A Reinforcement Learning Technique for Optimizing Downlink Scheduling in an Energy-Limited Vehicular Network

Ribal F. Atallah, Chadi M. Assi, *Senior Member, IEEE*, and Jia Yuan Yu

***Abstract*—In a vehicular network where roadside units (RSUs) are deprived from a permanent grid-power connection, vehicle-to-infrastructure (V2I) communications are disrupted once the RSU's battery is completely drained. These batteries are recharged regularly either by human intervention or using energy harvesting techniques, such as solar or wind energy. As such, it becomes particularly crucial to conserve battery power until the next recharge cycle in order to maintain network operation and connectivity. This paper examines a vehicular network whose RSU dispossesses a permanent power source but is instead equipped with a large battery, which is periodically recharged. In what follows, a reinforcement learning technique, i.e., protocol for energy-efficient adaptive scheduling using reinforcement learning (PEARL), is proposed for the purpose of optimizing the RSU's downlink traffic scheduling during a discharge period. PEARL's objective is to equip the RSU with the required artificial intelligence to realize and, hence, exploit an optimal scheduling policy that will guarantee the operation of the vehicular network during the discharge cycle while fulfilling the largest number of service requests. The simulation input parameters were chosen in a way that guarantees the convergence of PEARL, whose exploitation showed better results when compared with three heuristic benchmark scheduling algorithms in terms of a vehicle's quality of experience and the RSU's throughput. For instance, the deployment of the well-trained PEARL agent resulted in at least 50% improved performance over the best heuristic algorithm in terms of the percentage of vehicles departing with incomplete service requests.**

***Index Terms*—Energy, optimization, reinforcement learning, vehicular ad hoc networks (VANETs).**

I. Introduction

FUTURE safety and comfort features in vehicles will make extensive use of connectivity. This includes connectivity between devices within a vehicle, vehicle-to-vehicle connectivity, as well as vehicle-to-infrastructure (V2I) connectivity [1]. A full-fledged connected vehicular network helps prevent accidents, facilitate safe driving, and provide accurate real-time traffic information. With the rapid growth of real-time communication and service requirements, a fully connected vehicular network offering a high level of performance is the primary goal for researchers. Due to the pernicious environmental impact of the emissions associated with communication networks and their heavy contribution to the global power consumption, establishing an energy-efficient vehicular network has recently become an urgent priority. As the number of users as well as the network size increase, the need for energy-efficient networks prevail, hence forcing the research industry to develop ecofriendly communication networks that achieve acceptable quality-of-service (QoS), especially in highly dynamic environments, such as vehicular networks.

The highly mobile facet of a vehicular network leads to several road fragments where a vehicle, or even a small vehicular ad hoc network (VANET), may find itself isolated from any means of communication with other VANETs or RoadSide Units (RSUs). To extend a vehicular network's coverage area, a costly solution suggests the deployment of multiple RSUs over long roadway segments [2]. Another possible solution is to increase the coverage range of VANETs' wireless nodes (i.e., vehicles and RSUs): a solution which is associated with several drawbacks such as smaller data rates and increased packet collision, resulting in elevated delivery delays. Both proposed solutions result in high levels of energy consumption, especially by RSUs. According to the U.S. Department of Transportation [3], it is expected that 40% of all rural free-way roadside infrastructure would be equipped with a solar-powered battery by year 2050 due to the unavailability of a power-grid connection. Consequently, it is now remarkably important to schedule the RSUs' operation efficiently in order to achieve minimum energy consumption and maintain network operation.

In the case where the RSU uses transmit power control to maintain constant bit rate reception, the power consumed to transmit to closer vehicles is significantly less than that consumed when transmitting to farther ones. As a result, it seems like an RSU operating under a strict energy conservation mode tends to serve the vehicles residing in low energy consumption zones of its coverage range (i.e., the closest vehicles). It is true that such a strategy would preserve the maximum amount of RSU's available energy; however, the increasing number of vehicles leaving the RSU's communication range with incomplete service requests (SRs) instigate unsatisfied users, which is a clear indicator of an unacceptable QoS. This paper examines a vehicular network where an RSU is derived from a permanent grid-power connection, but instead, the RSU is equipped with large batteries that are periodically recharged. An example of such a scenario is illustrated in Fig. 1, where the RSU batteries are recharged using solar energy harvesting techniques.

Manuscript received March 30, 2016; revised September 17, 2016; accepted October 22, 2016. Date of publication October 26, 2016; date of current version June 16, 2017. This work was supported by Concordia University and National Science and Engineering Research Council of Canada (NSERC) grants. The review of this paper was coordinated by the Guest Editors of the CVS Special Issue.

The authors are with Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 2W1, Canada (e-mail: r.atalah@gmail.com; assi@ciise.concordia.ca; jiayuan.yu@concordia.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2016.2622180

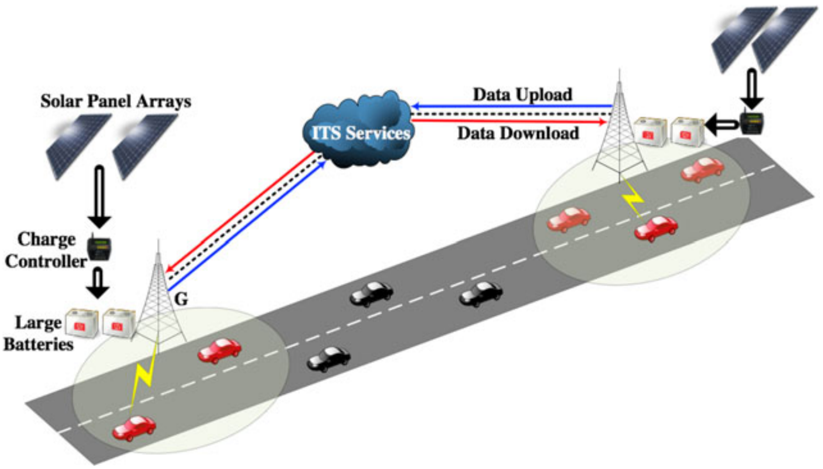

Fig. 1. Solar-powered RSU in an energy-limited VANET.

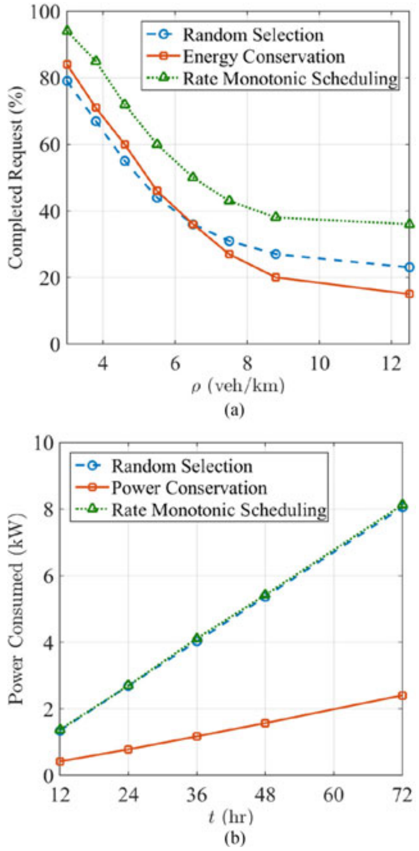

Fig. 2. V2I scheduling algorithms performance. (a) Completed request percentage. (b) Power consumption.

