V2V Cooperative Sensing using Reinforcement Learning with Action Branching

Mohamed K. Abdel-Aziz*, Cristina Perfecto†, Sumudu Samarakoon*, and Mehdi Bennis*
*Centre for Wireless Communications, University of Oulu, Finland
†University of the Basque Country UPV/EHU, Spain
E-mails: {mohamed.abdelaziz, sumudu.samarakoon, mehdi.bennis}@oulu.fi, cristina.perfecto@ehu.eus

***Abstract*—Cooperative perception plays a vital role in extending a vehicle's sensing range beyond its line-of-sight. However, exchanging raw sensory data under limited communication resources is infeasible. Towards enabling an efficient cooperative perception, vehicles need to address fundamental questions such as: what sensory data needs to be shared? at which resolution? with which vehicles? In this view, this paper proposes a reinforcement learning (RL)-based vehicular association, resource block (RB) allocation, and content selection of cooperative perception messages by utilizing a quadtree-based point cloud compression mechanism. Simulation results show the ability of the RL agents to efficiently learn the vehicles' association, RB allocation and message content selection that maximizes the fulfillment of the vehicles in terms of the received sensory information.**

***Index Terms*—V2V, cooperative perception, reinforcement learning, quadtree**

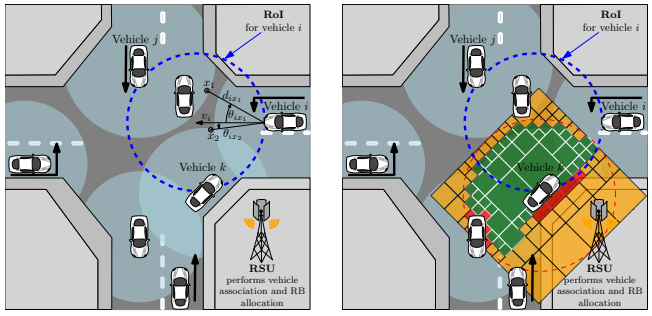
I. Introduction

Vehicles nowadays are equipped with a variety of sensors (e.g., RADAR, LiDAR, cameras) whose quality varies widely. These sensors enable a wide range of applications that assist and enhance the driving experience, from simple forward collision and lane change warnings, to the more advanced application of fully automated driving. However, the reliability of these sensors is susceptible to weather conditions, existence of many blind spots due to high density traffic or buildings, as well as sensors' manufacturing, and operating defects, all of which jeopardize these applications. In order to overcome this issue, the recent advancements of vehicle-to-vehicle (V2V) communication can be exploited. V2V communication is a promising facilitator for intelligent transportation systems [1]. It can facilitate the exchange of sensory information between vehicles to enhance the perception of the surrounding environment beyond their sensing range; such process is called *cooperative perception* [2]. The advantages of cooperative perception are validated in [3] demonstrating that it greatly improves the sensing performance. Motivated by its potential, several standardization bodies are currently focusing their efforts towards formally defining the cooperative perception message (CPM), its contents and generation rate [2], [4]. In addition, a growing body of literature has explored the use of cooperative perception in various ways [5]–[7]. In [5], the authors investigated which information should be included within the CPMs to enhance a vehicle's perception reliability. Cooperative perception from the sensor fusion point-of-view is tackled in [6] and a hybrid vehicular perception system that is able to fuse both local onboard sensor data as well as data received from a multi-access edge computing (MEC) server is proposed. Finally, the authors of [7] conducted a study focusing on raw-data level cooperative perception for enhancing the detection ability of self-driving systems; whereby sensory data collected from different positions and angles of connected vehicles is fused. Though interesting, neither of these works perform an in-depth analysis of the impact of wireless connectivity.

Cooperative perception over wireless networks cannot rely on exchanging raw sensory data, due to the limited communication resources availability [2]. Therefore, this raw sensory data should be compressed efficiently to save both the storage and the available communication resources. One possible technique that could be useful for such spatial raw sensory data is called *region quadtree* [8]. Region quadtree is a tree data structure used to efficiently store data on a two-dimensional space. A quadtree recursively decomposes the two-dimensional space into four equal sub-regions (blocks) till all the locations within a block have the same state or till reaching a maximum predefined resolution (tree-depth). Tailoring the number and resolution of the transmitted quadtree blocks to bandwidth availability is a challenging problem.

