Maximizing Information Usefulness in Vehicular CP Networks Using Actor-Critic Reinforcement Learning

Imed Ghnaya
Univ. Bordeaux, Bordeaux INP
CNRS, LaBRI, UMR5800
F-33400 - Talence, France
imed.ghnaya@u-bordeaux.fr

Toufik Ahmed
Univ. Bordeaux, Bordeaux INP
CNRS, LaBRI, UMR5800
F-33400 - Talence, France
tad@labri.fr

Mohamed Mosbah
Univ. Bordeaux, Bordeaux INP
CNRS, LaBRI, UMR5800
F-33400 - Talence, France
mohamed.mosbah @u-bordeaux.fr

Hasnaa Aniss
Gustave Eiffel University
COSYS-ERENA Lab
F-33067 – Bordeaux, France
hasnaa.aniss@univ-eiffel.fr

Abstract—Cooperative Perception (CP) allows Connected and Autonomous Vehicles (CAVs) to enhance their Environmental Awareness (EA) by sharing locally perceived objects through CP messages (CPMs). European Telecommunications Standards Institute (ETSI) has recently defined a set of CPM generation rules to achieve a trade-off between EA and Channel Busy Ratio (CBR) despite massive perception data. Nonetheless, these rules still lack the consideration of information usefulness, resulting in a considerable volume of useless information transmitted in the CP network. This limitation could increase CBR and thus decrease EA due to the loss of CPMs in the network. This paper introduces CloudAC-IU, a cloud-based deep reinforcement learning approach to lean CAVs to maximize perception information usefulness in the network. Simulation results highlight that the CloudAC-IU enhances EA by decreasing CBR and increasing CPM reception for CAVs compared to state-of-the-art works.

Keywords—connected and autonomous vehicles, V2X communications, cooperative perception, reinforcement learning, advantage actor-critic.

I. Introduction

Vehicles gather information using onboard sensors such as radars, lidars, and cameras to perceive the road environment. However, the perception capabilities of these sensors are limited to only visible objects in their Field-of-Views (FoVs) due to the occlusions caused by other users and buildings. Consequently, this limitation leads to a poor perception of the surrounding environment [1]. Alternatively, vehicle-to-everything (V2X) communications have emerged as a feasible solution to overcome this limitation. V2X enables Connected and Autonomous Vehicles (CAVs) to exchange information using wireless communication technologies, such as the European Telecommunications Standards Institute (ETSI) ITS-G5 [2]. In this context, Cooperative Perception (CP) [3] represents a new paradigm employed to improve road safety and increase Environmental Awareness (EA). This is accomplished by enabling CAVs to exchange perceived objects with other CAVs using Cooperative Perception Messages (CPMs). However, as the same object might appear simultaneously in the FoV of multiple CAVs, the CP may result in significant transmission redundancy due to the large amounts of perception data. This unnecessary exchange could increase the channel load and decrease EA for CAVs.

Recently, several kinds of research have demonstrated that paradigms such as supervised, unsupervised, and reinforcement learning (RL) provide better results in Cooperative Intelligent Transport Systems (C-ITS) [4]. In particular, RL in our context represents a powerful paradigm that learns a CAV to select and broadcast useful objects in CPMs based on the state of the environment in order to maximize a reward value such as usefulness. Although traditional RL is often used to fit simple models, it suffers from significant limitations in terms of scalability and performance when applied to complex tasks. On the other hand, Deep Neural Networks (DNNs) coupled with RL, called Deep RL (DRL), have been proved to be an excellent alternative for boosting the learning capacity of RL systems [5]. However, given the complexity of the driving environment, a significant issue may arise when dealing with large state and action spaces where the exploration process seems impossible. In this regard, [6] has introduced new policy-based RL approaches, such as actor-critic, that gradually fit and evaluate a strategy without exploring the whole action space.

