https://ieeexplore.ieee.org/document/8365853

# Scheduling the Operation of a Connected Vehicular Network Using Deep Reinforcement Learning

Ribal F. Atallah, Chadi M. Assi, and Maurice J. Khabbaz

***Abstract*— Driven by the expeditious evolution of the Internet of Things, the conventional vehicular ad hoc networks will progress toward the Internet of Vehicles (IoV). With the rapid development of computation and communication technologies, IoV promises huge commercial interest and research value, thereby attracting a large number of companies and researchers. In an effort to satisfy the driver's well-being and demand for continuous connectivity in the IoV era, this paper addresses both safety and quality-of-service (QoS) concerns in a green, balanced, connected, and efficient vehicular network. Using the recent advances in training deep neural networks, we exploit the deep reinforcement learning model, namely deep Q-network, which learns a scheduling policy from high-dimensional inputs corresponding to the current characteristics of the underlying model. The realized policy serves to extend the lifetime of the battery-powered vehicular network while promoting a safe environment that meets acceptable QoS levels. Our presented deep reinforcement learning model is found to outperform several scheduling benchmarks in terms of completed request percentage (10–25%), mean request delay (10–15%), and total network lifetime (5–65%).**

***Index Terms*— Internet of Vehicles, deep reinforcement learning, scheduling.**

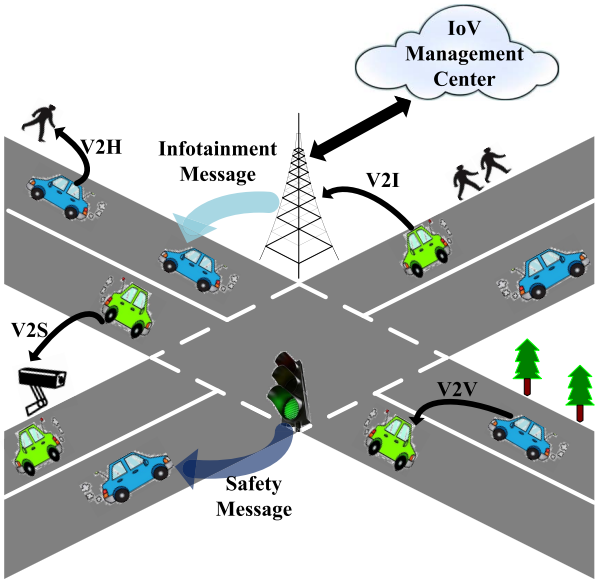


Fig. 1. The internet of vehicles.

## I. Introduction

TO CRUISE towards the 5G technology, intelligence, communication capabilities and processing power will need to be diffused across networks and mobile devices, empowering even the smallest of connected devices to do heavy computational tasks and run rich content and services. Soon enough, the Internet of Things (IoT) paradigm, which is a key enabling technology for the next generation 5G network, will become an absolute reality in modern wireless communications. At this point, enormous number of "things" are being (and will continue to be) connected to the Internet at an unprecedented rate realizing the idea of IoT. Unquestionably, the main strength of IoT is the high impact it will have on several aspects of our everyday-life. The Internet of Vehicles (IoV) is an inevitable convergence of the mobile Internet and the IoT. IoV technology, illustrated in Figure 1, refers to dynamic mobile communication systems that communicate between vehicles and public networks using V2V (vehicle-to-vehicle), V2I (vehicle-to-infrastructure), V2H (vehicle-to-human) and V2S (vehicle-to-sensor) interactions. It enables information sharing and the gathering of information on vehicles, roads and their surroundings. Moreover, IoV features the processing, computing, sharing and secure release of information onto information platforms. Based on this data, the system can effectively guide and supervise vehicles, and provide a variety of multimedia and mobile Internet application services. IoV leverages road objects (*e.g.* traffic lights, cameras, speed sensors, etc.) with the ability to see, hear, think and exchange information related to the safety and comfort of passengers. When commercial vehicles are supplemented with the latest IoV connected technology, the feasibility of vehicle dynamics monitoring, intelligent navigation, fleet management, and value-added services become endless. For this purpose, the transportation research society is working collaboratively to build an end-to-end full-fledge Intelligent Transportation System (ITS) that enhances the user experience, reduces operational costs and promotes a safe driving environment. A revolutionary transportation experience in the IoT era realizes several benefits, including, but not limited to: *a)* greater efficiency by reducing fuel consumption through fuel-saving assistance that takes into account the driving distance, road conditions and driving patterns, *b)* increased safety using remote vehicle diagnostics that promote the

Manuscript received May 24, 2017; revised October 4, 2017, December 21, 2017, and March 10, 2018; accepted April 18, 2018. Date of publication May 25, 2018; date of current version May 1, 2019. This work was supported in part by NSERC and in part by Concordia University. The Associate Editor for this paper was J. M. Alvarez. *(Corresponding author: Chadi M. Assi.)*

R. F. Atallah and C. M. Assi are with CIISE, Concordia University, Montreal QC H3G 1M8, Canada (e-mail: r.atalah@gmail.com; assi@ciise.concordia.ca).

M. J. Khabbaz is with the ECCE Department, Notre Dame University of Louaize, Zouk Mosbeh 72, Lebanon (e-mail: mkhabbaz@ndu.edu.lb).

Digital Object Identifier 10.1109/TITS.2018.2832219

responsiveness of service centers to driver drowsiness, vehicle theft, accidents as well as maintenance requests, *c)* higher reliability by avoiding vehicle downtime as well as reducing expensive unplanned repairs using a vehicle performance tracking system that send maintenance notifications, and *d)* enhanced quality of experience through the support of infotainment services and provisioning on-the-fly access to information systems for the purpose of recuperating some knowledge (*e.g.* about weather conditions) or identifying hot spots (*e.g.* rest stops, restaurants, parking spots, etc.).

### A. IoV Overview

The IoV is foreseen to support a full-fledged, smart and efficient ITS by providing real-time traffic information, context-aware advertising as well as drive-through Internet access, provisioned through the help of stationary IoT GateWays (IoT-GW) deployed along roadways. It is important to note that in a vehicular network an IoT-GW governs all communications taking place within its communication range (*i.e.,* an IoT-GW chooses to either broadcast safety-related messages or transmit non-safety-related service data requested by a particular vehicle). A significant barrier to the widespread deployment of IoT-GWs is the cost of provisioning electrical grid power connections, [1], as well as their remarkable energy consumption. Indeed, it has been reported that energy consumption of mobile networks is growing at a staggering rate [2]. As such, the U.S. Department of Energy is actively engaged in working with industry, researchers, and governmental sector partners through the National Renewable Energy Laboratory (NREL) in order to provide effective measures to reduce the energy use, emissions, and overall transportation system efficiency. NREL emphasizes that energy consumptions from wireless communications should be reduced to 90% by 2050 [3]. Furthermore, from the operators' perspective, energy efficiency not only has great ecological benefits, but also has significant economic benefits because of the large electricity bill resulting from the huge energy consumption of a wireless base station [4]. Following the emerging need for energy-efficient wireless communications as well as the fact that grid-power connection is sometimes unavailable for IoT-GWs, [5], it becomes clear and more desirable to deploy green energy-efficient IoT-GWs, which are equipped with large batteries rechargeable through renewable energy sources such as solar and wind power [6], [7]. Energy-efficient and QoS-oriented scheduling policies must be employed at the IoT-GW in order to guarantee a desired level of performance in an eco-friendly environment.

The major entangled challenge associated with the proper inauguration of a full-fledged connected vehicular network is the efficient control and management of the operation of the IoT-GWs. Indeed, the highly dynamic and stochastic nature of vehicular networks, the randomness in the vehicle arrival process as well as the diversity of the requested services give rise to a particularly challenging scheduling problem for the efficient operation of the IoT-GWs. Multiple studies in the literature have addressed the scheduling problem in the context of V2I communications. For instance, the work in [8], [9], [10], [11], [12] proposed novel V2I scheduling algorithms for a single RoadSide Unit (RSU) equipped with an infinite power source. Other studies addressed the energy consumption issue in the single-RSU (*e.g.,* [13] and [14]) and multi-RSU (*e.g.,* [15]) scenarios, where the RSU is privileged with a priori knowledge of vehicle arrival instances and requests in order to resolve a complex optimization problem.

### B. Motivation and Work Objectives

The successful inauguration of the IoV technology requires the acquisition of core technologies and standards, which will be crucial to securing a strategic advantage. However, the integration of the IoV with other infrastructures should be as important as the foundation of the IoV technology itself. Particularly, the ITS is envisioned to lay a solid basis for the IoV, which will eventually become an integral part of the larger IoT. The objective of this present work is to establish a universal, green, intelligent and scalable scheduling policy which acclimates to the random characteristics of a vehicular environment, overcomes the limiting assumptions and deficiencies of previous studies and finally, establish a vigilant backbone ITS that supports the development of the IoV.

Precisely, herein, this work considers a long roadway segment where several IoT-GWs are deployed in tandem and vehicles arrive/leave the considered road from various entry/exit points. A similar scenario is illustrated in Figure 2. Each IoT-GW is connected using fiber or cellular links to a backend ITS central server, which is the acting agent for all the communications that take place in the network. Following a discrete-time process, the IoT-GWs collect high-dimensional inputs corresponding to the network characteristics at the beginning of each time slot, forwards the collected data to the central agent that devises appropriate actions. In this work, the central agent is trained to realize an optimal scheduling policy that meets the following objectives:

1) Communicate safety messages with minimum latency.
2) Minimize the mean response time as well as the mean total delay of non-safety-related download requests.
3) Satisfy the vehicles' download requirements before their departure from the road.
4) Maintain the entire vehicular network up and running by balancing the power consumption at each IoT-GW.