The main contribution of this paper is to study the joint problem of associating vehicles, allocating RBs and selecting the content of the exchanged CPMs, with the objective of maximizing the vehicles' satisfaction in terms of the received sensory information. Solving such problem using conventional mathematical tools is complex and intractable. As a result, we resort to machine learning techniques, specifically deep reinforcement learning (DRL), which proved to be useful in such complex situations [9]. In our paper, we split the main problem into two sub-problems formulated as RL problems, one is solved at a road-side unit (RSU) where the objective is to learn the association and RB allocation that maximizes the average vehicular satisfaction, while the other is solved by each vehicle with the objective of learning which quadtree blocks to transmit and at which resolution to maximize the associated vehicle's satisfaction. Simulation results show that the policies achieving higher vehicular satisfaction could be learned at both the RSU and vehicles level. It is also shown that trained agents always outperform non-trained random

This work was supported in part by the INFOTECH Project NOOR, in part by the NEGEIN project, by the EU-CHISTERA projects LeadingEdge and CONNECT, the EU-H2020 project IntellIoT under grant agreement No. 957218, and the Academy of Finland projects MISSION and SMARTER.



(a) (b)

Figure 1. (a) Vehicles under the coverage of a single RSU, drive through a junction while dynamically exchanging sensory information. (b) Quadtree representation of a vehicle's sensing range, with a maximum resolution level $L = 5$. Green represents the unoccupied state s_-, red represents the occupied state s_+ and orange represents the unknown state s_0.

agents in terms of the achieved vehicular satisfaction.

The rest of this paper is organized as follows. Section II presents the different parts of the system model. The network-wide problem is formulated in Section III, followed by our proposed RL solution within the cooperative sensing scenario, in Section IV. Finally, in Section V, simulation results are presented while conclusions are drawn in Section VI.

II. System Model

A single RSU providing a coverage to a road junction, as shown in Fig. 1, is considered. Let $\mathcal{N} \triangleq \{1, 2, \cdots, N\}$ denotes the set of vehicles served by the RSU. We denote the location of each vehicle $n \in \mathcal{N}$ at time slot t by $l_n(t)$. For the sake of simplicity and without loss of generality, we assume that each vehicle is equipped with a single sensor having a fixed circular range of radius r. Each location sensed by vehicle n can have one of three states: occupied (s_+), unoccupied (s_-), or unknown (s_0). This latter unknown state corresponds to blind-spots, due to occlusion, or to points beyond the limits of the vehicle's sensing range. As a result and due to the sensor's faults and uncertainties, the probability of occupancy at location $\boldsymbol{x}$ with respect to vehicle n is,

$$p_n(\boldsymbol{x}) = \begin{cases} \lambda_n & \text{if } s_n(\boldsymbol{x}) = s_+, \\ 1 - \lambda_n & \text{if } s_n(\boldsymbol{x}) = s_-, \\ 1/2 & \text{if } s_n(\boldsymbol{x}) = s_0, \end{cases} \tag{1}$$

where $s_n(\boldsymbol{x})$ is the state of location $\boldsymbol{x}$ defined by vehicle n, and $\lambda_n \in (0.5, 1]$ corresponds to the reliability of its sensor. Let $q_n(\boldsymbol{x})$ denotes the worthiness (quality) of the sensed information at location $\boldsymbol{x}$ that depends on the probability of occupancy $p_n(\boldsymbol{x})$, and the freshness of the sensed information, which can be quantified by the age of information (AoI) metric $\Delta_n(\boldsymbol{x})$ [10]. This worthiness is given by,

$$q_n(\boldsymbol{x}) = |2p_n(\boldsymbol{x}) - 1|\mu^{\Delta_n(\boldsymbol{x})}, \tag{2}$$

with a parameter $\mu \in (0, 1)$. Note that $q_n(\boldsymbol{x})$ decreases as its AoI increases (*outdated* information) or the probability of occupancy approaches 1/2 (*uncertain* information).

Moreover, vehicle n is interested in extending its sensing range by a duration of t_{int} seconds along its direction of movement which is captured by a circular region of interest (RoI). The RoI of vehicle n has a diameter of $v_n t_{\text{int}}$, where v_n is the velocity of the vehicle, as shown on Fig. 1(a). Within the RoI, the vehicle has higher interest in the locations closer to its current position as well as locations closer to its direction of movement over any other location. Therefore, the interest of vehicle n in a location $\boldsymbol{x}$ is formally defined as follows

$$w_n(\boldsymbol{x}) = \begin{cases} \frac{v_n t_{\text{int}} \cos\theta - d}{v_n t_{\text{int}} \cos\theta} & d \leq v_n t_{\text{int}} \cos\theta \\ 0 & \text{o.w.}, \end{cases} \tag{3}$$

where d is the euclidean distance between the location $\boldsymbol{x}$ and the vehicle's position $l_n(t)$, and θ is the angle between the vehicle's direction of motion and location $\boldsymbol{x}$, as illustrated on Fig. 1(a). To capture the need of gathering new information, the interest $w_n(\boldsymbol{x})$ of vehicle n needs to be weighted based on the lack of worthy information, i.e., $1 - q_n(\boldsymbol{x})$. Hence, the modified interest of vehicle n in location $\boldsymbol{x}$ is given by,