In a highly mobile vehicular network, changes in the network's topology and data-traffic load are frequent. To ensure an energy-efficient RSU performance, a scheduling protocol must change its vehicle selection policy to account for the aforementioned changes in network conditions. In other words, the RSU should implement an intelligent and adaptive scheduling algorithm that efficiently exploits battery power during times where the RSU is delineated with a limited amount of energy. In fact, an RSU leveraged with a smart identity has the ability to adapt its scheduling policy to diverse network scenarios, which persuade an energy-efficient RSU operation. It is important to mention that the limited energy constraint arises in a multitude of scenarios, such as solar-powered battery discharge at night, as illustrated in Fig. 1, insufficient amounts of harvested energy to support the RSU operation, high cost of grid-power connection, etc. An RSU exercising a reinforcement learning (RL)-based scheduling protocol learns how to adapt to the persisting topology and load changes of a vehicular network and, hence, admits vehicles to service in such a way that limits the RSU power consumption while maintaining an acceptable QoS for the arriving vehicles.

A. *Problem Statement and Motivation*

In this paper, a protocol for energy-efficient adaptive scheduling using reinforcement learning (PEARL) is proposed for the purpose of optimizing the operation of an RSU during its battery discharge period. The operation of PEARL is summarized as follows.

1) Collect traffic and network information for a sufficient amount of time to realize the SR load as well as the number of vehicles residing within the RSU's communication range at equilibrium.
2) Observe the environment and constructs PEARL's state representation of the system.
3) Engage in an exploration phase that allows PEARL to learn an optimal scheduling policy that maximizes a designated reward expression.
4) Exploit the realized optimal scheduling policy at the beginning of each time slot and admits a single vehicle to service.

A Markov decision process (MDP) model is formulated for the purpose of efficiently utilizing the available RSU energy while maintaining the vehicular network's operation and acceptable QoS. An RL approach, in particular the Q-learning algorithm, is proposed in order to grant the RSU the required artificial intelligence to realize an optimal scheduling policy that minimizes energy consumption and achieves acceptable levels of QoS. The need for a competent, smart, and green vehicular environment motivates the establishment of an intelligent scheduling policy that meets multiple objectives. Fig. 2 plots the energy consumption, as well as the percentage of completed SRs in a downlink vehicular networking scenario similar to the one illustrated in Fig. 1.[1] Fig. 2(a) shows that whenever the RSU

[1] The simulation framework, scheduling algorithms, and parameters are presented in detail in Section VI.

is operating under the energy conservation mode (i.e., serving only the vehicles residing in low energy consumption zones), the percentage of the completed number of requests deteriorates dramatically as the vehicular density ρ increases. The random vehicle selection (RVS) as well as the rate monotonic scheduling (RMS) algorithms both outperform the greedy energy conservation method in terms of the completed request percentage, especially when the network load increases and more vehicles are present within the communication range of the energy-limited RSU. On the other hand, Fig. 2(b) plots the power capacity required to maintain network operation when the discharge period varies between 12 and 72 h. It is clear that the RSU power consumption is significantly small when the network is operating under an energy conservative scheduling policy compared with the other two scheduling disciplines. As such, it becomes remarkably important to find an optimal scheduling policy that realizes the two objectives of minimal energy consumption with the largest fraction of completed requests.

The remainder of this paper is structured as follows. Section II summarizes the related work and distinguishes the work presented herein from the existing studies in the literature. Section III presents the adopted vehicular traffic model. Section IV lays out the complete theoretical formulation of PEARL. The performance of PEARL is examined and compared with three RSU scheduling heuristics in Section V. Section VI concludes the paper and describes a wealth of open future directions.

II. Related Work

In the context of vehicular networking, scheduling is a decision-making process whose outcome is an efficient, ultimately optimal, joint channel access regulation and resource allocation policy that has to be adopted by vehicles, as well as RSUs for the purpose of realizing one or several objectives concurrently. In order to establish a fully operational intelligent transportation system (ITS), there exists a multitude of remarkably challenging objectives whose realization is possible through the design of appropriate scheduling algorithms. Section II-A lays out a selection of scheduling-based access methods, which supplement the RSU with an intelligent identity, allowing it to make vehicle selection decisions that contribute to realizing a desired objective.

A. Scheduling-Based Access Methods

Cheung *et al.* [4] indicated that the drive-thru Internet (DTI) application of V2I communications suffered from a random access problem. In fact, more than one of the vehicles residing within the RSU's communication range may require Internet access simultaneously. This gives rise to a joint random access and spectrum allocation problem whose resolution is challenging. To this end, the authors developed the dynamic optimal random access algorithm with the objective of maximizing the channel utilization subject to time-varying contention severity and capacity levels. In a similar scenario to [4], Niyato and Hossain [5] examined the V2I wireless access for streaming applications in a public transportation system. The authors formulated an optimization problem with the objective of providing a cost-minimal wireless connectivity that satisfies the end-users QoS requirements. Tan *et al.* [6] modeled the vehicular data download process using a series of transient Markov reward processes. Their objective was to characterize the distribution of a vehicle's downloaded data volume throughout its residence time within an RSU's range. The authors computed the influence of traffic density, vehicle speed, and RSUs transmission range on the amount of downloaded data. In [7], the authors proposed a basic low-complexity V2I access scheme called $D * S$ where the RSU stored the SRs and where the request with the least $D * S$ was served first. D is the SR's deadline and S is the data size to be uploaded to the RSU. The authors then studied the uplink MAC performance of a DTI scenario in [8]. Both the contention nature of the uplink and the realistic traffic model were taken into consideration. Atallah *et al.* [9] proposed two complexity minimal V2I access schemes and modeled 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.

The algorithms proposed in [4]–[9] overlooked the RSU energy consumption pertaining to the adopted 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. Section II-B surveys related research works that proposed RSU scheduling methods, which addressed the energy limitation in a vehicular environment.

B. Energy-Aware Vehicular Networks

Ibrahim [10] focused on using solar cell energy harvesting to provide an alternative power source for stationary RSUs. The goal was to design an efficient and adaptive energy-harvesting module which could be used with different types of embedded RSUs. A power management scheduler was deployed that predicts traffic status based on the historical data, and switches the RSU between ON (active) and OFF (idle or power saving) states. Hammad *et al.* [11] addressed the problem of scheduling for energy efficient RSU. Therein, the objective was to minimize the long-term power consumption subject to satisfying the communication requests associated with the passing vehicles. The authors first formulated lower bounds for total energy needed by an RSU in order to serve a finite set of vehicular arrival demands. Lower bounds were obtained by assuming that the total number of arriving vehicles and their respective speeds and associated requests are made available *a priori* to the RSU. Then, the authors proposed three online scheduling algorithms that used vehicles' locations and speeds as inputs for a linear optimization problem, which dynamically scheduled communication activity. However, under the three proposed algorithms, the scheduler is interrupted whenever a vehicle arrives to the RSU. More interruptions occur as the vehicle flow rate increases, and as such, the efficacy of the proposed algorithms becomes questionable.