Unlike the study presented in [16], in this work, the state space dimension is enormously large. Hence, the required computational time and effort to realize an optimal scheduling policy is prohibitively large, a phenomenon commonly referred to as the *curse of dimensionality* [17]. Consequently, recent advances in training deep neural networks (*i.e.* function approximation techniques) are exploited herein in order to overcome this complexity and, thus, promote the feasibility of instantiating a fictitious artificial agent that will: *a)* learn a scheduling policy from high-dimensional inputs using end-to-end deep reinforcement learning, *b)* derive efficient representations of the environment, and *c)* progress towards the development of a successful scheduling policy which meets the above-detailed objectives.

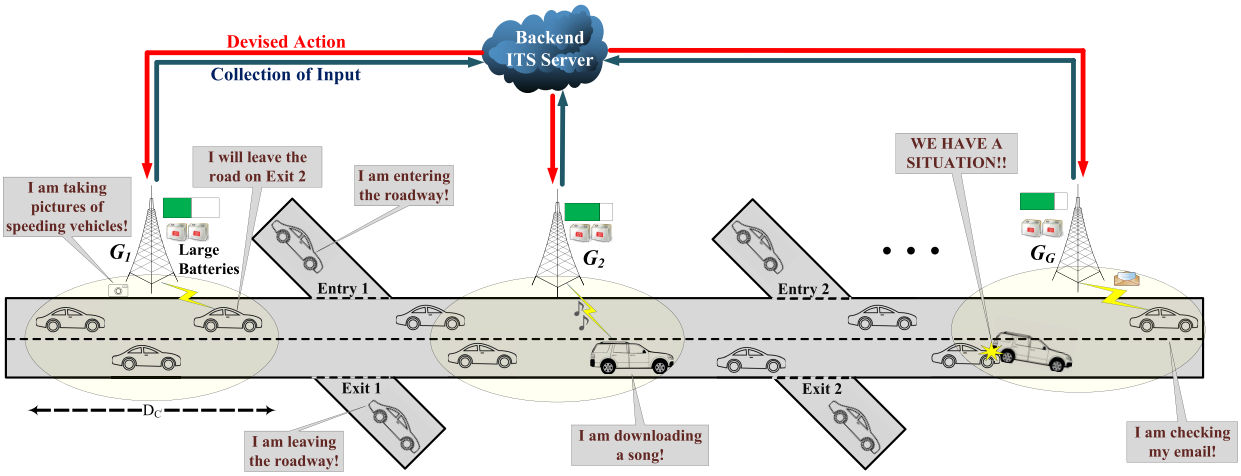


Fig. 2. Energy-limited multi-RSU vehicular network.

This work aims to establish an efficient resource allocation scheduling policy that governs the operation of the IoT-GWs, being the primary points of service in a vehicular environment. The objective of the proposed policy is to reduce the severity of (ultimately mitigate) a subset of the persisting challenges affronted in optimizing the eco-friendly operation of IoT-GWs in a highly dynamic ITS, subject to several random processes such as the emergence of safety-related messages, the vehicles' random speeds and service request sizes as well as the vehicles' arrival and departure points. Additionally, considering the scenario illustrated in Figure 2, it is clear that maintaining a balanced available energy at the IoT-GWs through proper communication and collaboration among the tandem of the connected IoT-GWs shall prolong the operation of the connected network. As such, safety-related information will continue to reach the maximum number of vehicles, thus, promoting a safer driving environment. Furthermore, the proper coordination of multiple IoT-GWs serves the purpose of fully satisfying a vehicle's service request by efficiently dividing the download requirements among the IoT-GWs which fall on that vehicle's path. Consequently, more vehicles will leave the roadway with a Complete Service Request (CSR), which is considered an important QoS metric for the network's overall performance. The work presented herein eliminates the limiting assumptions and addresses the shortcomings of previous studies (*e.g.,* [14], [15], [16]). This work exploits a deep reinforcement learning technique, namely Deep Q-Network, to establish a safe, connected and energy-efficient vehicular network with multiple IoT-GWs. The following points highlight the identifying contributions of this paper:

1) The adaptation of a vehicular network with a central ITS agent that governs the operation of multiple connected Iot-GWs deployed on a long road segment.
2) The realization of an energy-efficient and QoS-oriented scheduling policy, which dictates the operation of the energy-limited IoT-GWs. A deep reinforcement learning agent learns an adaptive dynamic scheduling policy that serves to balance the energy available at the IoT-GWs in order to maintain an operational connected ITS.
3) The exploitation of recent advances in training a deep reinforcement learning agent, which, first, constructs a state representation that implicitly contains the relevant and necessary information about the current situation of the vehicular network, then observes the cost/reward of choosing a particular action, and finally, learns from current and past experience in order to realize an optimal scheduling policy and minimize the overall costs.
4) Establishing a proper and competent simulation framework that adequately portrays the dynamic nature and characteristics of the underlying vehicular network. As such, the integration and training of a proficient deep reinforcement learning algorithm allows a central backend agent to devise optimal real-time decisions.
5) Artificial Intelligence provides the framework and tools to go beyond trivial real-time decision and automation in the IoV era. This work provides a solid ground to introduce machine learning techniques, in particular, deep reinforcement learning, in order to establish a smart, safe and green ITS, which complements the future IoV services and applications.

### C. Paper Organization

The remainder of this paper is structured as follows. Section II presents an overview of the most relevant related work in the context of IoV, V2I scheduling algorithms as well as deep reinforcement learning. A description of the V2I communication scenario is presented in Section III. Section IV presents the adopted vehicular mobility model. Section V lays out a detailed presentation of the deep reinforcement learning model. An MDP model is formulated in Section VI. The performance of the proposed deep reinforcement learning algorithm is examined and compared to other existing scheduling heuristics in Section VII. Finally, concluding remarks are presented in Section VIII.

## II. Related Work

### A. IoV Enabling Technologies

The complete realization of IoV could incur fundamental upgrades to the driving experience through the integration of

intelligence within existing ITS applications. However, various vehicular communications problems are yet to be resolved for the proper realization and quick penetration of IoV enabling applications. Challenges in the design and development of IoV are being investigated by researchers and practitioners interested in the future of vehicular communications. Cheng *et al.* [18] provided a comprehensive review of routing protocols applicable in the context of IoV. The authors described the routing taxonomy from five different perspectives and then laid out preliminary guidelines for the efficient development of IoV routing protocols and technologies. Gerla *et al.* [19] argued that the vehicular cloud constitutes an instance of the envisioned IoV comprising all the protocols and services required for the vehicle grid to operate efficiently and safely. They discussed the challenges associated with the exploitation of vehicular cloud computing to support autonomous and Internet-connected vehicles. Kumar *et al.* [20] argued that the vehicular density as well as the high mobility of vehicles led to constant topological changes, and hence, content distribution became a challenging task. For this purpose, the authors proposed a Bayesian Coalition Game for content distribution using learning automata. The proposed algorithm showed an increased probability of content distribution with a lower end-to-end delay when compared to other content distribution counterparts.

### *B. V2I Scheduling-Based Access Methods*

In the context of V2I communications, scheduling is a decision-making process of arranging, controlling and optimizing the operation of the service provider, and eventually construct an efficient resource allocation policy that meets one or several objectives concurrently. This subsection surveys a selection of existing scheduling-based access methods that supplement the RoadSide Unit (RSU) with an intelligent identity allowing it to make vehicle selection decisions that contribute to realizing a desired objective. Zhang *et al.* [21] and [22] proposed a Unified TDMA-based Scheduling Protocol (UTSP) for V2I communications for the purpose of optimizing the throughput of non-safety applications in Vehicular Ad-hoc Networks (VANETs). In the proposed TDMA scheduling-based method, the RSU collected information such as channel state, the speeds as well as the Access Category characteristic of the vehicles within its communication range. Cheung *et al.* of [8] addressed the Drive-Thru Internet (DTI) application of V2I communications and developed the Dynamic Optimal Random Access (DORA) algorithm with the objective of maximizing the channel utilization subject to time-varying contention severity and capacity levels. In [9], Zhang *et al.* proposed a basic low-complexity V2I access scheme called $D * S$ where the RSU stored the Service Requests (SRs) and the request with the least $D * S$ was served first. $D$ was the SR's deadline and $S$ was the data size to be uploaded to the RSU. The authors then studied the uplink MAC performance of a DTI scenario in [10]. Both the contention nature of the uplink and the realistic traffic model were taken into consideration. Atallah *et al.* [11] proposed two complexity-minimal V2I access schemes and modelled the vehicle's on-board unit buffer's queue as an $M/G/1$ queueing system and captured the V2I system's performance from a vehicle's perspective. Reis *et al.* [12] developed mathematical models to determine the average delay of a packet between a disconnected source-destination pair of vehicles in the presence of RSUs as relays or broadcasters of information.

The algorithms proposed in [8], [9], [10], [11], [12] overlooked the RSU energy consumption pertaining to the proposed scheduling discipline. Given the increasing concern over the energy consumption in wireless networks as well as the highly likely unavailability of permanent power sources in vehicular networks, the conventional design approaches may not be feasible to green communications and should be revisited. As such, Hammad *et al.* [14] addressed the problem of scheduling for energy efficient RSU. Therein, the objective was to minimize the long term energy consumption subject to satisfying the communication requests associated with the passing vehicles. The authors first formulated lower bounds for total energy needed by a RSU in order to serve a finite set of vehicular arrival demands. Then, the authors proposed three online scheduling algorithms which used vehicles' locations and speeds as inputs for a linear optimization problem which dynamically scheduled communication activity. In [23], Zhang *et al.* proposed an energy-efficient RSU deployment algorithm for the purpose of minimizing the transmit power of the RSUs subject to the constraints that all vehicles on the road should be covered. The authors then studied the network performance when the number of RSUs deployed on a roadway varied.