$$i_n(\boldsymbol{x}) = w_n(\boldsymbol{x})[1 - q_n(\boldsymbol{x})]. \tag{4}$$

Furthermore, a time-slotted communication with transmission slots of duration τ is considered, where each vehicle is allowed to exchange sensory information with at most one vehicle at each time slot. We define $E(t) = [e_{nn'}(t)]$ to be the global association matrix, where $e_{nn'}(t) = 1$ if vehicle n is associated (transmits) to vehicle n' at time slot t, otherwise, $e_{nn'}(t) = 0$. The association is assumed to be bi-directional, i.e., $e_{nn'}(t) = e_{n'n}(t)$. Moreover, we assume that each associated pair can communicate simultaneously with each other, i.e. each vehicle is equipped with two radios, one for transmitting and other is for receiving. Additionally, a set $\mathcal{K} \triangleq \{1, 2, \cdots, K\}$ of orthogonal resource blocks (RBs), with bandwidth ω per RB, is shared among the vehicles, where each radio is allocated with only one RB. We further denote the RB usage as $\eta^k_{nn'}(t) \in \{0, 1\}$, for all $k \in \mathcal{K}$ and $n, n' \in \mathcal{N}$, in which $\eta^k_{nn'}(t) = 1$ if vehicle n transmits over RB k to vehicle n' on time slot t and $\eta^k_{nn'}(t) = 0$, otherwise.

Let $h^k_{nn'}(t)$ be the instantaneous channel gain, including path loss and channel fading, from vehicle n to vehicle n' over RB k in slot t. We consider the 5.9 GHz carrier frequency and adopt the realistic V2V channel model of [11] in which, depending on the location of the vehicles, the channel model is categorized into: Line-of-sight, weak-line-of-sight, and non-line-of-sight. Thus, the data rate from vehicle n to vehicle n' on time slot t (in packets per slot) is expressed as

$$R_{nn'}(t) = \frac{\tau}{M} \sum_{k \in \mathcal{K}} \eta^k_{nn'}(t) \omega \log_2 \left(1 + \frac{P h^k_{nn'}(t)}{N_0 \omega + I^k_{nn'}(t)}\right), \tag{5}$$

where M is the packet length in bits, P is the transmission power per RB, and N_0 is the power spectral density of the additive white Gaussian noise. Here, $I^k_{nn'}(t) = \sum_{i,j \in \mathcal{N}/n,n'} \eta^k_{i,j}(t) P h^k_{in'}(t)$ indicates the received aggregate interference at the receiver n' over RB k from other vehicles transmitting over the same RB k.

Exchanging raw sensory information between vehicles about individual locations $\boldsymbol{x}$, would require huge communication resources for cooperative perception to be deemed useful. As a result, the region quadtree compression technique is utilized by each vehicle [8]. Within this technique, each vehicle converts its sensing range into a squared-block of side-length $2r$. This block is divided recursively into 4 blocks until reaching a maximum resolution level L or until the state of every location $\boldsymbol{x}$ within a block is the same. Without loss of generality, we assume that each block can be represented using M bits. Fig. 1(b) shows the quadtree representation of the sensing range of a vehicle with $L = 5$.

The state of a block b within the quadtree of vehicle n is said to be occupied if the state of any location $\boldsymbol{x}$ within the block is occupied, while the state of a block is said to be unoccupied if every location within the block is unoccupied. Otherwise, the block would have an unknown state. Let $\mathcal{B}_n(t)$ represents the set of quadtree blocks available for transmission by vehicle n at time slot t. For simplicity and without loss of generality, we assume that $\mathcal{B}_n(t)$ only contain blocks available from its own sensing range. Due to the quadtree compression, the cardinality of $\mathcal{B}_n(t)$ is upper bounded by: $|\mathcal{B}_n(t)| \leq \sum_{l=0}^{L-1} 4^l = \frac{1-4^L}{1-4}$.