Khezrian *et al.* [12] presented an energy-efficient scheduling scheme in the presence of multiple RSUs deployed along a highway, which are interconnected using cellular communication links. The authors considered the case of a unicast RSU-to-vehicle communication scenario only. Integer linear programming bounds were derived for the normalized minimum and maximum energy usage of a single RSU, and then, four online scheduling algorithms were proposed and evaluated in terms of total RSU energy consumption. The reported results in [12] showed that, in order to achieve near optimal energy consumption, online scheduling algorithms require some *a priori* information, which may not be always available. In [13], the authors addressed the same problem as [12] and introduced the concept of a virtual control node (VCN), which is considered connected to all RSUs through physical wires. The authors then derived a temporal graph that shows the connected nodes in the network, and used to find the minimum number of active RSUs needed to maintain a fully connected network. The reported results therein showed that the RSU's transmission range has a great impact on the total number of active RSUs required.

C. Novel Contributions

The following points highlight the identifying contributions of this paper.

1) Unlike the work presented in [4]–[9], this paper realizes a scheduling policy that recognizes that the RSU is equipped with a limited-lifetime power source.
2) An illiterate energy-limited RSU *G* is deployed alongside a road segment where arriving vehicles request access to the Internet infrastructure. *G* is not provided with any information related to the vehicle arrival process or the network expected load. *G*'s operation is dictated by PEARL, which explores the evolution of the vehicular network and exploits an adaptive dynamic policy in order to limit its energy consumption while retaining an acceptable QoS.
3) PEARL implements an RL algorithm which maximizes the long-term system rewards conceded by the total number of downloaded packets, as well as the number of completed SRs. PEARL considers that the event of the departure of a vehicle with incomplete download request is an undesired event, which induces remarkable penalties on the system's performance.
4) This paper develops, analyses, and evaluates an iterative algorithm that finds an optimal RSU scheduling policy using the history of interactions with the environment. By applying the *Q*-learning method of RL, the RSU maintains a state–action–rewards table, which is updated after each scheduling decision in order to tune the scheduling policy towards maximum returns.

This paper establishes the first step in introducing artificial intelligence and RL scheduling methods to vehicular environments for the purpose of conserving the RSU's battery while providing a competent QoS. The next sections lay out the vehicular traffic model, as well as PEARL's theoretical formulation.

III. Vehicular Traffic Model

This study adopts a discrete-time free-flow traffic model characterized by homogeneous and uninterrupted light vehicular traffic flowing over a 1-D roadway segment of fixed length (i.e., D_C in Fig. 1). The interarrival time of vehicles under free-flow traffic conditions is exponentially distributed. As such, vehicle arrivals from a single lane follow a Poisson process [14]. When multiple lanes are considered, vehicle arrivals from each of these lanes follow independent and identically distributed Poisson processes. It follows that the overall vehicle arrival process from all lanes is the sum of independent identically distributed Poisson processes, which is also a Poisson process with rate μ_V vehicles per second. In the vehicular traffic model presented herein, the total discharge period T is divided into N time slots (referred to as the planning horizon later), each of length τ seconds. Let S_n be the set of vehicles residing within the communication range of *G* at the beginning of the nth time slot (where $n = 1, 2, \ldots, N$). *G* schedules to serve one vehicle v_i, where $v_i \in S_n$, at the beginning of the nth time slot.

According to [15], under free-flow traffic condition, the speed v_i of an arbitrary arriving vehicle i is a normally distributed random variable whose probability density function (p.d.f.) is given by

$$f_{v_i}(v_i) = \frac{1}{\sigma_v\sqrt{2\pi}} e^{\left[-\left(\frac{v_i - \overline{V}}{\sigma_V\sqrt{2}}\right)^2\right]}. \tag{1}$$

Khabbaz *et al.* [14] assumed justifiably that $v_i \in [V_{\min}; V_{\max}]$, and accordingly, the speeds of arriving vehicles follow a truncated version of $f_{v_i}(v_i)$, which is given by

$$\widehat{f_{v_i}}(v_i) = \frac{2 f_{v_i}(v_i)}{\text{erf}\left(\frac{V_{\max} - \overline{V}}{\sigma_v\sqrt{2}}\right) - \text{erf}\left(\frac{V_{\min} - \overline{V}}{\sigma_v\sqrt{2}}\right)}. \tag{2}$$

Furthermore, since a vehicle's speed is maintained constant during the vehicle's navigation period within the RSU's coverage range [15], then, according to [14], the p.d.f. of a vehicle's residence time is

$$f_{J_i}(t) = \frac{\xi D_C \sigma_V^{-1}}{t^2\sqrt{2\pi}} \exp\left[-\left(\frac{\frac{D_C}{t} - \overline{V}}{\sigma_V\sqrt{2}}\right)^2\right] \tag{3}$$

where ξ is a normalization constant such that the integral of $f_{J_i}(t)$ over $[\frac{D_C}{V_{\max}}; \frac{D_C}{V_{\min}}]$ is 1. Let $R_i = \lceil \frac{J_i}{\tau} \rceil$ denote the discrete-time equivalent of J_i. R_i has been derived in [9] and is given by

$$f_{R_i}(r) = \frac{\xi}{2}\left[\text{erf}\left(\frac{\frac{D_C}{r\tau} - \overline{V}}{\sigma_V\sqrt{2}}\right) - \text{erf}\left(\frac{\frac{D_C}{(r-1)\tau} - \overline{V}}{\sigma_V\sqrt{2}}\right)\right] \tag{4}$$

where $R_{\min} \leq r \leq R_{\max}$. Note that, in what follows, the notation R_i^n is used to denote the discrete sojourn time remaining for vehicle i at the beginning of the nth time slot.

An arriving vehicle communicates its speed and download requirements as soon as it enters the coverage range of the RSU *G*. Consequently, *G* keeps record of all vehicles within its range as well as their associated service requirements. In this paper, the download SR of a vehicle i is a uniformly distributed random variable H_i between $H_{\min}$ and $H_{\max}$.

In the case where the RSU uses transmit power control to maintain constant bit rate reception, the power consumed to serve closer vehicles is significantly less than that consumed when serving farther ones. In fact, the RSU's power consumption increases exponentially as the receiving vehicle moves farther [16]. Moreover, less power is required to serve a vehicle at a specific distance under a lower data rate.

Section IV lays out the mathematical formulation of PEARL being an RL-based scheduling algorithm deployed at the RSU for the purpose of regulating its energy consumption.

