(\mathcal{N}_s\right)^g-\mathcal{C}\right)+\mathcal{O}\left(S_p\right)$

exponentially higher than that of the other algorithms, from the simulations we can see that for complex urban environments, MA-DRL algorithm achieves significantly better results than single-agent algorithms. However, this algorithm may not be suitable for non-complex environments as it will reach unnecessary overhead [39]–[41].

VI. Conclusion

An intelligent task offloading strategy in a RIS-aided MAEC IoV network was designed considering the state of the criticality of the network and priority-based size-based tasks of the vehicles. We developed an MA-DRL framework using MG that maximizes the mean utility of the IoV network and improves communication quality. The designed framework takes into consideration a hybrid workflow, where vehicles and the MAEC server located at the BS with unused resources are stimulated to share their resources with nearby task vehicles to provide auxiliary computation support. V2V offloads are done via direct links between vehicles since it is a short-ranged transmission, whereas V2I transmissions are aided by both direct links from vehicles to BS as well as via multiple-RISs. The numerical results demonstrated that the RIS-assisted IoV network using the proposed MA-DRL algorithm achieved higher utility than current state-of-the-art networks (not aided by RISs) and other baseline DRL algorithms. The proposed method improves the data rate of the offloaded tasks, reduced the mean delay, and also ensured that a higher percentage of offloaded tasks were completed compared to that of other DRL-based and non-RIS-assisted IoV frameworks.

References

[1] P. Hosseinioun, M. Kheirabadi, S. R. Kamel Tabbakh, and R. Ghaemi, “atask scheduling approaches in fog computing: A survey,” *Trans. Emerg. Telecommun. Technol. T*, vol. 33, no. 3, p. e3792, 2022.

[2] A. M. A. Hamdi, F. K. Hussain, and O. K. Hussain, “Task offloading in vehicular fog computing: State-of-the-art and open issues,” *Future Gener. Comput. Syst.*, 2022.

[3] B.Hazarika *et al.*, “DRL-based resource allocation for computation offloading in IoV networks,” *IEEE Trans Industr Inform*, 2022.

[4] J. Shi, J. Du, J. Wang, J. Wang, and J. Yuan, “Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning,” *IEEE Trans. Veh.*, vol. 69, no. 12, pp. 16 067–16 081, 2020.

[5] Y. Weng, H. Chu, and Z. Shi, “An intelligent offloading system based on multiagent reinforcement learning,” *Secur. Commun. Netw.*, 2021.

[6] F. Karim, B. Hazarika, S. K. Singh, and K. Singh, “A performance analysis for multi-RIS-assisted full duplex wireless communication system,” in *Proc.IEEE ICASSP*, 2022, pp. 5313–5317.

[7] Y. Xu, T. Zhang, Y. Zou, and Y. Liu, “Reconfigurable intelligence surface aided UAV-MEC systems with NOMA,” *IEEE Communications Letters*, vol. 26, no. 9, pp. 2121–2125, 2022.

[8] X. Zhang, Y. Shen, B. Yang, W. Zang, and S. Wang, “DRL based data offloading for intelligent reflecting surface aided mobile edge computing,” in *WCNC 2021*. IEEE, 2021, pp. 1–7.

[9] Y. Zhu, B. Mao, and N. Kato, “A dynamic task scheduling strategy for multi-access edge computing in irs-aided vehicular networks,” *IEEE Trans. Emerg. Top. Comput. Intell.*, 2022.

[10] Q. Wu and R. Zhang, “Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network,” *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 106–112, 2019.

[11] Y. Zhu, B. Mao, and N. Kato, “Intelligent reflecting surface in 6G vehicular communications: A survey,” *IEEE open j. veh. technol.*, vol. 3, pp. 266–277, 2022.

[12] Y. Zhu, B. Mao, Y. Kawamoto, and N. Kato, “Intelligent reflecting surface-aided vehicular networks toward 6G: Vision, proposal, and future directions,” *IEEE Veh. Technol. Mag.*, vol. 16, no. 4, pp. 48–56, 2021.

[13] C. He and J. Xiao, “Joint optimization in intelligent reflecting surface-aided UAV communication for multiaccess edge computing,” *Wirel. Commun. Mob. Comput.*, 2022.

[14] S. Mao, L. Liu, N. Zhang, M. Dong, J. Zhao, J. Wu, and V. C. Leung, “Reconfigurable intelligent surface-assisted secure mobile edge computing networks,” *IEEE Trans. Veh. Technol.*, 2022.

[15] T.Bai *et al.*, “Reconfigurable intelligent surface aided mobile edge computing,” *IEEE Wirel*, vol. 28, no. 6, pp. 80–86, 2021.

[16] J. Liu, M. Ahmed, M. A. Mirza, W. U. Khan, D. Xu, J. Li, A. Aziz, and Z. Han, “RL/DRL meets vehicular task offloading using edge and vehicular cloudlet: A survey,” *IEEE Internet Things J.*, 2022.