In this paper, a cloud-based DRL approach called CloudAC-IU is proposed. Its main goal is to lean CAVs to exchange only perceived objects that maximize the benefit of their neighboring CAVs. We start by formulating object usefulness as a maximization problem for each CAV over its communication coverage. Following that, we build a

978-3-903176-51-5 © 2022 IFIP

Partially Observable MDP (POMDP) framework [7] to characterize the CloudAC-IU environment under uncertain information that may come from onboard sensors and network restrictions. CAVs in CloudAC-IU cooperate in training a cloud-based Advantage Actor-Critic (A2C) [8] model. Once the training converges, each CAV uses a pre-trained copy of the global model individually to select which objects to include in a CPM. For that purpose, we introduce the A2C algorithm to conduct the learning process of the proposed approach. Obtained results show the ability of CAVs to conduct the training process on the cloud side and highlight the performances achieved by the proposal in terms of object redundancy, Channel Busy Ratio (CBR), and Packet Reception Ratio (PRR) compared to state-of-the-art works.

The rest of this article is structured as follows. Section 2 gives an overview of the most recent related works. Section 3 introduces the formulation of the information usefulness problem. In the next section, we present the CloudAC-IU design and algorithm. In Section 5, we present simulations and results. Finally, we conclude this work in Section 6.

II. Related Works

Despite the volume of data produced by onboard sensors and the restriction of the V2V-dedicated frequency range, the exchange of perception information between CAVs has been an active research topic in C-ITS. The work described in [9] is the first proposed system, called CarSpeak. It enables CAVs to exchange sensor information as 3D point clouds. This work encodes the raw perception information using an octree scheme, and CAVs broadcast their associated regions over networks. In the same context, the authors in [10] have introduced a multimodal cooperative perception system that provides drivers with see-through views using massive vision-based information from cameras and Lidars. Experiments in these works have shown that exchanging raw perception information may significantly degrade network performances. Regarding network constraints, the work in [11] has proposed the realization of CP by periodically exchanging only descriptions of tracked objects. This work was crucial in initiating the standardization of CP by the ETSI.

Regardless of the massive perception data and the restrictions of the V2V-dedicated frequency band, the ETSI has proposed a CPM format and a set of message generation rules based on tracked objects' dynamicity [3] to achieve a trade-off between EA and CBR. Authors in [12] have shown that employing CP may result in a large amount of useless information using these generation rules since CAVs do not analyze perceived information from their neighbors. This study also has demonstrated the need to design advanced techniques that dynamically control the information exchange on the wireless channel while ensuring the capacity of EA.

Various works have been proposed to reduce information redundancy. The authors of [13] have proposed a dynamics-based technique. Using this technique, CAVs analyze the last CPM received from all neighboring and exclude perceived objects that exceed a position and speed thresholds. In the same context, the authors in [14] have introduced an entropy-based technique to reduce redundant transmissions caused by multiple CAVs that may perceive the same object. During each CPM generation interval, each CAV estimates the entropy value of a perceived object to a potential receiver CAV based on an estimated history of received CPMs. Using this technique, a CAV excludes an object if all neighboring CAVs are anticipated to perceive it, and the relative entropy for all their neighbors is less than a predefined threshold. However, we highlight two significant drawbacks of these methods. First, they evaluate static thresholds that may not have suitable values in heterogeneous driving environments with changing vehicle densities. Second, they assume that all other CAVs in the surrounding employ the same technique, which is not always the case in mixed traffic when a group of vehicles cannot send or receive messages.

Authors in [15] have demonstrated that the existing message generation rules might result in a significant redundancy level in highway scenarios and then proposed a probabilistic data selection scheme [16]. This scheme enables each CAV to adapt the adaptive transmission probability of each detected object depending on its location and other road traffic information. As a DRL-based scheme, the work in [17] has proposed omitting duplicated perceived objects with objects that neighbor CAVs can perceive. Using this scheme, CAVs exchange perception information depending on an empty, occupied, or occluded state of the grid-based projection of their FoVs. However, these schemas transmit CPMs without considering information usefulness over the transmitter network coverage. This limitation still represents an open challenge in maintaining EA, especially in highly congested networks.

In the next section, we propose the formulation of object usefulness as a maximization problem, considering numerous perceptual settings such as position, distance, object size, viewing angle, and occlusions caused by other road users.