IV. Protocol for Energy-Efficient Adaptive Scheduling Using Reinforcement Learning Theoretical Formulation

A. Overview

The main objective of PEARL is to build an RSU that, at the beginning of each time slot, selects a vehicle to serve in such a way that maximizes its long-term reward. Recall that the RSU's reward is a performance metric for the total number of downloaded bits as well as the number of fulfilled vehicle requests per discharge period. PEARL is a well-trained agent that, given

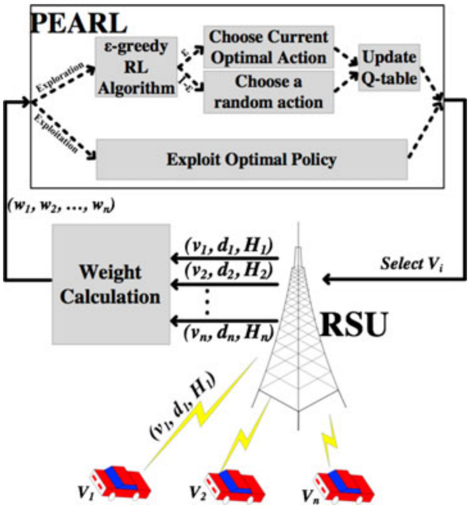

Fig. 3. PEARL's operation.

the traffic characteristics, the RSU's power budget, and the total discharge period, is able to achieve highest reward returns. In the exploration/learning phase, PEARL is tuned and enhanced as the vehicular network's operation evolves and the RSU explores the various observations, actions, and associated rewards. During this phase, and since PEARL is first unaware of the network conditions and state transitions, it follows an epsilon-greedy exploration method (see [17]), which allows the agent to keep exploring the evolution of the underlying network for the purpose of fine tuning its scheduling policy. Once the exploration phase is completed, the RSU now exploits PEARL's optimal scheduling policy during its recharge period. It is worthwhile to mention that PEARL learns an optimal scheduling policy for different traffic conditions, which is devised whenever the RSU is operating in an energy-limited mode.

This section is dedicated to present the various constituents of PEARL's properties and characteristics.

B. PEARL Operation

PEARL is a discrete-time learning protocol that develops an optimal scheduling policy of an energy-constrained RSU. At the beginning of each time slot, the vehicles residing within RSU's communication range broadcast beacon messages that the RSU collects in order to realize the network information corresponding to the number of vehicles within its range along with their respective speeds, locations, and remaining request size. The RSU then consults the PEARL agent in order to make a decision regarding which vehicle is granted service next. PEARL calculates each vehicle's weight according to the received information and, hence, establishes the observed system state. PEARL then engages in an exploration versus exploitation decision imposed by the ϵ-greedy method. In the exploration method, PEARL selects a random vehicle, which allows for the exploration of various system states. This operation is illustrated in Fig. 3. On the other hand, in the exploitation method, PEARL endorses the choice of the vehicle which mostly contributes to the maximization of the long-term system reward.

C. Finite Horizon MDP

This section lays out a precise definition of the PEARL's finite horizon MDP. The underlying Markov model, its characteristics, and its input data are presented in mathematical terms.

1) Preliminaries: Suppose that at time $t_n = 0$, G's battery is fully recharged, and its total power capacity is P_t. Also, assume that the time between two recharge periods is known, and the length of the discharge period is T seconds divided into N time slots, each of length τ. Let t_n be the beginning of the nth time slot where $t_n > 0$ and $n = \{1, 2, ..., N\}$. Let β_n be the number of vehicles residing within G's communication range at t_n, where $0 \leq \beta_n \leq \beta_{\max}$ and $\beta_{\max}$ is the maximum number of vehicles that can be present within the segment of length D_c at any time.

Let x_n be the state of the system at t_n. In the presented model herein, the system state is a vector whose elements represent the weights of the vehicles residing within G's communication range at t_n. Let w_i^n be an integer representing the weight of vehicle i at time t_n, where $n = \{1, 2, \ldots, \beta_n\}$ and $0 \leq w_i^n \leq w_{\max}$. w_i^n is a function of remaining residence time of vehicle i, remaining request size, and location along D_C, which indicates the vehicle's numerical ranking at time t_n. Note that selecting the value of $w_{\max}$ is remarkably strenuous as a large value of $w_{\max}$ poses a serious limitation to one's ability to solve the MDP accurately. This is known as the Bellman's *curse of dimensionality*, which is the well-known exponential increase in time and space required to compute an optimal solution to the MDP as the number of possible states (state space size) increases [18].

Now, it is important to mention that the number of vehicles present within the coverage range of G is not constant. In fact, since vehicles arrive to the RSU according to a Poisson process, then, according to [19], the number of vehicles present with the segment of length D_C follows the Poisson distribution. Hence, in order to avoid the complexity of having a variable vector size to represent the system state, we can justifiably assume that, at any arbitrary instant t_n at equilibrium, the number of vehicles present within the coverage range of the RSU is β_e. As such, the system state matrix at t_n is a vector of size β_e containing the corresponding weights of β_e vehicles. Note that, in the case where $\beta_e - \beta_n > 0$, the values of $\{w_{\beta_{n+1}}^n, w_{\beta_{n+2}}^n, \ldots, w_{\beta_e}^n\}$ will be set to $-\infty$. On the other hand, whenever $\beta_n > \beta_e$, the RSU will ignore the most recently arriving vehicles until either a vehicle completes its download request or a vehicle leaves its communication range. As such, the system state at time t_n is a fixed size vector denoted by $x_n = \{w_1^n, w_2^n, \ldots, w_{\beta_e}^n\}$.

2) Markov Decision Model: Consider the nonstationary Markov decision model with a planning horizon N consisting of the set of data (E, A, C_n, r_n, P_0) that can be defined as follows.

1) E is the state space where, at any time t_n, the system state $x_n \in E$.
2) A is the action space, where the action at time t_n is $a_n \in A$. Note that, $a_n = 1$ if the vehicle with associated weight w_1^n is selected, $a_n = 2$ if the vehicle with associated weight w_2^n is selected, and so forth. Therefore, $A = \{1, 2, \ldots, \beta_e\}$.
3) $C_n \subset E \times A$ is a measurable subset of $E \times A$ and denotes the set of possible state–action combinations at the beginning of the nth time slot [20]. C_n contains the graph of a measurable mapping, $g_n : E \to A$, i.e., $(x_n, g_n(x_n)) \in C_n$ for all $x_n \in E$. For $x_n \in E$, the set $C_n(x_n) = \{a_n \in A | (x_n, a_n) \in C_n\}$ is the set of valid (admissible) actions in state x_n at time t_n [20]. PEARL may only select a vehicle whose associated weight is not $-\infty$. Therefore, in the case where $\beta_e - \beta_n \geq 0$,

$C_n = \{1, 2, \ldots, \beta_n\}$, whereas when $\beta_e - \beta_n < 0$, $C_n = \{1, 2, \ldots, \beta_e\}$.

4) $r_n : C_n \rightarrow \mathbb{R}$ is a measurable function where $r_n(x_n, a_n)$ gives the single-step reward of the system at time t_n if the current state is x_n and action a_n is taken. According to PEARL, the single-step reward r_n is the number of downloaded bits to the selected vehicle at time t_n. Whenever a vehicle departs from *G*'s coverage range with an incomplete download request, the single-step reward is penalized by the remaining number of bits which need to be downloaded in order to fulfill that vehicle's request. As such, PEARL strives for a larger number of transmitted bits per time slot and, at the same time, tries to avoid the undesired event where a vehicle departs from *G*'s range with an incomplete download request. Note that, and according to [16], the RSU consuming the same amount of energy per time slot may transmit at larger rates for closer vehicles. Therefore, it becomes clear now that PEARL prefers to select vehicles closer to the RSU over the far ones. However, PEARL may be forced to schedule service for farther vehicles in order to prevent the penalty incurred by the above-described unfavorable event.