[17] F. Fu, Y. Kang, Z. Zhang, F. R. Yu, and T. Wu, “Soft actor–critic DRL for live transcoding and streaming in vehicular fog-computing-enabled IoV,” *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1308–1321, 2020.

[18] S. A. Kazmi, T. M. Ho, T. T. Nguyen, M. Fahim, A. Khan, M. J. Piran, and G. Baye, “Computing on wheels: A deep reinforcement learning-based approach,” *IEEE trans Intell Transp Syst*, 2022.

[19] J. Shi *et al.*, “DRL-based V2V computation offloading for blockchain-enabled vehicular networks,” *IEEE Trans. Mob. Comput.*, 2022.

[20] J. Wang, H. Ke, X. Liu, and H. Wang, “Optimization for computational offloading in multi-access edge computing: A deep reinforcement learning scheme,” *Comput. Netw.*, p. 108690, 2022.

[21] B. Hazarika, K. Singh, S. Biswas, S. Mumtaz, and C.-P. Li, “SAC-based resource allocation for computation offloading in IoV networks,” in *EUCNC*. IEEE, 2022, pp. 314–319.

[22] A. A. Baktayan, I. A. Al-Baltah, and A. A. Abd Ghani, “Intelligent pricing model for task offloading in unmanned aerial vehicle mounted mobile edge computing for vehicular network,” *J. Commun. Softw. Syst.*, vol. 18, no. 2, pp. 111–123, 2022.

[23] L. Yao, X. Xu, M. Bilal, and H. Wang, “Dynamic edge computation offloading for internet of vehicles with deep reinforcement learning,” *IEEE trans Intell Transp Syst*, 2022.

[24] C. Yu, W. Du, F. Ren, and N. Zhao, “Deep reinforcement learning for task allocation in UAV-enabled mobile edge computing,” in *INCoS*. Springer, 2021, pp. 225–232.

[25] M. Z. Alam and A. Jamalipour, “Multi-agent DRL-based hungarian algorithm (MADRLHA) for task offloading in multi-access edge computing internet of vehicles (IoVs),” *IEEE Trans. Wirel. Commun.*, 2022.

[26] X. Zhu, Y. Luo, A. Liu, M. Z. A. Bhuiyan, and S. Zhang, “Multiagent deep reinforcement learning for vehicular computation offloading in IoT,” *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9763–9773, 2020.

[27] H. Xiang, Y. Yang, G. He, J. Huang, and D. He, “Multi-agent deep reinforcement learning based power control and resource allocation for D2D communications,” *IEEE Wireless Commun. Lett.*, 2022.

[28] H. Zhang *et al.*, "Reconfigurable intelligent surface assisted multi-user communications: How many reflective elements do we need?" *IEEE Wireless Commun.*, vol. 10, no. 5, pp. 1098–1102, 2021.

[29] B. Di, H. Zhang, L. Song, Y. Li, Z. Han, and H. V. Poor, "Hybrid beamforming for reconfigurable intelligent surface based multi-user communications: Achievable rates with limited discrete phase shifts," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1809–1822, 2020.

[30] C. Blad, S. Bøgh, and C. Kallesøe, "A multi-agent reinforcement learning approach to price and comfort optimization in HVAC-systems," *Energies*, vol. 14, no. 22, p. 7491, 2021.

[31] A. Wong, T. Bäck, A. V. Kononova, and A. Plaat, "Multiagent deep reinforcement learning: Challenges and directions towards human-like approaches," *arXiv preprint arXiv:2106.15691*, 2021.

[32] J. Feng, "Learning to collaborate: Multi-scenario ranking via multi-agent reinforcement learning," in *Proc. IEEE ICWS*, 2018, pp. 1939–1948.

[33] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[34] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[35] R. Dubey, R. Loka, and A. M. Parimi, "Maintaining the frequency ofAI-based power system model using twin delayed DDPG (TD3) implementation," in *PARC*. IEEE, 2022, pp. 1–4.

[36] N. A. Mosali, S. S. Shamsudin, O. Alfandi, R. Omar, and N. Al-Fadhali, "Twin delayed deep deterministic policy gradient-based target tracking for unmanned aerial vehicle with achievement rewarding and multistage training," *IEEE Access*, vol. 10, pp. 23 545–23 559, 2022.

[37] T. Tiong, I. Saad, K. T. K. Teo, and H. bin Lago, "Deep reinforcement learning with robust deep deterministic policy gradient," in *2020 ICECIE*. IEEE, 2020, pp. 1–5.

[38] T.Haarnoja *et al.*, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Int. Conf. Mach. Learn. Cybern.* PMLR, Jul. 2018, pp. 1861–1870.

[39] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8577–8588, 2019.

[40] S. Jo, C. Jong, C. Pak, and H. Ri, "Multi-agent deep reinforcement learning-based energy efficient power allocation in downlink MIMO-NOMA systems," *IET Commun.*, vol. 15, no. 12, pp. 1642–1654, 2021.