III. Formulation object usefulness

To begin, let V denote a set of vehicles of various classes driving in a CP environment. Each vehicle may be represented as a rectangle of length l and width w. We assume that $V' \subseteq V$ is a set of CAVs equipped with 360° onboard sensors with a maximum sensing range m and a circular communication coverage of radius R. At each status-related time interval ts, each CAV v_i of index i broadcast its current status information $status_i^{ts}$ and receive other statuses $Status_{j\neq i}^{ts}$ from other CAVs within its communication coverage Cov_i^{ts}.The status information $status_i^{ts}$ can be described by one or a set of properties, including position, type, speed, length, and width. For simplicity, we define the $status_i^{ts}$ by a set consisting of the current position x_i^{ts} and y_i^{ts} in the global coordinate system, the length l_i, and width w_i as follows:

$$status_i^{ts} = \{x_i^{ts}, y_i^{ts}, l_i, w_i\} \qquad (1)$$

On the other hand, $Status_i^{ts}$ can be denoted by:

$$Status_i^{ts} = \{status_j^{ts}; j \neq i \ and \ v_j \in Cov_i^{ts}\}, \qquad (2)$$

where Cov_i^t can be formally described by:

$$Cov_i^{ts} = \{v_j \,; j \neq i \ and \ ED_i^{j,ts} \leq R\}, \qquad (3)$$

with $ED_i^{j,ts}$is the Euclidean-based distance between v_i and v_j at ts, computed as:

$$ED_i^{j,ts} = \sqrt{\left(x_i^{ts} - x_j^{ts}\right)^2 + \left(y_i^{ts} - y_j^{ts}\right)^2} \qquad (4)$$

We consider tcp as a CP-related time. Each v_i performs a local perception phase using onboard sensors and perceives a set of objects O_i^{tcp} identifying a set of features, such as position and size, for each object during each tcp. However, including the perceived objects O_i^{tcp} into a CPM and broadcast to the other CAVs without prior processing could result in redundant useless information in the CP network.

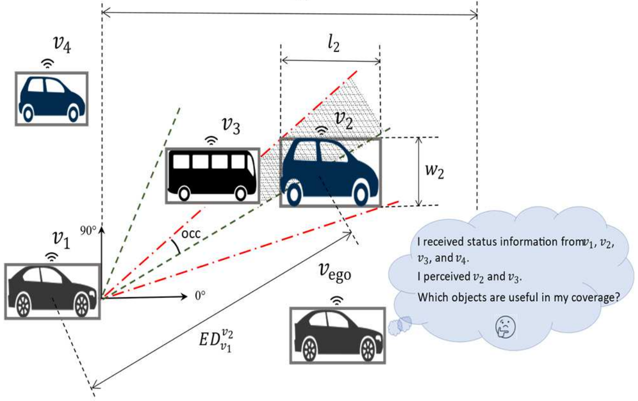
Fig. 1 depicts an Ego CAV v_{ego} that has received status information from multiple CAVs within its network coverage Cov_{ego}^{ts} at a given ts but has only perceived two CAVs within its sensing range. Aiming to reduce redundant useless information in the CP network, v_{ego} have to determine whether objects are useful to other CAVs within Cov_{ego}^{ts}. However, the usefulness of an object obj perceived by v_{ego} to another receiver CAV v_j might rely on distance and occlusion factors. The distance-related factor can be defined as a membership value in [0,1], of which obj appears in the FoV of v_j. It can be determined inversely proportional to m and means that the closer the distance is to m, the closer the membership value to zero, indicating that obj is being unperceivable by v_j. The distance-related factor can be denoted by:

$$DF_j^{obj,tcp} = \begin{cases} 0, \ ED_j^{obj,tcp} > m \\ 1 - \frac{ED_j^{obj,tcp}}{m}, \ otherwise \end{cases}, \qquad (5)$$

where $ED_j^{obj,tcp}$ is the Euclidean-based distance between obj and v_j as denoted in (4). On the other hand, the occlusion-related factor can be a membership value in [0,1] that refers to which obj is directly in the Line-of-Sight (LoS) of v_j. It can be determined proportionally to the sum sum of all angles that overlap and occlude the viewing angle $vAngle_j^{obj,tcp}$ from v_j to obj, meaning that the closer this sum is to $vAngle_j^{obj,tcp}$, the more the LoS to this object is occluded. The occlusion-related factor can be denoted by:

$$OF_j^{obj,tcp} = \begin{cases} 0, \ sum > vAngle_j^{obj,tcp} \\ 1 - \frac{sum}{vAngle_j^{obj,tcp}}, \ otherwise \end{cases}, \qquad (6)$$