III. Problem formulation

Each vehicle n is interested in associating (pairing) with another vehicle n' where each pair exchanges sensory information in the form of quadtree blocks with the objective of maximizing the joint satisfaction of both vehicles. The satisfaction of vehicle n with the sensory information received from vehicle n' at time slot t can be defined as follows:

$$f_{nn'}(t) = \sum_{b \in \mathcal{B}_{n'}(t)} \sigma^b_{n'}(t) \left(\frac{\sum_{\boldsymbol{x} \in b} i_n(\boldsymbol{x})}{\mathrm{Ar}(b)} . q_{n'}(b) \right), \quad (6)$$

where $\sigma^b_{n'}(t) = 1$ if vehicle n' transmitted block b at time slot t, and $\sigma^b_{n'}(t) = 0$ otherwise, and $\mathrm{Ar}(b)$ is the area covered by block b. Note that, vehicle n is more satisfied when receiving quadtree blocks with a resolution proportional to the weights of its RoI as per (4), which is captured by $\frac{\sum_{\boldsymbol{x} \in b} i_n(\boldsymbol{x})}{\mathrm{Ar}(b)}$. Furthermore, vehicle n is more satisfied when receiving quadtree blocks having more worthy sensory information, which is captured by $q_{n'}(b)$. As a result, our cooperative perception network-wide problem can be formulated as follows:

$$\max_{\boldsymbol{\eta}(t), E(t), \boldsymbol{\sigma}(t)} \quad \sum_{n, n' \in \mathcal{N}} f_{nn'}(t) \cdot f_{n'n}(t)$$

$$\text{s.t.} \quad \sum_{b \in \mathcal{B}_n(t)} \sigma^b_n(t) \leq \sum_{n' \in \mathcal{N}} R_{nn'}(t), \ \forall n \in \mathcal{N}, \forall t \quad (7a)$$

$$\sum_{n' \in \mathcal{N}} \sum_{k \in \mathcal{K}} \eta^k_{nn'}(t) \leq 1, \ \forall n \in \mathcal{N}, \forall t \quad (7b)$$

$$\sum_{n' \in \mathcal{N}} e_{nn'}(t) \leq 1, \ \forall n \in \mathcal{N}, \forall t \quad (7c)$$

$$e_{nn'}(t) = e_{n'n}(t), \ \forall n, n' \in \mathcal{N}, \forall t \quad (7d)$$

where the objective is to associate vehicles $E(t)$, allocate RBs $\boldsymbol{\eta}(t)$, and select the contents of the transmitted messages $\boldsymbol{\sigma}(t)$, in order to maximize the sum of the joint satisfaction of the associated vehicular pairs. Note that (7a) upper bounds the number of transmitted quadtree blocks of each vehicle by its Shannon data rate, while (7b) constrains the number of RBs allocated to each vehicle by 1 RB. Finding the optimal solution of this problem is complex and not straight-forward because it would require the frequent exchange of fast-varying information between the RSU and vehicles, yielding a huge communication overhead which is impractical. Hence, to solve (7) practically, we leverage the machine learning techniques which have proved themselves to be useful to deal with such complex situations, specifically deep reinforcement learning (DRL) techniques [9].

IV. RL in cooperative sensing

In order to solve (7), the timeline is splitted into two scales, a coarse scale called time frames and a fine scale called time slots. At the beginning of each time frame, the RSU associates vehicles into pairs and allocates RBs to those pairs. The association and RB allocation stay fixed during the whole frame which consists of X time slots. At the beginning of each time slot t, each vehicle selects the quadtree blocks to be transmitted to its associated vehicle. By utilizing RL[1] within this cooperative sensing scenario, we can formulate two different RL problems: Vehicular RL and RSU RL.

A. Vehicular RL

Within this RL problem, for a given association nn' and RB allocation, each vehicle n acts as an RL-agent who wants to learn which quadtree blocks to transmit to its associated vehicle n' in order to maximize vehicle's n' satisfaction. Accordingly, the *global state* of the RL environment is defined as $\langle \mathcal{B}_n(t), \mathcal{I}_{n'}(t), v_n, v_{n'}, l_n(t), l_{n'}(t) \rangle$, where $\mathcal{I}_{n'}(t)$ is the set of vehicle's n' RoI weights, as per (4), at time slot t. However, this global state cannot be observed by vehicle n, instead, the *local observation* of vehicle n is $\langle \mathcal{B}_n(t), v_n, v_{n'}, l_n(t), l_{n'}(t) \rangle$. At every time slot t and by utilizing this local observation, vehicle n would take an action $\boldsymbol{\sigma}_n(t)$, selecting which quadtree blocks to be transmitted to its associated vehicle n', and receives a feedback (*reward*) from vehicle n' equal to $f_{n'n}(t)$. In a nutshell, the elements of the RL problem at each vehicle n can be described as follows:

- Global state: $\langle \mathcal{B}_n(t), \mathcal{I}_{n'}(t), v_n, v_{n'}, l_n(t), l_{n'}(t) \rangle$
- Local observation: $\langle \mathcal{B}_n(t), v_n, v_{n'}, l_n(t), l_{n'}(t) \rangle$
- Action: $\boldsymbol{\sigma}_n(t)$
- Reward: $f_{n'n}(t)$

B. RSU RL

Within this RL problem, the RSU acts as the RL-agent where the *state* of this RL environment is given by the locations and velocities of all vehicles serviced by the RSU, $\langle v_n, l_n \, \forall n \in \mathcal{N} \rangle$. Based on this state at the beginning of each

[1]For detailed information regarding RL, please refer to [12].