[41] Y. Yang *et al.*, "An overview of multi-agent reinforcement learning from game theoretical perspective," *CoRR*, vol. abs/2011.00583, 2020.

Bishmita Hazarika (Student Member, IEEE) received the B.Tech. degree in Information Technology from Central Institute of Technology, Kokrajhar, India, in 2018, and the M.Tech. degree in Computer Science and Engineering from Indian Institute of Information Technology (IIITG), Guwahati, India, in 2020. She is currently working towards the Ph.D. degree in Communication Engineering at National Sun Yat-sen University (NSYSU), Kaohsiung, Taiwan. Her current research interests include Internet of Vehicles, vehicular fog and edge computing, resource allocation, deep reinforcement learning, wireless networks, digital twin & metaverse.

Keshav Singh (**Member, IEEE**) received the M.Sc. degree in Information and Telecommunications Technologies from Athens Information Technology, Greece, in 2009, and the Ph.D. degree in Communication Engineering from National Central University, Taiwan, in 2015. He currently works at the Institute of Communications Engineering, National Sun Yat-sen University (NSYSU), Taiwan as an Assistant Professor. Prior to this, he held the position of Research Associate from 2016 to 2019 at the Institute of Digital Communications, University of Edinburgh, U.K. From 2019 to 2020, he was associated with the University College Dublin, Ireland as a Research Fellow. He leads research in the areas of green communications, resource allocation, transceiver design for full-duplex radio, wireless edge caching, optimization and machine learning for wireless communications, and reconfigurable intelligent surface-assisted communications.

Sudip Biswas (**Member, IEEE**) received a Ph.D. degree in digital communications from the University of Edinburgh (UEDIN), U.K., in 2017. From 2017 to 2019, he worked as a Research Associate with the Institute of Digital Communications, UEDIN, on optimization and signal processing for 5G and beyond communications. He also has industrial experience with Tata Consultancy Services, India (Lucknow and Kolkata), where he was an Assistant Systems Engineer from 2010 to 2012. He currently works as an Assistant Professor in the Department of Electronics and Communications Engineering at the Indian Institute of Information Technology Guwahati (IIITG). He is an Editor in IEEE Transactions on Green Communications (TGCN). He has previously been involved in two EU FP7 projects: remote radio heads & parasitic antenna arrays (HARP) and dynamic licensed shared access (ADEL), a DST UKIERI project on wireless edge caching, and an EPSRC project on non-orthogonal multiple access (NoMA). He is currently leading the DST SERB-funded project "Signal Processing and Optimization for Configuring Intelligent Reflecting Surfaces for Smart Cities" and has completed the DST SERB-funded project "Signal Processing for Co-existence between Radar and Future Communication Systems" as PI. Furthermore, he also received the SERB Accelerate Vigyan grants on Karyashala and Vritika in 2022 and 2023.

Shahid Mumtaz (**Senior Member, IEEE**) is a Nottingham Trent University (NTU), UK professor. He is an IET Fellow, founder, and EiC of IET "Journal of Quantum Communication," Vice-Chair: Europe/Africa Region- IEEE ComSoc: Green Communications & Computing Society. He authorizes four technical books, 12 book chapters, and 300+ technical papers (200+ IEEE Journals/transactions, 100+ conferences, 2 IEEE best paper awards) in mobile communications. Most of his publication is in the field of Wireless Communication. He is a Scientific Expert and Evaluator for various research funding agencies. In 2012, he was awarded an "Alain Bensoussan fellowship." China awarded him the young scientist fellowship in 2017.

Chih-Peng Li (**Fellow, IEEE**) received the B.S. degree in Physics from National Tsing Hua University, Hsin Chu, Taiwan, and the Ph.D. degree in Electrical Engineering from Cornell University, NY, USA. Dr. Li was a Member of Technical Staff with Lucent Technologies. Since 2002, he has been with National Sun Yat-sen University (NSYSU), Kaohsiung, Taiwan, where he is currently a Distinguished Professor. Dr. Li has served various positions with NSYSU, including the Chairperson of Electrical Engineering Department, the VP of General Affairs, the Dean of Engineering College, and the VP of Academic Affairs. His research interests include wireless communications, baseband signal processing, and data networks. He is now the Director General of the Engineering and Technologies Department, National Science and Technology Council, Taiwan.

Dr. Li is currently the Chapter Chair of IEEE Broadcasting Technology Society Tainan Section. Dr. Li has also served as the Chapter Chair of IEEE Communication Society Tainan Section, the President of Taiwan Institute of Electrical and Electronics Engineering, the Editor of IEEE Transactions on Wireless Communications, the Associate Editor of IEEE Transactions on Broadcasting, and the Member of Board of Governors with IEEE Tainan Section. Dr. Li has received various awards, including the Outstanding Research Award of Ministry of Science and Technology and Outstanding Engineering Professor Award of Chinese Institute of Engineer. Dr. Li is a Fellow of the IEEE.