We employ the $atang2$ function to determine the angle between two global coordinate system positions. Since $atang2$ computes the angle between a given position and the X-axis, a simple subtraction can be performed to get the recommended angle.



m
l_2
v_4
v_3
v_2
w_2
v_1
90°
occ
0°
v_{ego}
$ED_{v_1}^{v_2}$
I received status information from v_1, v_2, v_3, and v_4.
I perceived v_2 and v_3.
Which objects are useful in my coverage?

Fig. 1. A geometric representation of the CP environment by the ego CAV v_{ego}, where occ is the only angle representing the occluded part of the viewing angle from v_1 to v_2 caused by v_3.

To that end, we can define a visibility membership value $Visibile_{j\neq ego}^{obj,tcp}$ that indicates the value in [0,1] to which obj is visible to v_j based on the distance-related and occlusion-related factors defined in (5) and (6), respectively.:

$$Visibile_j^{obj,tcp} = OF_j^{obj,tcp} * OF_j^{obj,tcp} \qquad (7)$$

Assuming that $M_{ego}^{tcp} \subseteq CO_{ego}^{tcp}$ is a subset of connected objects determined by v_{ego} at tcp. The visible value of M_{ego}^{tcp} to v_j can be determined as an average value of all objects it includes, as follows:

$$Visible_j^{M_{ego}^{tcp},tcp} = \frac{1}{|M_{ego}^{tcp}|} \sum\nolimits_{\substack{obj \in M_{ego}^{tcp} \\ obj \neq v_{ego}}} Visible_j^{obj,tcp} \tag{9}$$

According to (9), we can generalize the visible value of M_{ego}^{tcp} for multiple CAVs in Cov_{ego}^{ts}, as follows:

$$Visible_{Cov_{ego}^{ts}}^{M_{ego}^{tcp}} = \frac{1}{|Cov_{ego}^{ts}|} \sum\nolimits_{\substack{v_j \in Cov_{ego}^{ts} \\ v_j \neq v_{ego}}} Visible_j^{M_{ego}^{tcp}} \tag{8}$$

To that end, the objective of each transmitter CAV v_{ego} in the CP environment is to broadcast disconnected objects combined with only useful connected objects M_{ego}^{tcp} that maximizes the nonvisible value defined in (10), as follows:

$$\underset{M_{\text{ego}}^{\text{tcp}}}{\text{maximize}} \quad 1 - Visible_{Cov_{ego}^{ts}}^{M_{ego}^{tcp}} \tag{9}$$

Subject to:

$$\text{C1:} \quad ts + \epsilon < tcp < 2ts,$$

where C1 ensures that CAVs broadcast and receive status information before sharing their local perception and that the environment is unchanged between ts and tcp.

IV. Maximizing information usefulness using Actor-Critic reinforcement learning

This section introduces the system model and learning algorithm.

A. System design

In RL, an agent interacts with a stateful environment without prior information by maximizing its reward through action and feedback at each timestep. Typically, a fully observable environment is modeled as an MDP in which the RL agent collects complete state information of the environment. However, in our context, CAVs construct the state of the environment and provide it to the cloud device based on the status information received from other CAVs. This results in the RL agent receiving only partial state information on the cloud device due to network restrictions and message loss. Therefore, we develop a POMDP to characterize the state of the CloudAC-IU environment in the presence of uncertain data.