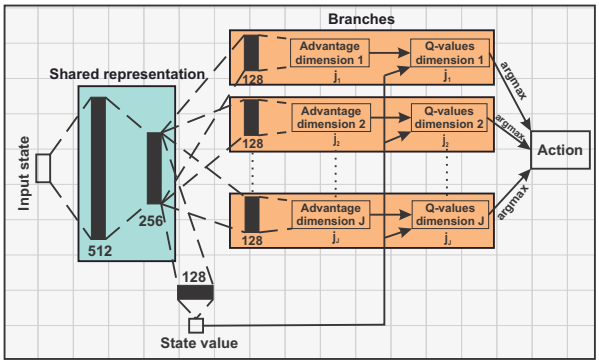


Figure 2. The BDQ neural network architecture utilized for both RSU and vehicular RL problems.

time frame, the RSU takes the *action* of vehicles association $E(t)$, and RB allocation $\boldsymbol{\eta}(t)$. Then, once the time frame ends, each vehicle will report back its mean satisfaction during the whole frame and the RL *reward* is computed as the mean of those feedbacks. In a nutshell, the elements of the RSU RL problem can be summarized as follows:

- State: $\langle v_n, l_n \, \forall n \in \mathcal{N} \rangle$
- Action: $E(t)$ and $\boldsymbol{\eta}(t)$
- Reward: $\frac{\sum_{n \in \mathcal{N}} \left(\sum_{t=i}^{i+X} f_{nn'(t)} \right) / X}{|\mathcal{N}|}$

In order to solve these two RL problems, the deep Q-Network (DQN) algorithm [9] can be utilized. Within DQN, the Q-value for each possible action should be estimated before deciding which action to take. As a result, its application to high dimensional, discrete action spaces is still arduous. At this point, it should be noted that our two RL problems suffer from the high dimensionality of action spaces. Specifically, within the RSU RL problem, the RSU needs to select $E(t)$ and $\boldsymbol{\eta}(t)$: The association matrix $E(t)$ is of size $N \times N$, and due to our one-to-one association assumption, the number of possible actions for the association problem would be $\Pi_{n=1}^{\lfloor N/2 \rfloor} (2n-1)$. Moreover, the RB allocation matrix $\boldsymbol{\eta}(t)$ is of size $N \times K$, as a result, the number of possible actions is K^N, assuming that each vehicle is allocated only 1 RB. Similarly, within the vehicular RL problem, each vehicle needs to select $\boldsymbol{\sigma}_n(t)$ whose dimension is $|\mathcal{B}_n|_{\max} \times 1$, yielding a total number of possible actions equal to $2^{|\mathcal{B}_n|_{\max}}$. These huge number of actions can seriously affect the learning behavior of DQN.

Recently, the authors in [13] have introduced a novel agent called Branching Dueling Q-Network (BDQ) leading to a novel neural network architecture that allows to distribute the representation of the action dimensions across individual network branches while maintaining a shared module that encodes a latent representation of the input state and helps to coordinate the branches. Remarkably, this neural network architecture exhibits a linear growth of the network outputs with increasing action space as opposed to the combinatorial growth experienced in traditional DQN network architectures. Fig. 2 demonstrates this neural network architecture.

In this work, we adopt these BDQ agents from [13] within our RL problems. As a result, the neural network at the RSU agent will have N branches[2] constructed as follows:

- $\lfloor N/2 \rfloor$ branches corresponding to the association action with each branch having $j_i = N - 2i + 1$ sub-actions, where i is the branch ID. For example, let us consider a simplified scenario with $N = 6$, then $\lfloor N/2 \rfloor = 3$ vehicular pairs could be formed: the first branch representing the first vehicle would have $N - 2 \cdot (1) + 1 = 5$ candidate vehicles to pair with, while for the second branch the candidates are reduced to 3 and so on. This leads to a unique vehicular association for any combination of sub-actions selected at each of the branches.
- $\lfloor N/2 \rfloor$ branches corresponding to the RB allocation with each branch having $\binom{K}{2}$ sub-actions, knowing that each associated pair is allocated 2 orthogonal RBs (one RB for each vehicle).