To the best of our knowledge, thus far, the literature has overlooked the possibility of establishing scheduling policies using machine learning techniques. Particularly, deep reinforcement learning augments the RSU with the ability to observe and analyse the environment, make decisions, learn from past experience, and eventually, perform optimal actions, which will serve for the establishment of an efficient ITS in the IoV era. This present work capitalizes on the novel adaptation of Deep Reinforcement Learning techniques to intelligent transportation systems. Precisely, a central ITS agent is deployed and trained to observe and analyze the random turn of vehicular traffic events for the purpose of establishing an optimal IoT-GW energy-aware and Qos-oriented scheduling policy that aims at maximally prolonging the ITS's underlying vehicular network's lifetime. As opposed to the above-surveyed existing work that unrealistically assumes the a priori knowledge of vehicular traffic and data communication parameters (*e.g.* vehicle arrival times, speeds, download request sizes, residence times, etc), this present work relaxes such an assumption and sheds the light over the learning capabilities of the deployed ITS agent that initiates its operation being totally network-information-agnostic. Once adequately trained, this agent will have the ability to cope with the real-time dynamic vehicular traffic variations and schedule the service of arriving vehicles at appropriate IoT-GWs in such a way to maximize the system's overall performance and rewards. Our earlier work presented in [16] was a first step that aimed at investigating the possibility of exploiting

Reinforcement Learning Techniques (RLTs) to govern the operation of a single stationary RoadSide Units (RSUs). Further investigations, have revealed that the stage-one precursary RLTs used therein cannot be applied to scenarios that encompass multiple RSUs. This is especially true since, multi-RSU scenarios such as the one considered herein, suffer from an explosive number of inputs. This, indeed, denies the convergence of the employed RLTs in deriving an optimal scheduling policy. This motivated the exploration and testing of a deeper and more advanced learning technique, namely, Deep Q-Learning, and its application in governing the operation and functionality of an ITS's core vehicular network as presented hereafter. The next subsection lays out a brief introductory overview of the state-of-the-art deep reinforcement learning models.

### C. Deep Reinforcement Learning

In the Reinforcement Learning (RL) paradigm, an agent autonomously learns from past experience in order to maximize some reward signal. Learning to control an agent directly from high-dimensional inputs such as vision and speech is an extremely tedious task, known as the curse of dimensionality. The literature encloses numerous solutions which address this problem (*e.g.,* linear function approximation [24], hierarchical representations [25], state aggregation [26], etc.). These methods greatly rely on the system state representations, thus making the agent not fully autonomous and reducing its flexibility. The use of non-linear function approximation techniques was relinquished as these methods turned out to be unstable and non-converging when used to represent the action-value function [24]. Only recently, Mnih *et al.* [27] presented the Deep Q-Network (DQN) algorithm and tested it in a challenging framework composed of several Atari games. DQN achieved dramatically better results than earlier approaches and professional human players and showed a robust ability to learn representations from very high-dimensional input. In [28], Mnih *et al.* compared their DQN algorithm with the best performing methods from the reinforcement learning literature on 49 Atari games. Results showed that the DQN method outperformed the best existing reinforcement learning methods on 43 of the games without incorporating any of the additional prior knowledge about Atari games used by other approaches. Furthermore, the DQN agent performed at a level that was comparable to that of a professional human games tester across the set of 49 games, achieving more than 75% of the human score on more than half of the games. Mnih *et al.* [27] and [28] exploited an experience replay mechanism ([29]) and a batch reinforcement learning technique ([30]), which made the convergence and stability of the proposed DQN model possible. DQN is foreseen to address the major long-standing challenge of RL by learning to control agents directly from high-dimensional inputs and state spaces. At this level, it becomes necessary to investigate the feasibility of similar techniques in reinforcement learning scenarios where an agent makes decisions that affect the state of the environment. This current work examines a variant of DQN in the context of a V2I communication scenario in a connected vehicular network with multiple RSUs, each of which serves as an IoT-GW.

## III. V2I Communication Scenario

As illustrated in Figure 2, consider a vehicular network consisting of a set of IoT-GWs equipped with large rechargeable batteries. The IoT-GWs are connected to a backend ITS central agent using fiber or cellular communication links. In point of fact, exploiting fiber communication links allows the IoT-GWs to communicate with each other as well as with a backend agent with negligible delays (*i.e.,* in the order of $\mu s$) [31]. As such, actions devised by the intelligent agent to the IoT-GWs are communicated instantly. IOT-GWs are equipped with a single radio for downlink communication and operate independently from one another without any interference. The channel access time is assumed to be time-slotted, and IoT-GWs use transmit power control to maintain constant bit rate reception in downlink V2I communications regardless of the vehicle's location within the IoT-GW's coverage range. It is important to mention that, since there is a strong deterministic component of path loss versus distance, and the power consumption of an energy-efficient IoT-GW is dominated by downlink transmission power [32], then an IoT-GW will generally prefer to communicate with nearby vehicles rather than with more distant vehicles in order to minimize its energy consumption.

Vehicles enter the considered roadway segment from any of the several entry points according to a Poisson process. The vehicle arrival rate from entry $i$ is equal to the vehicle departure rate from exit $i$. As such, the vehicular density in the considered road segment is contingent to the vehicular arrival process from the main entry point to that segment (A well-known technique for flow smoothing in traffic modelling[33]). Upon a vehicle's arrival to an IoT-GW's coverage range, it communicates its downlink service request as well as its expected exit point from the road. It is assumed that each vehicle has a single download request of random size. A vehicle may travel through one or more IoT-GWs communication zones. In the event where a vehicle or a sensor residing within the communication range of an IoT-GW senses hazardous road conditions (*e.g.,* a speeding car or a traffic collision), it raises a safety flag in order to notify the IoT-GW about the existence of a safety message, which should be communicated to the backend ITS server. For this purpose, in the considered scenario herein, a safety message is generated by a randomly selected vehicle in the considered vehicular network at rate of $\lambda_s$ messages per second [34]. It is assumed that there is perfect synchronization between the backend ITS server, the IoT-GWs as well as the vehicles residing within the considered roadway segment with the use of a Global Positioning System (GPS). This work borrows the communication rules from the WAVE protocol suite, where each node in the considered vehicular network (*i.e.,* sensor, vehicle or IoT-GW) periodically broadcasts announcement beacon messages containing information identifying the offered applications (in the case of the IoT-GW), information about the speed, location, direction of travel, download request size, and

existence of a sensed safety message (in the case of a vehicle), and finally, sensed data and status information (in the case of a sensor). Vehicles that are requesting a communication link with the IoT-GW coordinate with the nearby IoT-GW in order to establish a connection.

A deep reinforcement learning agent keeps track of the network conditions of the IoT-GWs. At the beginning of each time slot, each IoT-GW forwards information about all the vehicles and sensors residing within its communication range to the central agent. The collected information is then fed to the deep reinforcement learning agent which devises a scheduling decision. The communicated information via beacon messages is truly very small (in the order of kilobytes), and hence it is not considered as overhead, and consequently, signalling overhead in this context is marginal. Upon receiving the devised action from the central agent, each IoT-GW grants access to a single vehicle or sensor with a permission to either upload the safety message it is carrying or continue downloading the requested data (in the case of a vehicle). In case the IoT-GW chooses to receive a safety message, it notifies the selected vehicle or sensor to transmit the carried safety message then the IoT-GW forwards the safety message to the backend ITS server, which, in turn, broadcasts that message to all IoT-GWs. The safety message is now available at each IoT-GW and can be disseminated to the vehicles. Once a vehicle or a sensor receives the safety message they are carrying through a broadcast, they will drop their safety flag. During the time an IoT-GW is receiving a safety message, its energy consumption is minimal. However, when serving a vehicle's download service request, the IoT-GW's consumed energy increases as the distance to the receiving vehicle increases.

The deep reinforcement learning agent ensures proper communication and coordination among the IoT-GWs in order to maintain a balanced energy between the IoT-GWs. This serves for the extension of the connected network's lifetime. In fact, knowing a vehicle's exit point, the agent arranges distributed service among the IoT-GWs for the completion of the vehicle's download request before its departure from the considered highway segment. Now, it is true that this work interprets download requests as delay tolerant, however, commuting passengers would appreciate a prompt response time during their residence time within an IoT-GW communication range. For instance, if passengers were requesting to download a video file, they would like to start buffering the data and watch the video as soon as possible. The file can be completely downloaded during their transit in forthcoming IoT-GWs. Finally, and most importantly, the IoT-GWs should manage to broadcast safety messages to concerned vehicles with minimal latency in order to conserve a safe driving environment.

## IV. Vehicular Traffic Model

Consider a vehicular network similar to the one illustrated in Figure 2. Several IoT-GWs are deployed in tandem on a long roadway segment with multiple lanes where vehicles enter the road from several entry points. For convenience, we describe the system with unidirectional vehicular traffic; however, the established scheduling policy is applicable to the bi-directional case.[1] Assume that the considered highway segment is experiencing steady free-flow traffic.

According to [34], [35], [36], the vehicular arrival process from a particular entry point of the considered segment follows a Poisson process. Consequently, the overall arrival of vehicles to the entire segment is also a Poisson process [37]. The per-vehicle speeds are independent and identically distributed random variables in the range $[V_{\min}; V_{\max}]$. These speeds are drawn from a truncated Normal distribution with average $\overline{V}$ and standard deviation $\sigma_V$. It is assumed that vehicles maintain their respective speeds constant during their navigation period within the communication range of an IoT-GW [34],[36]. In this work, time is divided into time slots of length $\tau$. Let $J_i$ be the discrete version of the vehicle's residence within any IoT-GW, and let $D_C$ be the length of the segment that falls within the communication area of that IoT-GW. The p.m.f. of $J_i$ has been derived in [38], and is given by:

$$f_{J_i}(j) = \frac{\xi}{2}\left[\operatorname{erf}\left(\frac{\frac{D_C}{j\tau} - \overline{V}}{\sigma_V\sqrt{2}}\right) - \operatorname{erf}\left(\frac{\frac{D_C}{(j-1)\tau} - \overline{V}}{\sigma_V\sqrt{2}}\right)\right] \quad (1)$$

where $J_{\min} \leq j \leq J_{\max}$, and $\xi$ is a normalization constant.