Given that the CP environment varies among CAVs and that the training and storage of experienced state-decision are performed centrally in the cloud, a position-based representation may not conduct the training process. Therefore, as a position-independent representation, the environment state at timestep t

$$s^t = \{(ED_{ego}^{j,\text{t}}, vAngle_{ego}^{j,\text{t}}, l_j, w_j); \text{v}_\text{j} \in Cov_{ego}^t\}, \tag{12}$$

Where $ED_{ego}^{j,\text{t}}$ and $vAngle_{ego}^{j,\text{t}}$ are the distance and viewing angle from an ego that is providing its state to the cloud device and a CAV j in its communication coverage. As the environment is partially observable, we define an observation $o^t \subseteq s^t$ as a partial set from s^t that is computed based on received status information in $Status_{ego}^t$.

To define the action a^t of the RL agent at timestep *t*, we propose the region-based FoV that divides the FoV of each CAV into p pistes and s sectors. We represent the region-based FoV of a v_{ego} at t by a set of regions of size $s * p$ as follows:

$$FoV_{ego}^t = \{R_{0,0}^t, R_{0,1}^t, \dots, R_{0,s-1}^t, R_{1,0}^t, \dots, R_{p-1,s-1}^t\}, \tag{13}$$

where $R_{p',s'}^t$ is the region indexed by piste p' and sector s' at t. To that end, given a received environment observation from v_{ego} by the cloud device, the action a^t is selecting the regions whose objects are useful for the other CAVs in Cov_{ego}^t.

Finally, the reward represents the usefulness value denoted in (10) of the generated objects from the selected regions.

B. Learning Algorithm

We develop the A2C algorithm as a centralized cloud-based model to conduct the learning process. In particular, A2C is a policy gradient method used to solve large action spaces in complex RL problems. Its main goal is to find an optimal policy to obtain optimal rewards.

Algorithm 1 illustrates the CloudAC-IU learning algorithm. The main goal is to train a cloud-based A2C model to maximize the usefulness of CPM objects in the coverage of each transmitter CAV. Therefore, we define Algorithm 1 by the following three stages.

1) Initialization stage: At this stage, we initialize an actor-network π with random weights θ, critic-network Q with random weights θ', and a buffer of experiences D on the cloud device.

2) Action stage: This stage consists of the building, acting, and storing processes performed $N_{stepPerUpdate}$ times. Particularly, the building process consists of constructing the observation describing current environment of a given each CAV as defined in (12) and sending to the cloud device. The acting process of the cloud-based agent involves predicting an action according the policy model π and sending it back to the target CAV. This policy is modeled as a mapping from locally stored action-observation history τ to an action. On the other hand, the training process can be unstable or even diverge when nonlinear approximator functions such as DNN models are employed for the critic model [18]. To overcome this issue, we adopt the concept of experience replay [19]. Thus, the storing process involves recoding an experience $e^t = (s^t, a^t, r^t, s^{t+1})$ at each timestep t in D.

Algorithm 1: The CloudAC-IU learning algorithm

```
Input
$N_{updates}$ // Number of learning updates
$N_{stepPerUpdate}$ // Number of steps per learning update
Output
Learned model
Initialization stage
Initialize a critic network $V$ with random weights $\theta'$
Initialize a buffer of experiences $D$
Initialize a policy network $\pi$ with random weights $\theta$
Action stage
$updates \leftarrow 0$
while $updates < N_{updates}$ do
    $step \leftarrow 0$
    while $step < N_{slotsPerUpdate}$ do
        Each CAV builds $o^{step}$ and sends it to the cloud device
        Cloud: predict an action $a^{step}$ and send it to the target CAV
        Cloud:  receive a reward $r^t$ and  Store $e^{step}$= ($o^{step}, a^{step}, r^{step}, o^{step+1}$ ) into $D$
        $step \leftarrow step + 1$
    end while
Update stage
        Sample minibatch $b$ from $D$
        Compute Critic loss in $b$ based on (15) and update $\theta'$ according to (14)
        Compute Actor loss in $b$ based on (17) and update $\theta$ according to (16)
        $updates \leftarrow updates + 1$
end while
Cloud : send a pre-trainind copy of the policy network to each CAV in the driving scenario
End.
```