The aftermath of using the BDQ agent is that, to select an association action $E(t)$, the Q-value is only estimated for $\sum_{n=1}^{\lfloor N/2 \rfloor} (2n-1)$ actions instead of for $\Pi_{n=1}^{\lfloor N/2 \rfloor} (2n-1)$ actions with a non-branching network architecture. Similarly, selecting an RB allocation $\boldsymbol{\eta}(t)$, requires the Q-value estimation of $\frac{N}{2} \times \binom{K}{2}$ actions instead of the $\binom{K}{2}^{N/2}$ values involved in a traditional DQN architecture. Equivalently, by utilizing the BDQ agent within our vehicular RL problem, for the message content selection $\boldsymbol{\sigma}_n(t)$, the Q-value is estimated for $2 \times |\mathcal{B}_n|_{\max}$ actions only instead of $2^{|\mathcal{B}_n|_{\max}}$ actions.

For the RSU and vehicular agents training purposes, DQN [9] is selected as the algorithmic basis[3]. The detailed training algorithm is shown in Algorithm 1. Note that the loss function used for training any of the agents is as follows [13]:

$$L(\phi) = \mathbb{E}_{(s,a,r,s') \sim U(D)} \left[\frac{1}{J} \sum_i \left(y_i - Q_i(s, a_i) \right)^2 \right],$$

where for an action dimension $i \in \{1, \ldots, J\}$ with $|\mathcal{A}_i| = j_i$ discrete sub-actions, the individual branch's Q-value at state $s \in \mathcal{S}$ and sub-action $a_i \in \mathcal{A}_i$ is expressed in terms of the common state value $V(s)$ and the corresponding state-dependent sub-action advantage $A_i(s, a_i')$ by $Q_i(s, a_i) = V(s) + \left(A_i(s, a_i) - \frac{1}{j_i} \sum_{a_i' \in \mathcal{A}_i} A_i(s, a_i') \right)$. Moreover, $y_i = r + \gamma \frac{1}{J} \sum_i Q_i^- \left(s', \arg\max_{a_i' \in \mathcal{A}_i} Q_i(s', a_i') \right)$ is the temporal difference targets, with a discount factor of γ[4].

V. Simulation Results and Analysis

In this section, simulations are conducted based on practical traffic data to demonstrate the effectiveness of the proposed approach. A traffic light regulated junction scenario is considered. The scenario contains vehicles of different dimensions to mimic assorted cars, buses and trucks. The vehicles' mobility traces are generated using the simulation of urban mobility (SUMO) application [14]. Unless stated otherwise, the simulation parameters are listed in Table I.

[2] $N - 1$ branches if N is odd.

[3] DQN is selected for its simplicity, powerfulness, and off-policy algorithm.

[4] For more details on the choice of the loss function and its components, please refer to [13].

Algorithm 1 Training a BDQ agent for cooperative sensing

Initialize the replay memory of each agent to capacity C.
Initialize each agent's neural network with random weights ϕ.
Initialize each agent's target neural network with weights $\phi^- = \phi$.
foreach RSU episode **do**
  Reset the RSU environment by selecting a random trajectories for vehicles within the junction scenario.
  The RSU observes its current state $\langle v_n, l_n \, \forall n \in \mathcal{N} \rangle$
  foreach Z frames **do**
    With probability ϵ, the RSU agent selects a random association and RB allocation action, otherwise it selects the action with maximum Q-value.
    The RSU transmits its decision to the corresponding vehicles.
    foreach X slots **do**
      Each vehicle n computes its local observation $\langle \mathcal{B}_n(t), v_n, v_{n'}, l_n(t), l_{n'}(t) \rangle$.
      With probability ϵ, each vehicle's agent selects random sensory blocks to be transmitted to its associated vehicle, otherwise it selects the sensory blocks with maximum Q-value.
      The selected sensory blocks is transmitted over the allocated RB to the associated vehicle which only receives a random subset of these blocks depending on the data rate $R_{nn'}(t)$ as per (5).
      Each vehicle n calculates its own satisfaction $f_{nn'}(t)$ with the received blocks and feeds it back as a reward to its associated vehicle.
      Each vehicle n receives the reward, observes the next local observation and stores this experience (s_t, a_t, r_t, s_{t+1}) in its replay memory.
      if vehicle n collected a sufficient amount of experiences **do**
        Vehicle n samples uniformly a random mini-batch of experiences e^n from its replay memory.
        Using these samples, a gradient decent step is performed on $L(\phi)$ w.r.t. ϕ.
      end **if**
    end **for**
    Each vehicle feeds back its average received reward during the whole frame to the RSU.
    The RSU calculates the mean of all the received feedbacks and use the result as its own reward.
    The RSU stores its own experience, (s_i, a_i, r_i, s_{i+1}), in its replay memory.
    if the RSU collected a sufficient amount of experiences **do**
      Sample uniformly a random mini-batch of experiences e^{RSU} from its replay memory.
      Using these samples, a gradient decent step is performed on $L(\phi)$ w.r.t. ϕ.
    end **if**
  end **for**
end **for**

Table I
SIMULATION PARAMETERS.