5) ϕ_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 semantics of ϕ_0 is the following: For $U_{n+1} \subset C_{n+1}$, $\phi_0(U_{n+1}|x_n, a_n)$ gives the probability that the next state and its associated reward belong to the set U_{n+1} provided that the current state is x_n and the action taken is a_n. The transition probability kernel ϕ_0 gives rise to the state transition probability kernel ϕ, which, for any $(x_n, a_n, x'_{n+1}) \in E \times A \times E$ 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.

3) Input From the Environment: At the beginning of each time slot, *G* collects all the parameters associated with the set of in range vehicles and then feeds the network information to PEARL, which, at time t_n, becomes aware of the following:

1) P_n being the remaining power in the RSU's battery, $0 \leq P_n \leq P_t$;
2) T_n being the time until the next recharge, $1 \leq T_n \leq T$;
3) β_n being the number of vehicles residing within *G*'s communication range, $0 \leq \beta_n \leq \beta_{\max}$;
4) $\overline{R_n} = \{R_1^n, R_2^n, \ldots, R_{\beta_e}^n\}$ being a vector of size β_e containing the remaining sojourn times of each vehicle v_i, $i \in (1, 2, \ldots \beta_e)$ and $0 \leq R_i^n \leq r_{\max}$;
5) $\overline{H_n} = \{H_1^n, H_2^n, \ldots, H_{\beta_e}^n\}$ being a vector of size β_e containing the remaining request sizes for each vehicle v_i, $0 \leq H_i^n \leq H_{\max}$;
6) $\overline{d_n} = \{d_1^n, d_2^n, \ldots, d_{\beta_e}^n\}$ being a vector of size β_e containing the distances between *G* and each of the in-range vehicles, $0 \leq d_i^n \leq G_R$, where $G_R = D_C/2$.

At this stage, and according to Fig. 3, PEARL will calculate the weight associated with each vehicle within *G*'s communication range and hence realize the system state.

4) Vehicle Weights: As earlier mentioned, the power required for *G* to communicate with a vehicle residing within its coverage range increases remarkably as the separation distance between the transmitter and the receiver increases. Therefore, in a greedy power saving mode, the RSU may always prefer to serve the closest vehicle in order to consume the least amount of energy and, thus, conserve its battery power for subsequent SRs. However, under such operational policy, vehicles may suffer from a deteriorated QoS, especially when leaving the communication range of the RSU without a completed SR. This event is referred to as an undesired in PEARL's formulation in the next sections.

Now, in order to bias the RSU toward serving vehicles with the least amount of consumed energy while avoiding undesired events, the weight of vehicle i at time t_n, previously defined as w_i^n, gives it a priority depending on its remaining sojourn time, remaining request size, and its separation distance from *G*. In fact, as a vehicle comes closer to *G* (i.e., d_i^n decreases), its weight increases since now the RSU may transmit data at a high rate rather than transmitting data to farther vehicles using the same amount of energy. Furthermore, whenever the remaining sojourn time R_i^n decreases, the weight of vehicle i increases as well, which is a signal for the RSU to complete the download request of vehicle i before it leaves its communication range (and avoid being penalized). Finally, the weight of vehicle i increases as H_i^n increases, which will prioritize vehicles with larger remaining request size.

Recall that the system state x_n at time t_n is a vector whose elements correspond to the weights of the set of in-range vehicles, where, as previously defined, $0 \leq w_i^n \leq w_{\max}$. Hence, the size of the state space E is $(w_{\max} + 1)^{\beta_e}$. It becomes clear now that a large value of $w_{\max}$ results in an intractable MDP whose state-space size is remarkably huge. On the other hand, a smaller value of $w_{\max}$ might not give PEARL enough information and differentiation between the different vehicles requesting service at a particular time slot. As such, the choice of the value of $w_{\max}$ has to account for the tradeoff between the time and space needed to achieve an optimal policy and the level of differentiation and prioritization between vehicles.

5) System Dynamics: This section lays out the following equations that govern the evolution of the system dynamics.

1) Power remaining in the next time slot:

$$P_{n+1} = P_n - P_c^n \tag{5}$$

where P_c^n is the power consumed by *G* in the nth time slot and $1 \leq n \leq N$.

2) A vehicle's remaining request size:

$$H_i^{n+1} = \begin{cases} H_i^n, & \text{if } a_n = i \\ H_i^n - K_i^n \times \tau, & \text{if } a_n \neq i \end{cases} \tag{6}$$

where $K_i^n = g(P_c^n, d_i^n)$ is the rate at which the RSU serves the selected vehicle. Note that, for a fixed amount of transmit power, the data rate decreases drastically as the separation distance between the transmitter and receiver increases.

3) A vehicle's remaining sojourn time:

$$R_i^{n+1} = R_i^n - \tau, \text{ for } i = \{1, 2, \ldots, \beta_n\}. \tag{7}$$

In the case where $\beta_n > \beta_e$, whenever a vehicle i either departs from *G*'s communication range or completes its download request during the time slot starting at t_n, *G* will consider the weight of another vehicle $j \neq i$ at t_{n+1}, where H_j^{n+1} and R_j^{n+1} are Random variables with known distributions.

D. MDP Solution Approach

1) Optimal Policy: The optimal control of an MDP requires the determination of a stationary policy π defining which action a_n should be applied at time t_n in order to maximize an aggregate objective function of the immediate costs. As such, a policy π induces a stationary mass distribution over the realizations of the stochastic process (x_n, a_n). A sequence of functions $\pi = \{a_1, a_2, \ldots, a_N\}$, with each $a_n : E \to A, 1 \leq n \leq N$, is said to be an admissible policy if $a_n \in A_n \; \forall x_n \in E$. Let Π denote the set of all admissible policies.