3) Update stage: During the learning process, the policy and critic models are updated N_{update} times. The update of an embedded critic is usually performed by minimizing the Temporal Difference (TD) calculated between the estimated and the actual values on a sampled minibatch of experiences $b = (s, a, r, s') \sim U(B)$ of size n, as, follows:

$$\theta' = \theta' + \alpha L(\theta'), \tag{10}$$

where α is the learning rate used to adjust model parameters and $L(\theta')$ is the loss function calculated as:

$$L(\theta') = \mathbb{E}_{(s,a,r,s')\in b}[(r \ + \gamma Q_{\theta'}(s') - Q_{\theta'}(s))^2]$$

$$= 1/n \sum_{t=0}^{n-1} (r^t + \gamma Q_{\theta'}(s^{t+1}) - Q_{\theta'}(s^t))^2 \tag{11}$$

Following that, we perform an update step of the policy model as follows:

$$\theta = \theta + \alpha \nabla_\theta J(\theta), \tag{12}$$

where $\nabla_\theta J(\theta)$ is gradient calculated based on the policy gradient theorem [20], as follows:

$$\theta \nabla_\theta J(\theta) = \mathbb{E}_{s,a\sim\tau}[\nabla_\theta \, log \, \pi_\theta \, (a|s) A(s,a)]$$

$$= \sum_{t=0}^{T=|\tau|-1} \nabla_\theta \, log \, \pi_\theta \, (a^t |\, s^t) A(s^t \, , a^t), \tag{13}$$

where $A(s^t \, , a^t)$ is the advantage value denoting how good the chosen action is in the state s^t. It is calculated as:

$$A(s^t, a^t) = r^t + \gamma Q_\theta(s^{t+1}) - Q_\theta(s^t) \tag{14}$$

where γ ∈ [0,1] is a discount factor given to early discounted accumulative rewards to reduce their impact. Once the training process reaches $N_{updates}$ model updates, the cloud device sends a pre-trained copy of the policy network to each CAV in order to employ it to select which useful objects to include in a CPM at each generation interval.

V. Simulation and Results

We conduct simulations using the Artery [21] and SUMO (Simulation of Urban Mobility) [22] frameworks on a 10 km2 real-world map of Bordeaux, France. The map is obtained from OpenStreetMap and includes various road traffic scenarios, such as the city center and highways with various situations, such as ramps and T-junctions. We randomly generate vehicles with different classes and sizes. We consider an ITS-G5 communication profile for each CAV and a coverage of 500 m where it can send and receive messages. We set the transmission power to 23 dBm. CAVs exchange CAMs and CPMs every 0.1s and 0.15s, respectively, using a 6Mbps data rate and the control channel (CCH) without Decentralized Congestion Control (DCC). We configure all CAVs with the same local perception capabilities. We set up 360° radar and lidar sensors on each CAV with a maximum sensing range of 100 m.

We implemented the CloudAC-IU by building Multilayer Perceptron (MLP) networks based on the

PyTorch library. The simulation spans 50s. The training process starts at the first simulation step. The training phase consists of $N_{updates} = 3000$ update times, where each update time be made up every $N_{slotsPerUpdate} = 10$ time steps. To conduct the training process, we applied the RMSprop optimizer [23] with a learning rate $\alpha = 10^{-3}$, a minibatch size $|b| = 64$, a discount factor $\gamma = 0.99$, and a buffer size $|B| = 10^{6}$. Finally, for complexity reasons, we divide the CAV FoVs equally into 3 pistes and 3 sectors, resulting in 9 distinct FoV regions.

We start by analyzing the training convergence of the proposal. The cloud continuously receives environment states from CAVs and performs a learning step at each received state. Fig. 2 depicts the average reward representing the average CPM usefulness as a function of model update steps. As illustrated, the average CPM usefulness shows a high variance during the first update steps due to the dynamicity of the environment and the lack of sufficient data. However, this metric increases with the update steps and is maximized after around 1000 model update steps. This means that CAVs have collaboratively trained a centralized object selection policy after around 10000 learning steps successfully.