Parameter	Value	Parameter	Value
K	10	N_0	-174 dBm/Hz
ω	180 KHz	P	10 dBm
τ	2 ms	t_{int}	2 sec
M	100 Byte	L	5
λ_n	1	r	20
X	5 slots	Z	10 frames

Moreover, the hyperparameters used for training the RSU and vehicular agents are discussed next. Common to all agents, training always started after the first 1000 steps, after which one step of training is run at every time step. Adam optimizer was used with a learning rate of 10^{-4}. Training

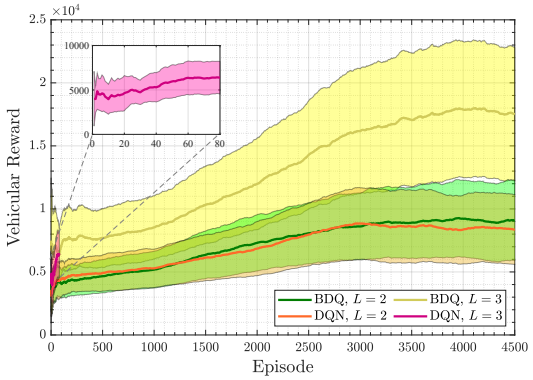


Figure 3. Learning curves for the vehicular RL environment using BDQ vs DQN agents, for different L. The solid lines represent the average over all the vehicles, where the learning curve of each vehicle is smoothed by the moving average over a window size of 1000 episodes, while the shaded areas show the 90% confidence interval over the vehicles.

was done with a mini-batch size of 64 and a discount factor $\gamma = 0.99$. The target network is updated every 1000 time steps. A rectified non-linearity (ReLU) is used for all hidden layers and a linear activation is used on the output layers, for all the neural networks. Each neural network had two hidden layers with 512 and 256 units in the shared network module and one hidden layer per branch with 128 units. Finally, a buffer size of 10^6 is set to the replay memory of each agent.

First of all, we verify that the BDQ agent can deal with the huge action space problem without any performance degradation compared to the classical DQN agent. For this purpose, we vary the size of the action space of the vehicular RL problem by varying the maximum quadtree resolution L. Note that, when $L = 2$, the maximum number of blocks available is $\frac{1-4^L}{1-4} = 5$, resulting in a total number of actions of $2^5 = 32$, and when $L = 3$, the maximum number of blocks available is 21, leading to a total number of actions of $2^{21} \approx 2{\times}10^6$. Fig. 3 shows the learning curve of both BDQ and DQN agents, for each case of L. When $L = 2$ (small action space), the learning curve of both the BDQ and DQN agents are comparable and they learn with the same rate. However, when L increases to 3 (huge action space), the training process of the DQN agent could not be completed because it is computationally expensive. This is due to the huge number of actions that need to be explicitly represented by the DQN network and hence, the extreme number of network parameters that need to be trained every iteration. On the other hand, the BDQ agent performs well with robustness against the huge action space, which demonstrates the suitability of BDQ agents to overcome the frequent scalability problems faced by other forms of RL.

Next, in Fig. 4 we focus on the training dynamics of the RSU agent for different N. The results show that the RSU reward increases gradually with increasing the number of episodes, meaning that the RSU and vehicles learn a better association, RB allocation and message content selection over the training period. However, it can be noted that the rate

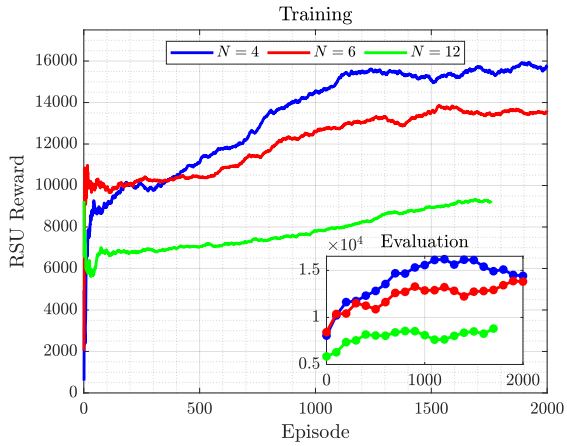


Figure 4. Training/Evaluation curves of the RSU agent for different N. Each curve is smoothed by the moving average over a window size of 500 episodes.

of increase of the RSU reward decreases as the number of served vehicles N increases and hence, more episodes are required to reach the same performance. The reason lays on the increase in the state space of the RSU agent experimented as N increases, which ultimately requires more episodes to be explored/discovered. Moreover, Evaluations were conducted every 100 episodes of training for 10 episodes with a greedy policy. The progress of the evaluation process during training is shown in Fig. 4. It verifies that agents learn better policies along the training duration.