The goal of this study is to find an optimal policy $\pi*$ that will maximize the total discounted rewards over the discharge period. Whenever the RSU is following the scheduling policy π, the action at t_n is $a_n = \pi(x_n)$. Therefore, the single-step reward becomes $r_n = r(x_n, \pi(x_n))$. Let Ω be the total discounted sum of the single-step rewards given by

$$\Omega = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{N-1} r_N = \sum_{n=1}^{N} \gamma^{n-1} r_n. \quad (8)$$

Note that γ is a discount factor which is set between 0 and 1. Now, when following policy π, the single-step reward becomes $r_n = r(x_n, \pi(x_n))$, and as such, the total discounted sum of rewards becomes

$$\Omega_\pi = \sum_{n=1}^{N} \gamma^{n-1} r\left(x_n, \pi(x_n)\right). \quad (9)$$

The optimal policy is hence given by

$$\pi^* = \underset{\pi \in \Pi}{\text{argmax}} \;\; \Omega_\pi. \quad (10)$$

Recall that $\phi(x_{n+1}|x_n, \pi(x_n))$ is the probability of going from state x_n to state x_{n+1} when following policy π. Hence, according to [20], for a given admissible policy $\pi \in \Pi$, the value function $V^\pi : E \to \mathbb{R}$ satisfies the following Bellman equation:

$$V^\pi(x_n) = r(x_n, \pi(x_n)) + \gamma \sum_{x_{n+1}} \phi(x_{n+1}|x_n, \pi(x_n)) V^\pi(x_{n+1}) \quad (11)$$

for all $x_n \in E$. Therefore, it becomes clear now that the optimal policy π^* gives the optimal value function $V^* : E \to \mathbb{R}$ defined by

$$V^*(x_n) = \max_{\pi \in \Pi} \quad V^\pi(x_n). \quad (12)$$

According to [20], the optimal policy associated with the optimal value function given in (12) achieves the maximum reward expression laid out in (9). However, in order to solve (12), the knowledge of the transition probability function $\phi(x_{n+1}|x_n, \pi(x_n))$ is required. Note that the formulated Markovian domain herein lacks the state transition mapping, i.e., $\phi(x_{n+1}|x_n, a_n)$. Therefore, the RL method, *Q*-learning, presents itself as a simple way for the RSU to learn the optimal policy by experiencing the consequences of actions without the requirement of an established transition function.

Consequently, PEARL implements a stochastic iterative *Q*-learning algorithm and uses observations from online samples in order to realize the optimal scheduling policy π^*. Section IV-D-2 lays out the *Q*-learning algorithm used to solve (12).

2) Q-Learning Algorithm: *Q*-learning is a model-free RL technique which is widely used to find an optimal action selection policy for any given finite MDP. It works by learning an action-value function that eventually achieves the optimal reward of taking a given action in a given state and following the optimal policy thereafter. Define the optimal *Q*-values $Q^*(x_n, a_n)$, for all $(x_n, a_n) \in C_n$. According to [21], $Q^*(x_n, a_n)$ is given by

$$Q^*(x_n, a_n) = r_n(x_n, a_n) + \gamma \sum_{x_{n+1} \in E} \phi(x_{n+1}|x_n, \pi(x_n)) V^*(x_n). \quad (13)$$

Note that since $V^*(x_n) = \max_{a_n} \;\; Q^*(x_n, a_n)$, therefore the optimal policy $\pi^*(x_n)$ is given by

$$\pi^*(x_n) = \underset{a_n}{\text{argmax}} \;\; Q^*(x_n, a_n). \quad (14)$$

Since the *Q*-function makes the action explicit, the *Q*-values can be estimated using the following online incremental update stochastic *Q*-learning algorithm:

$$Q(x_n, a_n) := Q(x_n, a_n) + \alpha(n) \left[r(x_n, a_n) + \gamma \max_{a_{n+1}} Q(x_{n+1}, a_{n+1}) - Q(x_n, a_n) \right]. \quad (15)$$

Note that the choice of the value of γ 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. Furthermore, $\alpha(n)$ 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. The learning rate satisfies the following conditions:

$$\sum_n \alpha(n) = \infty$$

$$\sum_n \alpha(n)^2 < \infty. \quad (16)$$

The first condition ensures that the algorithm does not prematurely converge, whereas the second condition ensures that the noise in the algorithm asymptotically vanishes [17]. Most often, the step sizes are simply chosen to be $\alpha(n) = 1/n$. The convergence of the underlying algorithm has been widely analyzed and proven. The optimal *Q*-values, as well as the optimal policy, are obtained upon the convergence of the following algorithm. Note that several studies have addressed the problem of improving the convergence speed of *Q*-learning algorithms, where recommendations for faster convergence were empirically derived. For instance, the use of bootstrapping methods may accelerate the convergence of RL algorithms. However, bootstrapping requires *a priori* initial knowledge of the action-value function using supplied initial policies, which is outside the scope of this paper.

V. Simulation Results

A. Simulation Setup

In the simulation setup of this paper, the simple free-flow traffic model (SFTM), which was laid out in [14], is adopted. Using the discrete-event simulator Veins (see [22]), the vehicular traffic model presented in Section III is validated. Furthermore, the realistic mobility traces generated by Simulation for Urban MObility were fed as a mobility input for PEARL exploration

Algorithm 1: Q-Learning Method to Compute π^*.

```
For each $(x_n, a_n) \in C_n$, initialize $\mathbf{Q}(x_n, a_n) = 0$
Repeat (for each discharge period)
for all $\tau \in N$ do
    Observe $\beta_{\mathbf{n}}, \overline{R_n}, \overline{H_n}, \overline{d_n}$
    Evaluate $x_n$
    Select action $a_n$ using $\epsilon$-greedy method
    Execute $a_n$ and observe $\mathbf{r}(\mathbf{x_n}, \mathbf{a_n})$ and $x_{n+1}$
    Update $\mathbf{Q}(x_n, a_n)$ [according to (15)].
    $x_n \leftarrow x_{n+1}$
Until $\mathbf{Q}(x_n, a_n)$ converges $\forall \boldsymbol{(x_n, a_n) \in C_n}$
```

TABLE I
SIMULATION INPUT PARAMETERS

Parameter	Value
Discharge period	$T = 12$ (h)
Time slot length	$\tau = 0.1$ (ms)
Vehicular arrival rate	$\mu_V \in [0.1; 0.277]$ (veh/s)
Min and Max vehicle speed	$V_{\min} = 3$, $V_{\max} = 50$ (m/s)
Min and Max request size	$Q_{\min} = 1$, $Q_{\max} = 10$ (MB)
RSU covered segment	$D_C = 1000$ (m)
Channel data bit rate	$B_c \in [1; 27]$ (Mb/s)
Learning rate	$\alpha(n) = 1/n$
Discount factor	$\gamma = 0.5$

phase. The presented results herein were averaged over multiple runs of the simulations. In this section, the performance of PEARL is evaluated in terms of

1) incomplete request percentage;
2) collected rewards percentage;
3) average per-vehicle fulfilled request percentage;
4) network throughput.

PEARL is compared with three other scheduling algorithms, namely the following:

1) RVS: Random vehicle selection algorithm where, at time t_n, the RSU randomly chooses a vehicle $v_i \in S_n$ to be served [9];
2) GPC: Greedy power conservation algorithm where, at time t_n, the RSU chooses the vehicle $v_i \in S_n$ which resides in the lowest energy consumption zone compared with the remaining vehicles residing within *G*'s communication range.
3) RMS: Rate monotonic scheduling algorithm without preemption where, at time t_n, the RSU chooses the vehicle with the highest priority. A vehicle's priority, according to traditional RMS algorithms [23], is inversely proportional to its period, i.e., the shorter the period, the higher the priority and *vice versa*. Herein, a vehicle's period is the time it requires until it completes its download request.