An arriving vehicle communicates its speed, download requirements as soon as it enters the coverage range of an IoT-GW. Also, with the use of the GPS, vehicles inform the IoT-GW about their exit points. Consequently, each IoT-GW keeps track of the characteristics of the vehicles residing within its coverage range. Note that, a vehicle $i$'s download service request size is a uniformly distributed Random Variable $H_i$ between $H_{min}$ and $H_{max}$. $H_i$ is expressed in bits.

## V. Deep Reinforcement Learning Background

Markov Decision Processes (MDPs) offer a standard formalism for describing multi-state decision making in a probabilistic environment. More precisely, an MDP is a discrete-time stochastic control process, where, at reach time step, the process is in some state $x$ and the decision maker chooses a feasible action $a$. Accordingly, the process then moves to a new state $x'$ and awards the decision maker a corresponding reward $r(x, a, x')$. The probability that the process moves into its new state $x'$ is influenced by the chosen action. It is defined by the state transition function $T(x'|x, a)$, which satisfies the Markov Property: given the current state $x$ and an action $a$, the next state $x'$ is conditionally independent of all previous states and actions. The core problem of MDPs is to find a policy for the decision maker, defined by a function $\pi$ that specifies the action $\pi(x)$ that the decision maker will choose when in state $x$. The goal of MDP is to choose a policy $\pi$ that will maximize the following cumulative function of the random rewards.

$$R = \sum_{n=0}^{\infty} \gamma^n r(x_n, a_n) \quad (2)$$

[1]The IoT-GW will learn the characteristics, dynamics and traffic conditions of the underlying environment, and realizes a scheduling policy accordingly.

where $n$ indicates the time step and $\gamma$ is a discount factor between 0 and 1. Note that, the choice of the value of $\gamma$ becomes very crucial for the convergence of the Q-values presented above. In fact, $\gamma = 0$ will make the agent short-sighted by only considering current rewards, while a factor approaching 1 will make it strive for a future high reward. A classical MDP can be solved by value iteration or policy iteration [39] to determine the optimal policy, however, these two methods assume that the decision maker accurately knows the transition function and the reward for all states in the environment. Whereas, in practice, the decision maker may not have an explicit representation of the transition and reward functions. Fortunately, there is a way to learn these functions; the decision maker trades learning time for a priori knowledge through a form of reinforcement learning known as Q-learning. Q-learning is a form of model-free learning, which teaches the decision maker how to behave in an MDP when the transition and/or reward functions are unknown. In Q-learning, each state is first assigned an initial value, called a Q-value, and the correct Q-value can be estimated using the online incremental update stochastic Q-learning algorithm [40] shown in Equation (3), as shown at the bottom of this page, where $r(x_n, a_n)$ is the single step reward and $\alpha(n)$ is the step-size learning rate which is set between 0 and 1. Note that, whenever $\alpha(n) = 0$ the Q-values are not updated and hence nothing is learnt. However, setting a high value for $\alpha(n)$ means that learning occurs quickly.

This technique is known as the value iteration algorithm and it converges to the optimal action-value function, $Q(x_n, a_n) \rightarrow Q^*(x_n, a_n)$ as $n \rightarrow \infty$. In this work, the usability of this approach is impractical due to the extremely high-dimensional system state. As a result, the time needed for the classical Q-learning algorithm to converge increases immeasurably, a phenomenon commonly referred to as the "curse of dimensionality" [17]. As a result, function approximation techniques are used to overcome this widely known limitation of MDPs. In particular, neural networks are exceptionally good at coming up with good features for high dimensional input data. In fact, the action-value function can be represented with a neural network, which takes the current system state and action as input and outputs the corresponding Q-value. This technique is commonly known as deep reinforcement learning.

A neural network with weights $\theta$, referred to as a Q-network [27], is a non-linear function approximator which approximates the action-value function $Q(x_n, a_n)$ by $Q(x_n, a_n; \theta)$. A Q-network can be trained in order to learn the parameters $\theta$ of the action-value function $Q(x_n, a_n; \theta)$ by minimizing a sequence of loss functions, where the $i^{th}$ loss function $L_i(\theta_i)$ is given by:

$$L_i(\theta_i) = \mathbb{E}\left[r_n + \max_{a_{n+1}} Q(x_{n+1}, a_{n+1}; \theta_{i-1}) - Q(x_n, a_n; \theta_i)\right]^2 \tag{4}$$

Note that, $\theta_i$ are the neural network's parameters at the $i^{th}$ update, and the parameters from the previous update, $\theta_{i-1}$ are held fixed when optimizing the loss function $L_i(\theta_i)$. Thus, the term $\left\{r_n + \max_{a_{n+1}} Q(x_{n+1}, a_{n+1}; \theta_{i-1})\right\}$ is the target for iteration $i$, which depends on the neural network's parameters from the last update. Hence, the objective is to find the neural network's set of weights which make the above cost expression as small as possible. This is done using an algorithm known as gradient descent, which repeatedly computes the gradient $\nabla_{\theta_i} L_i(\theta_i)$, and updates the neural network's weights in order to reach a global minimum. Hence, differentiating the loss function with respect to the neural network's parameters at iteration $i$, $\theta_i$ gives the gradient as expressed in equation (5), as shown at the bottom of this page.

The use of batch methods, which utilize the full training set to compute the next update to parameters at each iteration tend to converge very well to local optima. However, often in practice, computing the cost and gradient for the entire training set is extremely slow and sometimes intractable on a single machine, especially when the training dataset is large. Therefore, rather than computing the full expectations in the above gradient equation, it is computationally desirable to optimize the loss function using the Stochastic Gradient Descent (SGD) method. SGD updates the neural network's parameters after seeing only a single or a few training examples. The use of SGD in the neural network setting is motivated by the high cost of running back propagation over the full training set. Reinforcement learning methods tend to diverge when used with non-linear function approximators such as a neural network. In order to avoid the divergence of deep reinforcement learning algorithms, three techniques were introduced in [27], namely:

1) Experience Replay: The parameters of the neural network are updated by performing a SGD step on random samples of past experience rather than the most recent samples of experience. This reduces the correlations between successive updates applied to the network, hence breaking the strong correlations of the training data and reducing the variance of the updates.
2) Fixed Target Network: The neural network's parameters used to compute the target for the $i^{th}$ iteration are held

$$Q(x_n, a_n) := Q(x_n, a_n) + \alpha(n)\left[r(x_n, a_n) + \gamma \max_{a_{n+1}} \{Q(x_{n+1}, a_{n+1})\} - Q(x_n, a_n)\right] \tag{3}$$

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}\left[\left(r_n + \max_{a_{n+1}} Q(x_{n+1}, a_{n+1}; \theta_{i-1}) - Q(x_n, a_n; \theta_i)\right) \nabla_{\theta_i} Q(x_n, a_n; \theta_i)\right] \tag{5}$$

$$L_i(\theta_i) = \mathbb{E}_{x_n, a_n, r_n, x_{n+1} \sim \mathbb{D}}\left[r_n + \max_{a_{n+1}} Q(x_{n+1}, a_{n+1}; \theta^-) - Q(x_n, a_n; \theta_i)\right]^2 \tag{6}$$

fixed for intervals of several thousand SGD steps, then they are updated with the current realized parameters. A target Q-network reduces the correlations between the target and the obtained Q-values, thus making the problem less non-stationary.

3) Reward Normalization: This technique limits the scale of the error derivatives and ensures that gradients are well-conditioned by eliminating the instability in error back-propagation, such that no outlier update has too much impact on the learning. However, the learning agent can no longer differentiate between small and large rewards. As such, normalizing the rewards adaptively to sensible range increases the robustness of the derived gradients and stabilizes the deep learning process.

As a result, deep reinforcement learning is used to minimize the loss function given in equation (6), as shown at the bottom of the previous page. where $\mathbb{D}$ is the experience replay memory and the symbol $\sim$ is used to denote that a minibatch of transitions $(x_n, a_n, r_n, x_{n+1})$ is sampled from $\mathbb{D}$. $\theta^-$ are the parameters of the target Q-network. By using the above three techniques, the convergence of the underlying deep reinforcement learning algorithm has been empirically proven in [27] and [28]. On the other hand, the drawback of using the experience replay is the substantial memory requirements.

The proposed algorithm herein is an off-policy algorithm as it learns an optimal action $a_n = \max_a Q(x_n, a_n; \theta)$ while still choosing random actions to ensure adequate exploration of the state space. In fact, a common problem often faced by autonomous learning agents is the trade-off between acting to gain information and acting to gain rewards. A widely used technique for active exploration in off-policy algorithms is the $\epsilon$-greedy strategy where the agent selects the current optimal action with a probability $1 - \epsilon$, and selects a random action with probability $\epsilon$.

## VI. Markov Decision Process Model

In this work, a deep reinforcement learning agent, deployed at each IoT-GW, interacts with the vehicular environment in a sequence of actions, observations, and costs. At each time step, the agent selects an action from the set of feasible actions at that time. The IoT-GW will either suspend its services to listen to an announced safety message, or transmit data to a vehicle with a download request. The agent then observes the changes in the environment and modifies the system state representation. The agent also receives a reward accordingly. In order to achieve the goals laid out in Section III, all the IoT-GWs should operate in a consistent, orderly and efficient way to balance the vehicular network's available energy between them, report safety-related messages promptly and deliver a pleasing quality of experience for the travelling vehicles. After each selected action, an IoT-GW receives a step reward, which is a normalized indicator of how well is it contributing to accomplishing the previously-mentioned goals. The objective of learning is to construct an optimal action selection policy at each IoT-GW that serves to maximize the overall network's performance. A natural measure of performance is the discounted cumulative reward function laid out in Equation (2). Now, it is worthwhile mentioning that the received single-step reward depends on the whole previous sequence of actions and observations. As such, the impact of an action may only be seen after several hundreds/thousands of time-steps ahead.