Finally, after obtaining the trained RSU and vehicular agents, we deploy those trained agents within a newly generated vehicular mobility trajectory scenario that runs for 20000 slots. Fig. 5 shows the complementary cumulative distribution function (CCDF) of the vehicular rewards of all the vehicles for different N for two cases: the case of trained agents and the non-trained agents case, that randomly selects its actions. Note that, the trained agents achieve a better vehicular reward distribution both for $N = 4$ and $N = 6$. This proves that RL has taught the RSU and vehicular agents to take better actions for association, RB allocation and message content selection, to maximize the achieved vehicular satisfaction with the received sensory information.

VI. Conclusion

In this paper, we have studied the problem of associating vehicles, allocating RBs and selecting the contents of CPMs in order to maximize the vehicles' satisfaction in terms of the received sensory information while considering the impact of the wireless communication. To solve this problem, we have resorted to the DRL techniques where two RL problems have been modeled. In order to overcome the huge action space inherent to the formulation of our RL problems, we applied the dueling and branching concepts. Simulation results show that policies achieving higher vehicular satisfaction could be learned at both the RSU and vehicular levels leading to a higher vehicular satisfaction.

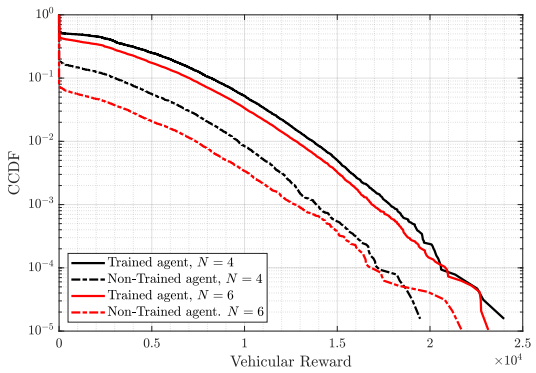


Figure 5. The CCDF of the vehicular reward achieved by trained and non-trained agents for different N.

References

[1] J. Park, S. Samarakoon, H. Shiri, M. K. Abdel-Aziz, T. Nishio, A. Elgabli, and M. Bennis, "Extreme urllc: Vision, challenges, and key enablers," 2020. [Online]. Available: https://arxiv.org/pdf/2001.09683

[2] ETSI TR 103 562 V2.1.1, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2," Dec. 2019.

[3] Y. Wang, G. de Veciana, T. Shimizu, and H. Lu, "Performance and scaling of collaborative sensing and networking for automated driving applications," in *Proc. of IEEE International Conference on Communications Workshops*, Kansas City, MO, USA, May 2018, pp. 1–6.

[4] 3GPP TR 22.886 V16.2.0 , "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on enhancement of 3GPP Support for 5G V2X Services (Release 16)," Dec. 2018.

[5] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Generation of cooperative perception messages for connected and automated vehicles," *IEEE Transactions on Vehicular Technology*, pp. 1–1, Nov. 2020.

[6] M. Gabb, H. Digel, T. Müller, and R.-W. Henn, "Infrastructure-supported perception and track-level fusion using edge computing," in *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, Paris, France, Jun. 2019, pp. 1739–1745.

[7] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3D point clouds," in *Proc. of IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, Jul. 2019, pp. 514–524.

[8] H. Samet, "The quadtree and related hierarchical data structures," *ACM Comput. Surv.*, vol. 16, no. 2, pp. 187–260, Jun. 1984. [Online]. Available: https://doi.org/10.1145/356924.356930

[9] V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb 2015.

[10] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Proc. of 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Salt Lake City, UT, USA, Jun. 2011, pp. 350–358.

[11] T. Mangel, O. Klemp, and H. Hartenstein, "A validated 5.9 GHz non-line-of-sight path-loss and fading model for inter-vehicle communication," in *2011 11th International Conference on ITS Telecommunications*, St. Petersburg, Russia, Aug. 2011, pp. 75–80.

[12] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.

[13] A. Tavakoli, F. Pardo, and P. Kormushev, "Action branching architectures for deep reinforcement learning," *CoRR*, vol. abs/1711.08946, 2017. [Online]. Available: http://arxiv.org/abs/1711.08946

[14] P. A. Lopez, M. Behrisch, L. Bieker-Walz *et al.*, "Microscopic traffic simulation using sumo," in *Proc. of 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 2575–2582.