Vehicular nodes arrive at a unidirectional highway segment of length D_C with multiple lanes according to a Poisson process and travel with a constant average speed drawn from a truncated Gaussian distribution. Each vehicle has an associated SR size to be downloaded from the RSU. A vehicle admitted to service may download at a rate of B_c Mb/s, depending on its corresponding separation distance with the RSU. Upon its departure from the RSU's communication segment, its associated remaining request size and average throughput are recorded for the network's performance analysis. Table I lists the simulator's input parameters.

B. Simulation Results

Fig. 4 evaluates PEARL's performance when compared with the three previously described scheduling algorithms, namely, RVS, GPC, and RMS. Fig. 4(a) plots the percentage of vehicles leaving *G*'s communication range with an incomplete SR as a function of the vehicular arrival rate. It is clear that the number of incomplete requests increases as the vehicular arrival rate increases under the three scheduling algorithms. In fact, an increase in μ_V is accompanied by an increase in the number of vehicles present within the range of the RSU. As μ_V 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 *G*'s communication range with incomplete SRs will increase. Fig. 4(a) also shows that PEARL outperforms RVS, GPC, as well as RMS in terms of incomplete SRs. Under RVS, the selection method is random, and no service differentiation is applied, and therefore, the number of vehicles whose associated download request is not fulfilled increases remarkably as more vehicles are present within *G*'s communication range. Now, for GPC, *G* is admitting to service the vehicle that resides in the minimal energy consumption zone compared with the set of in-range vehicles. Whenever μ_V is small and the vehicular density is low, a large portion of the vehicles have enough time to complete their download request whenever they are residing in low energy consumption zones; however, when μ_V increases, more vehicles will concurrently reside in low energy consumption zones and *G* will randomly choose between the multiple available vehicles, and therefore, the time during which a vehicle receives service is now not enough to complete the download request. Under RMS, the vehicle with the smallest remaining download request size is selected regardless of its location along D_C. Whenever the vehicular load is small, i.e., $\mu_V < 0.15$, RMS performs relatively well. However, as the network load increases, the number of incomplete requests increases remarkably. This is an expected result, knowing that the RMS is only a good scheduling algorithm whenever the requests are schedulable within a specific time frame. Finally, recall that, for PEARL, a vehicle departing *G*'s range with an incomplete SR is an undesired event which the agent is trained to avoid. Therefore, the deployment of the well-trained PEARL agent guarantees that the majority (more than 95%) of departing vehicles have completed their download SR. Fig. 4(b) plots the percentage of collected rewards, which is defined as the total number of downloaded bits over the total number of requested bits in a discharge period. In practice, a higher percentage of collected rewards results in a better QoS as well as increased RSU revenues. Fig. 4(b) shows that the portion of total downloaded bits over the total requested bits decreases as more vehicles are present within *G*'s communication range. Furthermore, it is clear that the adoption of the RL algorithm PEARL results in higher collected rewards than its counterparts RVS, GPC, and RMS. It is important to note that admitting the vehicle residing in the minimal energy consumption zone into service according to the GPC scheduling algorithm results in a high percentage of collected rewards since *G* is transmitting data to the selected vehicle at the highest achievable data rate at that particular instant. Consequently, and as previously stated, whenever the vehicular density is low, *G* is able to transmit a large portion of the requested file size. As μ_V increases, and more vehicles are present within the segment of length D_C, the smart scheduling algorithm PEARL outperforms GPC in terms

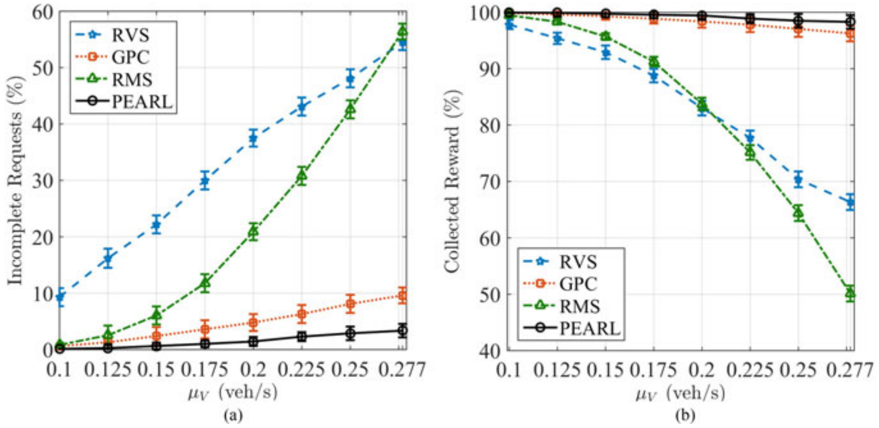

Fig. 4. PEARL performance evaluation. (a) Incomplete request percentage. (b) Collected reward percentage.

of the percentage of collected rewards as PEARL learns to wait for approaching vehicles to enter the lowest energy consumption zone and transmit their data at a rate of 27 Mb/s rather than admitting these vehicles to service when they are in higher energy consumption zones when no other vehicles are present in the lowest energy zones. Fig. 5 shows that the percentage of vehicles departing from RSU's coverage range with an incomplete SR increases as the average request size increases. The QoS also deteriorates as the vehicular arrival rate increases, which emphasizes the result in Fig. 4(a). It is clear that PEARL outperforms all the other scheduling benchmarks irrespective of the size of the average SR.

Fig. 6 plots the per-vehicle and the network throughputs when the RSU is operating under three different scheduling algorithms. Fig. 6(a) shows that the per-vehicle throughput deteriorates remarkably as μ_V increases under the RVS and RMS scheduling disciplines. This is expected since, under RVS, the RSU may choose a vehicle residing in high energy consumption zones, and the associated service transmission rate is therefore very small. Hence, the amount of time a vehicle spends receiving service, in this case, is extremely inefficient. In addition, under RMS, the vehicle whose remaining request size is smallest may highly likely be present in a high energy consumption zone. This means that *G* is transmitting at a low rate. Hence, under RMS, the per-vehicle throughput decreases quickly as more vehicles are present within *G*'s communication range. On the other hand, GPC and PEARL show significant enhancements on the level of per-vehicle throughput when compared with the RVS method. This is, in fact, due to the frequent selection of vehicles residing in low energy consumption zones allowing the RSU to transmit at a fast data rate. Fig. 6(a) also shows that PEARL tops GPC in terms of the per-vehicle throughput under all considered vehicular arrival rates. Following the same reasoning presented for Fig. 4(b), PEARL is trained to select the vehicle that most contributes to the total rewards, thus rendering the exploitation of the service time slot highly efficient.