### A. Input From the Environment

At the beginning of an arbitrary time slot (time $t_n$), each IoT-GW $G_i$ observes the surrounding vehicular environment, collects all the parameters associated with the set of in-range vehicles and sensors, and chooses its action from the set of feasible actions at $t_n$. The input of IoT-GW $G_i$ from the environment at time $t_n$ is:

- $\overline{P_{G_j,n}}$: a vector of size $G$ containing the remaining energy at each IoT-GW $G_j$ where $j = 1, 2, \cdot, G$ and $0 \leq P_{G_j,n} \leq P_t$, where $P_t$ is the IoT-GW's battery capacity.
- $N_{j,n}$: the number of vehicles residing within $G_j$'s communication range, $0 \leq N_{j,n} \leq N_{\max}$.
- $\overline{J_{j,i,n}} = \{J_{j,i,1}, J_{j,i,2}, \cdots, J_{j,i,N_{i,n}}\}$: $G$ vectors of respective sizes $N_{j,n}$ containing the remaining discrete sojourn times of each vehicle $v_{j,i}$, $i \in (1, 2, \cdots N_{j,n})$ and $0 \leq J_{j,i,n} \leq J_{\max}$
- $\overline{H_{j,i,n}} = \{H_{j,i,1}, H_{j,i,2}, \cdots, H_{j,i,N_{j,n}}\}$: $G$ vectors of respective sizes $N_{j,n}$ containing the remaining request sizes for each vehicle $v_{j,i}$, $0 \leq H_{j,i,n} \leq H_{\max}$.
- $\overline{W_{j,i,m}} = \{W_{j,i,1}, W_{j,i,2}, \cdots, W_{j,i,N_{j,n}+S_j}\}$: a vector of size $N_{j,n} + S_j$, where $S_j$ is the number of sensors within $G_j$'s communication range. $\overline{W_{j,i,m}}$ contains the waiting times of the safety messages in the buffer of vehicles or sensors. In the case where vehicle $v_{j,i}$ or sensor $s_j$ has no safety message to upload, its corresponding $W_{j,i,m}$ is set to a negative value (-1).
- $\overline{d_{j,i,n}} = \{d_{j,i,1}, d_{j,i,2}, \cdots, d_{j,i,N_{j,n}}\}$: $G$ vectors of respective sizes $N_{j,n}$ containing the separation distances between $G_j$ and each of its in-range vehicles, $0 \leq d_{j,i,n} \leq G_R$, where $G_R$ is the transmission range of the IoT-GWs.

The agent fully observes the current network state and is able to realize the system state representation at time $t_n$, denoted herein by $x_n$.

### B. Immediate and Expected Costs

Let $a_{j,n}$ denote $G_j$'s action at time step $t_n$. Let $A_{j,n}$ be the set of admissible actions for $G_j$ at $t_n$, therefore, $a_{j,n} \in A_{j,n}$. At time $t_n$, each IoT-GW chooses its action, and accordingly, the network pays an immediate cost, a scalar value that reflects the righteousness of the IoT-GWs' actions. The immediate cost (negative reward) is the sum of the following normalized quantities:

1) Power consumed by each IoT-GW to transmit data to a selected vehicle. Note that the received signal strength decays exponentially as the separation distance between the IoT-GW and the selected vehicle increases [32]. Assuming that the IoT-GW controls its transmit power to maintain constant bit rate transmission, the power consumed to transmit to closer vehicles is significantly

less than that consumed when transmitting to farther ones.

2) Normalized waiting time of the vehicles that have not received any service yet. This incurred cost trains the IoT-GW to minimize the average vehicles' response time.
3) Normalized total delay of completed service requests. Once a vehicle completes its download request, the system is charged a cost corresponding to the total delay of that completed service request. Since the deep reinforcement learning agent thrives to maximize its rewards (minimize negative rewards, *i.e.,* costs), it will learn to minimize the total end-to-end delay of download requests.
4) Penalty incurred on network due to the departure of a vehicle with an incomplete service request. The value of this penalty is a normalized quantity proportional to the remaining request size of the departing vehicle. As such, the IoT-GW is encouraged to fulfill the vehicles' download requests before their departure from the considered roadway. Recall that a vehicle communicates its exit point upon its arrival to the highway, and the IoT-GWs should coordinate to completely serve that vehicle before its departure from the highway.
5) Penalty incurred on network due to the early cut-off of one of the IoT-GWs. The network receives this penalty when any of the IoT-GWs shuts down once its battery is drained. The value of this penalty is proportional to the available energy at the other IoT-GWs. As a result, the network learns to balance the power consumption among the IoT-GWs such that when one of the IoT-GWs shuts down, the other IoT-GWs have very little amounts of energy remaining in their batteries.

Obviously, the reward/cost expression does not have a closed-form mathematical expression, and hence the use of deep reinforcement learning methods to explore the effect of the IoT-GWs' actions on the system. Now, it is important to mention that even if the impact of the occurrence of the above-described event unveils in a single time-step (*i.e.,* when a vehicle departs or when an IoT-GW shuts down), the deep reinforcement learning agent realizes that the sequence of all its previous actions lead to this current system state. This is a clear example that the feedback about an action may sometimes be received after many thousands of time steps have elapsed.

### C. Markov Decision Model

Consider the non-stationary MDP model consisting of the set of data $(E, A, C_n, r_n, \phi_0)$ where:

- $E$ is the state space where, at any time $t_n$, the system state $x_n \in E$.
- $A_n$ is the action space at time $t_n$, where the set of feasible actions for $G_j$ at time $t_n$ is $A_{j,n} \subset A_n$. $A_{j,n} = \{0, 1, 2, \cdots, N_{j,n}\}$, where if $a_{j,n} = 0$, then $G_j$ will receive a safety message and directly forward it to the backend ITS server. The IoT-GWs will broadcast the safety message in the following time slot. If $a_{j,n}$ is any value $k$ between 1 and $N_{j,n}$, then $G_j$ chooses to transmit packets to the $k^{th}$ vehicle. Also, let $a_n = \{a_{1,n}, a_{2,n}, a_{3,n}\}$.
- $C_n \subset E \times A_n$ is a measurable subset of $E \times A_n$ and denotes the set of possible state-action combinations at the beginning of the $n^{th}$ time slot [39]. $C_n$ contains the graph of a measurable mapping, $g_n : E \rightarrow A_n$, *i.e.,* $(x_n, g_n(x_n)) \in C_n$ for all $x_n \in E$.The set $C_n(x_n) = \{a_n \in A_n | (x_n, a_n) \in C_n\}$ is the set of valid actions in state $x_n$ at time $t_n$ [39].
- $r_n : C_n \rightarrow \mathbb{R}$ is a measurable function where $r_n(x_n, a_n)$ gives the single step reward (negative reward or cost in our case) of the system at time $t_n$ if the current state is $x_n$ and actions $\{a_{1,n}, a_{2,n}, a_{3,n}\}$ were taken.
- $\phi_0$ is a stochastic transition probability kernel that assigns to each state-action pair $(x_n, a_n) \in C_n$ a probability measure over $E \times \mathbb{R}$. The transition probability kernel $\phi_0$ gives rise to the state transition probability kernel, $\phi$, which, for any $(x_n, a_n, x_{n+1}) \in E \times A_n \times E$ triplet gives the probability of the transition from state $x_n$ to another state $x_{n+1}$ provided that action $a_n$ was made in state $x_n$.

The goal of the agent is to interact with the IoT-GWs and select actions that maximize future rewards. Herein, future rewards are discounted by a factor of $\gamma$ per time step. Therefore, the future discounted return at time $t_n$ is $R_n$ given by:

$$R_n = \sum_{t=t_n}^{\infty} \gamma^{t-t_n} r_n \tag{7}$$

### D. Proposed Solution Approach

Recall that, the objective of this work is to realize an optimal scheduling policy which will govern the operation of the IoT-GWs in order to minimize the total expected negative rewards and, as a result, achieve the goals laid out in Section III. This problem has been formulated as an MDP whose states are modelled as a Markov chain, and a state-action dependent cost is incurred at each stage. The next section lays out the neural network training process using a deep reinforcement learning algorithm, which will serve to realize an optimal policy for the above-presented MDP. The deep Q-learning algorithm is presented in Algorithm 1.

## VII. Simulation and Results Discussion

### A. The Learning Phase

At the beginning of each time slot, each IoT-GW forwards the underlying network characteristics to the deep reinforcement learning agent. As such, the agent's input, detailed in Section VII.A becomes ready to be fed to the neural network. The input to the neural network consists of $(G \times N_{max} \times 4)$ corresponding to the characteristics of all vehicles within all $G$ IoT-GWs, and $\sum_j S_j$ that correspond to the safety messages' waiting time in the sensors' buffers. Recall that $S_j$ is the number of sensors within $G_j$'s communication range. We use a neural network consisting of two hidden layers as it was shown in [41] that two stages of feature extraction yields

**Algorithm 1** Deep Q-Learning With Experience Replay and Fixed Target Network

Initialize replay memory $\mathbb{D}$ to capacity $\boldsymbol{C}$
Initialize Q-network with random weights $\boldsymbol{\theta}$
Initialize target Q-network with random weights $\boldsymbol{\theta^- = \theta}$
**for** episode = 1, M **do**
  Collect network characteristics to realize state $\boldsymbol{x_0}$
  **for** n = 0, T **do**
    $\boldsymbol{a_n} = \text{argmax}_a \ \boldsymbol{Q(x_n, a_n; \theta)}$ with probability $\boldsymbol{1 - \epsilon}$.
    Otherwise, actions in $\boldsymbol{a_n}$ are selected randomly.
    Execute $\boldsymbol{a_n}$ and observe $\boldsymbol{r_n}$ and $\boldsymbol{x_{n+1}}$
    Store transition $\boldsymbol{(x_n, a_n, r_n, x_{n+1})}$ in $\mathbb{D}$
    Sample random minibatch of transitions $\boldsymbol{(x_n, a_n, r_n, x_{n+1})}$ from $\mathbb{D}$
    Set the target to $\boldsymbol{r_n}$ if episode terminates at $\boldsymbol{n+1}$, otherwise, target is $\boldsymbol{r_n} + \max_{a_{n+1}} \boldsymbol{Q(x_{n+1}, a_{n+1}; \theta^-)}$
    Perform a SGD update on the current Q-network parameters $\boldsymbol{\theta}$
    Every $\boldsymbol{C}$ steps, set the target Q-network parameters $\boldsymbol{\theta^- = \theta}$
  **end for**
**end for**

better accuracy than one. The first hidden layer of the neural network convolves $G$ $(N_{max} \times 4)$ filters and then applies a rectifier nonlinearity [41]. The second hidden layer convolves $G$ $(N_{max} \times 1)$ filters, again followed by a rectifier nonlinearity. The final hidden layer is fully-connected and consists of $(G \times N_{max})$ rectifier units. The output layer is a fully connected linear layer with a single output for each valid action.

The examined scenario herein considers a vehicular network consisting of three IoT-GWs, as depicted in Figure 2. In this case, the size of the Neural Network input vector is $(3 \times N_{max} \times 4)$, where each input is a continuous random variable. As such, the state space enumerating all the possible input cases is infinite, thus corroborating the curse of dimensionality problem. Note that, it is true that extending the considered roadway and deploying more IoT-GWs will further enlarge the input vector to the neural network, thus requiring more learning and simulation time.

The literature encloses several deep reinforcement learning methods that use the system state and action as input to the neural network, *e.g.* [42]. The disadvantage of such techniques is the need for a separate forward pass to compute the Q-value of each action, resulting in a cost that scales linearly with the number of actions. The work of [28] suggested an architecture in which there is a separate output for each possible action, and the system state is the only input to the neural network. The outputs correspond to the predicted Q-values of the individual actions for that input state. As such, a single forward pass through the neural network can compute the Q-values for all possible actions in a given state. This current work adopts this method to train the neural network. In the training phase of the simulations, an $\epsilon$-greedy method was applied where $\epsilon$ decreased linearly from 1.0 to 0.1 over the first million training samples, and fixed at 0.1 thereafter.

TABLE I
SIMULATION INPUT PARAMETERS

| Parameter | Value |
|---|---|
| IoT-GW Battery Capacity | $3 \times 50Ah$ batteries |
| Time slot length | $\tau = 0.1$ (s) |
| Vehicular densities | $\rho \in [2; 11]$ (veh/km) |
| Min and Max vehicle speed | $[60; 140]$ (km/h) |
| Min and Max request size | $[2; 12]$ (MB) |
| Safety Message Arrival rate | $\lambda_s = 1/60$ (msgs/s) |
| IoT-GW covered segment | $D_C = 1000$ (m) |
| Vehicles and IoT-GW radio range | 500 (m) |
| Channel data bit rate | $B_c = 9$ (Mbps) |
| Learning rate | $\alpha(n) = 1/n$ |
| Discount factor | $\gamma = 0.5$ |

The agent was trained on a total of 50 million samples, which corresponds to two months of actual vehicular network operation. The replay memory used in training the neural network was of size 1 million most recent training samples, and the size of the minibatches was 100. Table I lists the values of all the hyper-parameters and optimization parameters as well as the simulator's input parameters. After the training phase, the resulting neural network is exploited in order to devise scheduling decisions for the IoT-GWs.

### *B. Simulation Setup*

The V2I communication scenario described in Section IV is simulated taking into account the vehicular traffic model presented in Section V. Thorough simulations were conducted using the Simulation for Urban MObility (SUMO) simulator [43], which provides realistic vehicular mobility traces used as input for the learning and testing simulation process. The SUMO mobility traces were used in order to run the simulation of the considered vehicular network and generate 50 million samples each corresponding to a snapshot of the system at a particular time slot. These samples are used to train the deep neural network in order to realize an optimal scheduling policy. After establishing the optimal schedule determined by the DQN algorithm, another set of mobility traces was used from SUMO to test the performance of the realized policy. Network simulations are done using Veins simulation framework [44]. The realized centralized MAC algorithm is simulated and built on top of the WAVE PHY layer, taking into account all the requirements and characteristics of the IEEE 802.11p protocol (*i.e.,* frequency band, channel bandwidth, communication distance, length of time slot, etc.), which have been extensively studied in the literature (*e.g.,* [21], [22], and [45]). It is also important to mention here that the considered V2I communication scenario assumes the availability of a Line-of-Sight (LOS) between the IoT-GW and the vehicle with which it is communicating. As such, it is reasonably justifiable to simplify the complex environment by assuming that the effect of shadowing is insignificant [46]. Finally, the path loss exponent's value for free-space communications is set to 3 [14], [15]. Recall that, the IoT-GWs are using a transmit power control which

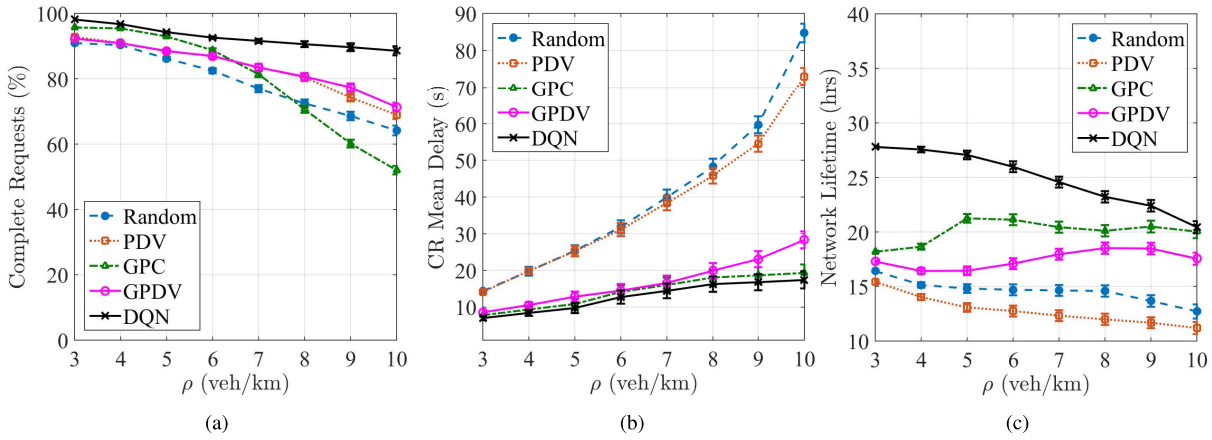


Fig. 3. Performance evaluation and comparisons. (a) CR Percentage. (b) CR mean delay. (c) Network lifetime.

evaluates the required power to maintain a constant Signal-to-Noise Ratio (SNR). Clearly, as the distance separating the vehicle and the IoT-GW increases, the minimum required Received Signal Strength Indicator (RSSI) increases as well, thus forcing the IoT-GW to spend more power to maintain a constant data rate transmission without increasing the Bit Error Rate (BER) [47]. Now, in a typical wireless communication environment with the availability of a LOS between transmitter and receiver and a noise floor of $-85$ dBm, an SNR value greater than or equal to 5 is required in order to maintain a constant bit rate of $B_c = 9$ Mbps and keep the BER to a value less than $10^{-9}$ [48].

The simulations are run multiple times with the same set of mobility traces and the DQN algorithm is compared with other heuristics under the same network conditions to ensure a fair comparison. The presented results herein were averaged over multiple runs to ensure a 95 % confidence interval. It is also worthwhile mentioning that the reported DQN results pertain to simulation only and cannot be compared to analytical results due to the fact that it is impossible to theoretically characterize the scheduling policy resulting from the deep reinforcement learning algorithm. The proposed DQN algorithm is compared with four other scheduling algorithms namely:

1) Random: Random vehicle selection algorithm where, at time $t_n$, the IoT-GW randomly chooses a vehicle within its communication range to be served [11].
2) PDV: Prioritizing Departing Vehicles, where at time $t_n$, the IoT-GW randomly chooses a vehicle from the set of vehicles, which are departing from the next exit. This heuristic is inspired by the algorithm presented in [9].
3) GPC: Greedy Power Conservation algorithm where, at time $t_n$, the IoT-GW chooses the closest vehicle which contributes to the lowest energy consumption compared to the remaining vehicles residing within its communication range [14].
4) GPDV: Greedily Prioritize Departing Vehicles, where at time $t_n$, the IoT-GW chooses the closest vehicle from the set of vehicles, which are departing from the next exit. This hybrid heuristic is a combination of PDV and GPC algorithms.

Under all the above scheduling algorithms, if there exists a safety message flag within any of the IoT-GWs' communication ranges, the corresponding IoT-GW will halt its services and notify the vehicle or sensor carrying that message to transmit it. The IoT-GW will then forward the safety message to the backend ITS server, and in the forthcoming time slot, all the IoT-GWs will broadcast the received safety message.

In this section, the performance of the proposed DQN algorithm is evaluated in terms of: *a)* Vehicles' Completed Request (CR) Percentage which is the percentage of completely fulfilled vehicle requests, *b)* CR Mean Delay and *c)* Network Lifetime being the time until one of the IoT-GWs cuts-off.