Fig. 6(b) plots the overall network throughput under the four implemented scheduling algorithms for various free-flow vehicular arrival rates. It is clear that PEARL outsmarts the three scheduling disciplines RVS, GPC, and RMS in terms of the achievable network throughput. Under RVS, *G* randomly admits a vehicle residing within its communication range into service. As more vehicles are present within the considered roadway segment of length D_C, more vehicles are present in low energy consumption zones, and whose selection allows for faster data transmission. Hence, the network throughput increases as μ_V increases. Now, when the network is operating under GPC and the vehicular arrival rate is small, *G* is serving the vehicle residing in the lowest energy consumption zone compared with the set of all vehicles present within *G*'s communication range. In the very likely event that no vehicles reside in low energy consumption zones, *G* has no choice but to serve vehicles in high energy consumption zones, resulting in slow data transmission and, hence, decreased network throughput. However, as μ_V increases, GPC results in higher network throughput than that in RVS since now more vehicles are present within the segment of length D_C and *G* is serving vehicles in lower energy consumption zones, and hence transmitting data at a faster rate. Now, under RMS, when the vehicle arrival process is slow, the vehicles with smaller remaining request sizes are residing in low-to-medium energy consumption zones, which allows the RSU to transmit at an acceptable data rate. However, once more vehicles are present within the segment of length D_C, the vehicles having the smallest remaining file size fall at the edge of *G*'s communication range where the data download rate is smallest. Hence, the network throughput under RMS has this parabolic shape. When the RSU's operation is dictated by PEARL's optimal scheduling policy, the achieved network throughput is greater than that

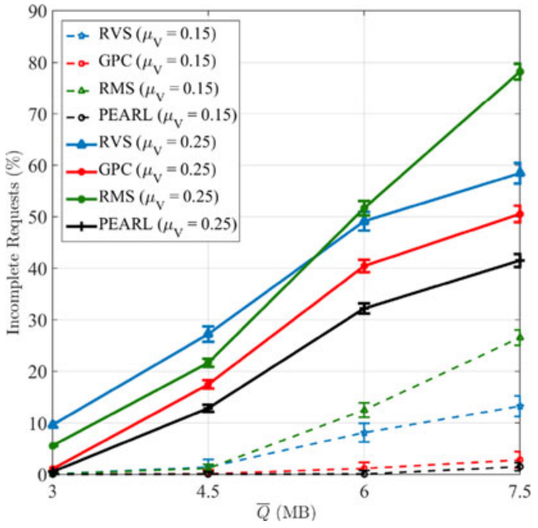

Fig. 5. QoS under variable request size.

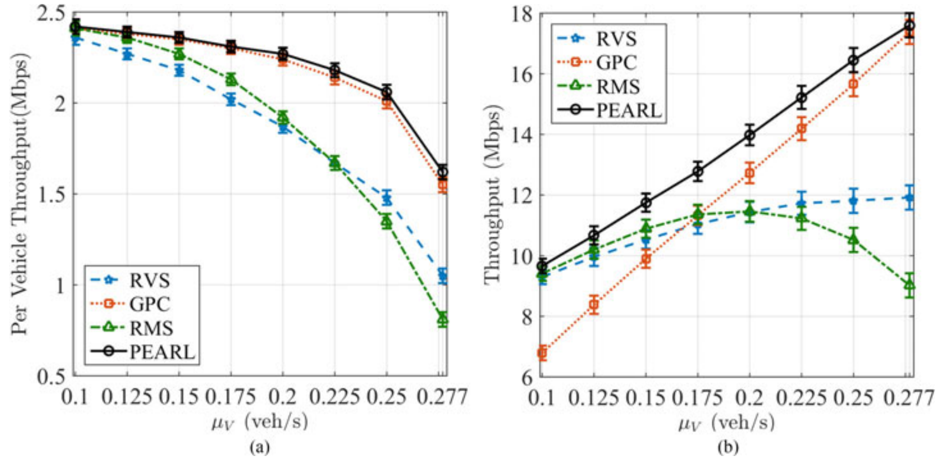

Fig. 6. Per-vehicle and network throughputs. (a) Per-vehicle throughput. (b) Network throughput.

achieved under RVS and GPC for all considered vehicular arrival rates. In fact, PEARL is trained to efficiently schedule the vehicle's download service in such a way that maximizes the number of downloaded bits per unit time. The result illustrated in Fig. 6(b) is an improved network throughput under PEARL compared with the other three scheduling algorithms.

VI. Conclusion and Future Research Direction

This paper addresses the problem of energy-limited RSUs in a vehicular network. A Markov decision process is formulated and solved using an RL technique, namely, the Q-learning algorithm. The resolution is PEARL, which is proposed for the purpose of increasing the number of downloaded bits per unit time, as well as avoiding the undesired event of a vehicle departing from the RSU's communication range with an incomplete SR. After a sufficient training period, PEARL exploits the realized optimal scheduling policy, which outperforms three benchmark scheduling algorithms in terms of several QoS metrics. In particular, the deployment of PEARL complements the RSU with the required intelligent identity, which serves to maintain the RSU's operation throughout the whole discharge period, as well as to decrease the number of vehicles departing from the RSU's coverage range with an incomplete SR.

References

[1] M. Kafsi *et al.*, "Vanet connectivity analysis," arXiv:0912.5527, 2009.

[2] A. Abdrabou and W. Zhuang, "Probabilistic delay control and road side unit placement for vehicular ad hoc networks with disrupted connectivity," *IEEE J. Select. Areas Commun.*, vol. 29, no. 1, pp. 129–139, Jan. 2011.

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

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

[5] D. Niyato and E. Hossain, "A unified framework for optimal wireless access for data streaming over vehicle-to-roadside communications," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 3025–3035, Jul. 2010.

[6] W. Tan, W. C. Lau, O. Yue, and T. H. Hui, "Analytical models and performance evaluation of drive-thru Internet systems," *IEEE J. Select. Areas Commun.*, vol. 29, no. 1, pp. 207–222, Jan. 2011.

[7] Y. Zhang, J. Zhao, and G. Cao, "On scheduling vehicle-roadside data access," in *Proc. 4th ACM Int. Workshop Veh. ad hoc Netw.*, 2007.

[8] 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.

[9] R. 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.

[10] Q. Ibrahim, "Design, implementation and optimisation of an energy harvesting system for vehicular ad hoc networks' road side units," *IET Intell. Transport Syst.*, vol. 8, no. 3, pp. 298–307, May 2014.

[11] 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.

[12] 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.

[13] F. Zou, *et al.*, "Energy-efficient roadside unit scheduling for maintaining connectivity in vehicle ad-hoc network," in *Proc. 5th Int. Conf. Ubiquitous Inf. Manage. Commun.*, 2011.

[14] M. 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.

[15] R. Roess, E. S. Prassas, and W. R. McShane, *Traffic Engineering*. Upper Saddle River, NJ, USA: Pearson, 2011.

[16] B. Alawieh, C. M. Assi, and H. Mouftah, "Investigating the performance of power-aware IEEE 802.11 in multihop wireless networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 287–300, Jan. 2009.

[17] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[18] W. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, vol. 703. Hoboken, NJ, USA: Wiley, 2007.

[19] 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.

[20] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.

[21] C. Szepesvári, *Algorithms for Reinforcement Learning* (Synthesis Lectures Artificial Intelligence and Machine Learning), vol. 4. San Rafael, CA, USA: Morgan & Claypool, 2010.

[22] 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.

[23] J. Lehoczky, L. Sha, and Y. Ding, "The rate monotonic scheduling algorithm: Exact characterization and average case behavior," in *Proc. Real Time Syst. Symp.*, 1989, pp. 166–171.

Authors' photographs and biographies not available at the time of publication