### C. Simulation Results

Figure 3 evaluates the performance of the deep reinforcement learning agent when compared with the four previously described scheduling algorithms. Figure 3(a) plots the percentage of vehicles leaving the vehicular network with a complete service request as a function of the vehicular density. It is clear that the number of CRs decreases as $\rho$ increases under all the scheduling algorithms. In fact, as $\rho$ increases, the likelihood of selecting a certain vehicle will decrease, independent of the scheduling discipline. Consequently, a vehicle will spend less time receiving service and the total number of vehicles departing from the vehicular network's communication range with complete service requests will decrease. Figure 3(a) also shows that the DQN method outperforms all four other scheduling algorithms in terms of CR percentage. Under a random selection policy, there is no vehicle prioritization, and therefore, the number of vehicles whose associated download request is fulfilled decreases remarkably as the vehicular density increases. Now, for GPC, each IoT-GW is admitting to service the vehicle which resides in the minimal energy consumption zone compared to the set of remaining in-range vehicles. Whenever $\rho$ is small, a large portion of the vehicles have enough time to complete their download request while being the closest to the IoT-GW, however, when $\rho$ increases, the time during which a vehicle is closest to the IoT-GW is not enough to fully complete its download request. This clearly results in deteriorated QoS levels for larger values of $\rho$ when the IoT-GWs are implementing a greedy power conservation

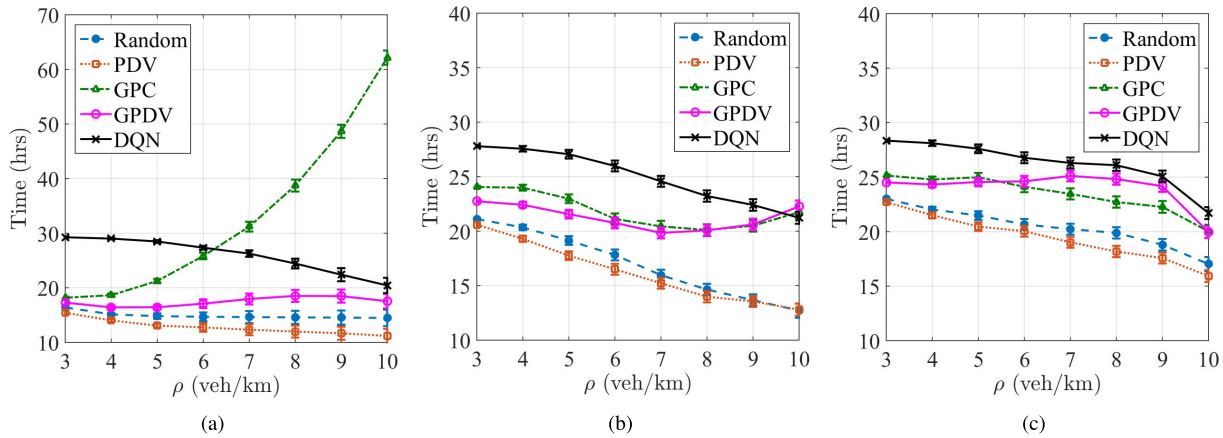


Fig. 4. IoT-GWs' lifetime. (a) Lifetime of $G_1$. (b) Lifetime of $G_2$. (c) Lifetime of $G_3$.

scheduling policy. Under both PDV and GPDV, the IoT-GWs are prioritizing the vehicles that are departing from the next exit, and consequently, the two scheduling algorithms show similar results in terms of CR percentage. Furthermore, as $\rho$ increases, and due to the prioritization notion, PDV and GPDV show better results than the random selection algorithm as well as the GPC method. Finally, it is clear from Figure 3(a) that the DQN agent outperforms all the scheduling algorithms it is compared to. Specifically, on average, DQN outperforms *1)* the Random scheduling algorithm by 18.6 %, *2)* the PDV algorithm by 12 %, *3)* the GPC algorithm by 21.2 %, and *4)* the GPDV algorithm by 10.9 %. Recall that during the training phase, a vehicle departing from the network with an incomplete service request is a undesired event which incurs a high cost. Therefore, the DQN agent will learn to avoid such an event and as such, the deployment of the well-trained DQN agent guarantees that the majority of departing vehicles have completed their download service request.

Figure 3(b) plots the CRs mean total delay as a function of vehicular density. Under the random and PDV algorithms, it is clear that most of the vehicles are receiving service from multiple IoT-GWs. On the other hand, under GPC, GPDV as well as DQN, it is evident that most of the CRs are fulfilled by a single IoT-GW as the mean CR delay is less than the average vehicle residence time within an IoT-GW's communication range (*i.e.,* at an average vehicles' speed of 100 km/h, a vehicle spends 36 seconds in an IoT-GW's communication range). Although GPC and GPDV show good results in terms of CR mean delay, however, DQN outperforms both these algorithms by 10.2 % and 21.1 % respectively. This indicates that the DQN agent is sometimes choosing to serve a vehicle even if it is neither the closest nor departing soon since that decision contributes to minimizing the mean CR total delay.

Figure 3(c) plots the network lifetime, defined as the time until an IoT-GW cuts-off. The improvement DQN achieves over the other four scheduling algorithms varies from 5 to 60 % in terms of the network lifetime. This is a clear implication of the penalty incurred on the DQN agent, in the training phase, due to the early cut-off of one of the IoT-GWs. Recall that since this penalty is proportional to the remaining energy at the other IoT-GWs, the DQN agent learns to balance the power consumption among the tandem of IoT-GWs. This fact is illustrated in Figure 4 where it is clear that all three IoT-GWs have a similar lifetime. It is worthwhile mentioning that, according to [32], the power consumption increases exponentially as the distance between the transmitter and receiver increases. As a result, under GPC, the lifetime of $G_1$ increases as the vehicular density increases. The interpretation of this result is as follows: as more vehicles reside within the IoT-GW's coverage range, it becomes more likely that the closest vehicle which still requires service is very close to that IoT-GW. As such, the latter consumes minimal amounts of energy to serve the closest vehicle, and as such, its battery lifetime increases. However, this is only true for $G_1$ since when the vehicles arrive to $G_2$ and $G_3$, a large portion of them have completed their service, and since an IoT-GW has to choose the closest vehicle which still requires service, it is less likely to find that vehicle in low energy consumption zones. As such, under GPC, the network lifetime is influenced by the lifetime of the downstream IoT-GWs, which is illustrated in Figure 4.

Figure 5 compares the performance of the DQN algorithm with the other benchmark scheduling algorithms as the vehicle mean service request is varied between 6 and 10 MB under a fixed vehicular density of $\rho = 6$ veh/km. Figures 5(a) and 5(b) show that the percentage of CRs as well as their mean delay decrease as the vehicles' mean service size increases under all scheduling algorithms. This is an expected and consistent result. Also, it is clear from Figures 5(a) and 5(b) that the DQN method achieves better results in terms of completed requests percentage and mean delay when compared to the other four scheduling algorithms. Specifically DQN records an enhanced average CR percentage of 18 % over Random selection algorithm, 12 % over PDV, 9.7 % over GPC, and 5 % over GPDV.

In terms of network lifetime, the DQN method outperforms all its counterparts for vehicles' mean request sizes less than 8 MB by 13 % (over GPDV) to 71 % (over PDV). However, when $\bar{Q} = 10$ MB, the GPC algorithm results in a 9 % longer network lifetime than DQN. This is due to the fact that

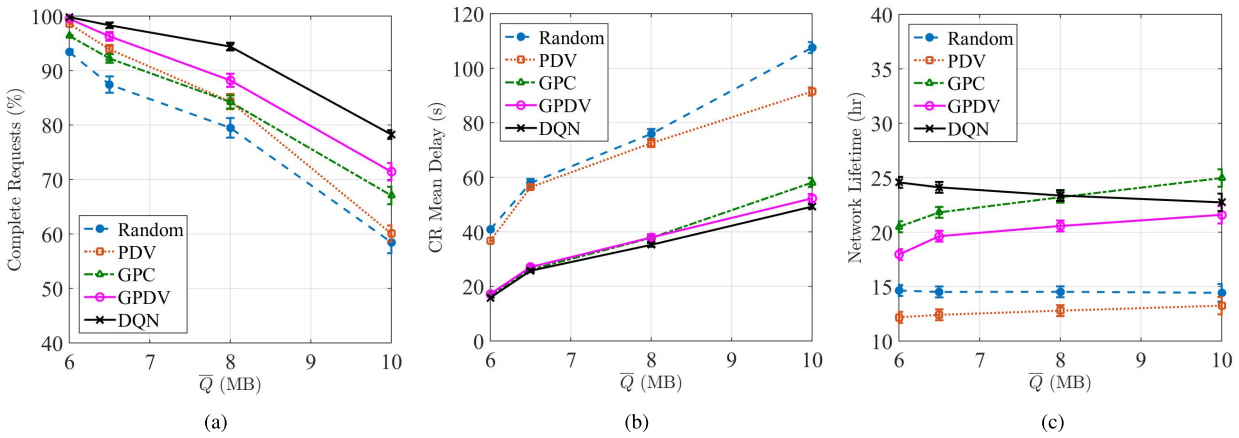


Fig. 5. Performance Evaluation for $\rho = 6$ (veh/km). (a) CR percentage. (b) CR mean delay. (c) Network lifetime.

when arriving vehicles request to download large file sizes, it becomes less likely to serve these vehicles by a single IoT-GW, and chances are high that the closest vehicle to an IoT-GW still requires service. As earlier mentioned, an IoT-GW serving close-by vehicles consumes less amounts of energy, and as such, it extends its battery lifetime. It is true that GPC outperforms DQN in terms of network lifetime for larger values of vehicle requests, however, this is at the expense of deteriorated QoS levels revealed by a smaller percentage of complete service requests and longer CR delays.

## VIII. Conclusion

Given the recent advances and concerns accompanying the Internet of Things, it has become extremely necessary to realize a vigilant backbone ITS that supports the development of the Internet of Vehicles. For this purpose, this paper developed an artificial DQN agent to support the efficient operation of a vehicular network by learning a scheduling policy from high-dimensional inputs using end-to-end deep reinforcement learning. This agent derives efficient representations of the environment, learns from past experience, and progress towards the realization of a successful scheduling policy in order to extend the lifetime of a green, safe and connected vehicular network and achieve acceptable levels of QoS. The proposed DQN algorithm outperformed several existing scheduling benchmarks in terms of completed request percentage (average improvement of 10.9 - 21.2 %), mean request delay (average improvement of 10.2 % - 21.1 %) and total network lifetime (improvement of 13 - 71 %) under variable vehicular densities and vehicle request sizes.

## References

[1] S. Pierce, "Vehicle-infrastructure integration (VII) initiative: Benefit-cost analysis: Pre-testing estimates," Intell. Transp. Syst. Joint Program Office, U.S. Dept. Transp., Washington, DC, USA, Tech. Rep., Mar. 2007.

[2] E. C. Strinati and L. Hérault, "Holistic approach for future energy efficient cellular networks," *e i Elektrotechnik und Informationstechnik*, vol. 127, no. 11, pp. 314–320, 2010.

[3] U.S. Department of Energy, NREL. (2016). *Transportation and the Future of Dynamic Mobility Systems*. [Online]. Available: http://www.nrel.gov/transportation/sustainable-mobility-initiative.html

[4] D. Lister, "An operators view on green radio," in *Proc. IEEE Int. Workshop Green Commun.*, 2009.

[5] K. Tweed, "Why cellular towers in developing nations are making the move to solar power," *Sci. Amer.*, Jan. 2013. [Online]. Available: https://www.scientificamerican.com/article/cellular-towers-moving-to-solar-power/

[6] V. Chamola and B. Sikdar, "Solar powered cellular base stations: Current scenario, issues and proposed solutions," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 108–114, May 2016.

[7] R. Atallah, M. Khabbaz, and C. Assi, "Energy harvesting in vehicular networks: A contemporary survey," *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 70–77, Apr. 2016.

[8] M. H. Cheung, F. Hou, V. W. S. Wong, and J. Huang, "DORA: Dynamic optimal random access for vehicle-to-roadside communications," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 4, pp. 792–803, May 2012.

[9] Y. Zhang, J. Zhao, and G. Cao, "On scheduling vehicle-roadside data access," in *Proc. 4th ACM Int. Workshop Veh. Ad Hoc Netw.*, 2007, pp. 9–18.

[10] Y. Zhuang, J. Pan, V. Viswanathan, and L. Cai, "On the uplink MAC performance of a drive-thru Internet," *IEEE Trans. Veh. Technol.*, vol. 61, no. 4, pp. 1925–1935, May 2012.

[11] R. F. Atallah, M. J. Khabbaz, and C. M. Assi, "Modeling and performance analysis of medium access control schemes for drive-thru Internet access provisioning systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3238–3248, Dec. 2015.

[12] A. B. Reis, S. Sargento, F. Neves, and O. K. Tonguz, "Deploying roadside units in sparse vehicular networks: What really works and what does not," *IEEE Trans. Veh. Technol.*, vol. 63, no. 6, pp. 2794–2806, Jul. 2014.

[13] Z. Yan, B. Li, X. Zuo, and T. Gao, "Fair downlink traffic scheduling for energy sustainable vehicular roadside infrastructure," in *Proc. Int. Conf. Connected Vehicles Expo (ICCVE)*, Nov. 2014, pp. 1092–1097.

[14] A. A. Hammad, T. D. Todd, G. Karakostas, and D. Zhao, "Downlink traffic scheduling in green vehicular roadside infrastructure," *IEEE Trans. Veh. Technol.*, vol. 62, no. 3, pp. 1289–1302, Mar. 2013.

[15] A. Khezrian, T. D. Todd, G. Karakostas, and M. Azimifar, "Energy-efficient scheduling in green vehicular infrastructure with multiple roadside units," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 1942–1957, May 2015.

[16] R. F. Atallah, C. M. Assi, and J. Y. Yu, "A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4592–4601, Jun. 2017.

[17] W. B. Powell, "What you should know about approximate dynamic programming," *Naval Res. Logistics*, vol. 56, no. 3, pp. 239–249, Apr. 2009.

[18] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, "Routing in Internet of vehicles: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2339–2352, Oct. 2015.

[19] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Mar. 2014, pp. 241–246.

[20] N. Kumar, J. J. P. C. Rodrigues, and N. Chilamkurti, "Bayesian coalition game as-a-service for content distribution in Internet of vehicles," *IEEE Internet Things J.*, vol. 1, no. 6, pp. 544–555, Dec. 2014.

[21] R. Zhang, J. Lee, X. Shen, X. Cheng, L. Yang, and B. Jiao, "A unified TDMA-based scheduling protocol for vehicle-to-infrastructure communications," in *Proc. WCSP*, Oct. 2013, pp. 1–6.
[22] R. Zhang, X. Cheng, L. Yang, X. Shen, and B. Jiao, "A novel centralized TDMA-based scheduling protocol for vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 411–416, Feb. 2015.
[23] B. Zhang, G. Zhu, S. Xu, and N. Zhang, "Energy-efficient roadside units deployment in vehicular ad hoc networks," in *Proc. 6th Int. Conf. Wireless, Mobile Multi-Media (ICWMMN)*, Nov. 2015, pp. 6–10.
[24] J. N. Tsitsiklis and B. Van Roy, "Analysis of temporal-diffference learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 1075–1081.
[25] P. Dayan and G. E. Hinton, "Feudal reinforcement learning," in *Advances in Neural Information Processing Systems*. Burlington, MA, USA: Morgan Kaufmann, 1993, p. 271.
[26] S. P. Singh, T. Jaakkola, and M. I. Jordan, "Reinforcement learning with soft state aggregation," in *Proc. Adv. Neural Inf. Process. Syst.*, 1995, pp. 361–368.
[27] V. Mnih *et al.* (2013). "Playing Atari with deep reinforcement learning." [Online]. Available: https://arxiv.org/abs/1312.5602
[28] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.
[29] L.-J. Lin, "Programming robots using reinforcement learning and teaching," in *Proc. AAAI*, 1991, pp. 781–786.
[30] S. Lange, T. Gabel, and M. Riedmiller, "Batch reinforcement learning," in *Reinforcement Learning*. Berlin, Germany: Springer, 2012, pp. 45–73.
[31] I. Yaqoob, I. A. T. Hashem, Y. Mehmood, A. Gani, S. Mokhtar, and S. Guizani, "Enabling communication technologies for smart cities," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 112–120, Jan. 2017.
[32] T. S. Rappaport, *Wireless Communications: Principles and Practice*, vol. 2. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.
[33] Transport for London. (2010). *Traffic Modelling Guidelines: TfL Traffic Manager and Network Performance Best Practice*. [Online]. Available: http://content.tfl.gov.uk/traffic-modelling-guidelines.pdf
[34] M. Khabazian and M. K. M. Ali, "A performance modeling of connectivity in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2440–2450, Jul. 2008.
[35] S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy, "Analytical model for connectivity in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3341–3356, Nov. 2008.
[36] M. J. Khabbaz, W. F. Fawaz, and C. M. Assi, "A simple free-flow traffic model for vehicular intermittently connected networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1312–1326, Sep. 2012.
[37] E. Cascetta, *Transportation Systems Engineering: Theory and Methods*, vol. 49. Medford, MA, USA: Springer, 2013.
[38] M. J. Khabbaz, H. M. K. Alazemi, and C. M. Assi, "Delay-aware data delivery in vehicular intermittently connected networks," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 1134–1143, Mar. 2013.
[39] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.
[40] C. Szepesvári, *Algorithms for Reinforcement Learning* (Synthesis Lectures on Artificial Intelligence and Machine Learning), vol. 4. Medford, MA, USA: Morgan & Claypool, 2010.
[41] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2146–2153.
[42] M. Riedmiller, "Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method," in *Proc. Eur. Conf. Mach. Learn.*, Berlin, Germany: Springer, 2005, pp. 317–328.
[43] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—Simulation of urban mobility: An overview," in *Proc. Int. Conf. Adv. Syst. Simulation*, 2011, pp. 1–6.
[44] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2011.
[45] A. M. S. Abdelgader and L. Wu, "The physical layer of the IEEE 802.11 p wave communication standard: The specifications and challenges," in *Proc. World Congr. Eng. Comput. Sci.*, vol. 2. 2014, pp. 22–24.
[46] C. Sommer, S. Joerer, M. Segata, O. Tonguz, R. L. Cigno, and F. Dressler, "How shadowing hurts vehicular communications and how dynamic beaconing can help," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1411–1421, Jul. 2015.
[47] R. Steele, J. Whitehead, and W. C. Wong, "System aspects of cellular radio," *IEEE Commun. Mag.*, vol. 33, no. 1, pp. 80–87, Jan. 1995.
[48] J. P. Pavon and S. Choi, "Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 2. May 2003, pp. 1108–1113.

**Ribal F. Atallah** received the B.E. degree in computer engineering from the Notre Dame University of Louaize, Lebanon, in 2009, the M.Sc.E. degree in computer engineering from Lebanese American University in 2012, and the Ph.D. degree in information and systems engineering from Concordia University, Montreal, Canada, in 2017. He is currently establishing machine learning algorithms to protect the smart grid against cyber attacks. His research interests include intelligent transportation systems, queuing theory, cyber security, artificial intelligence, and deep learning.

**Chadi M. Assi** received the Ph.D. degree from City University of New York (CUNY) in 2003. He is currently a Full Professor with Concordia University. His current research interests are network design and optimization, network modeling, and network reliability. He was a recipient of the Prestigious Mina Rees Dissertation Award from CUNY in 2002 for his research on wavelength-division multiplexing optical networks. He is on the Editorial Board of IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE TRANSACTIONS ON COMMUNICATIONS, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGIES.

**Maurice J. Khabbaz** received the Ph.D. degree from Concordia University, Montreal, Canada, in 2012. He is currently an Assistant Professor of electrical, computer and communications engineering and the Director of the CISCO Networking Academy, Notre Dame University of Louaize, Lebanon. His research interests are 5G, Internet of Things, intelligent transportation systems, vehicular networks, cloud and datacenter networking, network function virtualization, software defined networking, delay-/disruption-tolerant networking, network optimization, modeling and performance analysis. He also serves on the Editorial Board of IEEE COMMUNICATIONS LETTERS.