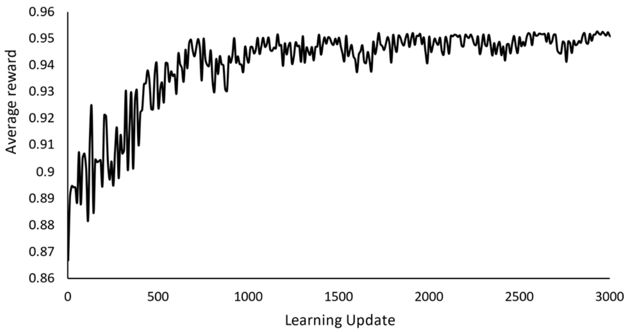


Fig. 2. Average Reward variation as a function of learning update steps.

After the training update times reaches the predefined number of updates steps, each CAVs receives a pre-training copy of the policy model and then uses it to select useful objects to include in a CPM at each generation interval. To that end, we compare the CloudAC-IU performance to the ETSI CPM generation rules [3] (baseline) and the dynamics-based technique [13]. We consider an object redundant for both techniques if the absolute speed or position value difference is less or equal to 0.5 m/s and 4 m, respectively.

Fig. 3 depicts an Object Redundancy (OR) metric, which identifies the number of times a CAV receives an update about the same object over the selected time interval as a function of the distance between the perceived object and the CAV receiving it. As shown, the baseline results in higher OR levels at short distances because perceived objects are successfully detected and exchanged by multiple CAVs simultaneously in the network. However, analyzing the last CPM from all CAVs by the dynamic-based technique has a marginally reduced OR compared to the baseline. On the other hand, CloudAC-IU reduces OR considerably at short and medium ranges of less than 300 m., which means that CAVs exchange only useful objects in the CPMs.

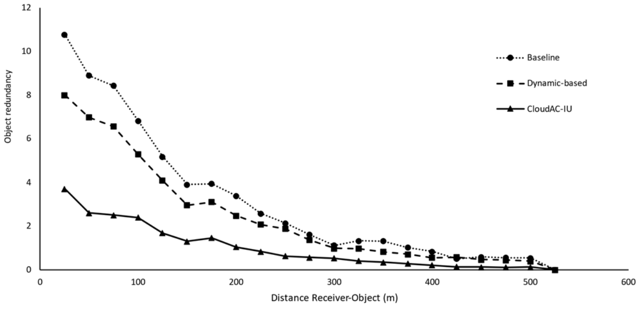


Fig. 3. OR as a function of the distance between the perceived object and the CAV receiving it.

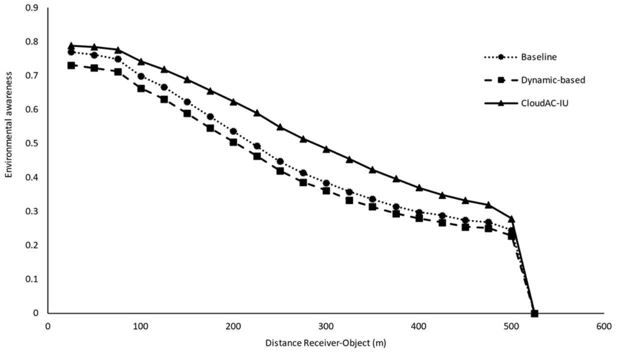


Fig. 4. EA as a function of the distance between the detected object and the vehicle receiving it.

Typically, the improvement achieved in OR improves the network reliability. We measure this reliability using Channel Busy Ratio (CBR) and the Packet Reception Ratio (PRR) metrics as follows. Table I illustrates the average CBR and PRR for all CAVs in the driving scenarios. We notice that the baseline approach results in a higher CBR value of 40,25 % as it generates a high number of redundant objects depicted by the OR. However, the dynamic-based technique improves slightly by reducing the CBR value by around 4 %. On the other hand, CloudAC-IU significantly reduces CBR by around 10 % and 14 %, compared to the dynamic-based technique and the baseline approach, respectively. These improvements in CBR by the CloudAC-IU ensure an increase of 10% and 11% in the average PRR compared to the baseline approach and the dynamic-based technique.

TABLE I. CHANNEL BUSY RATIO (CBR) AND PACKET RECEPTION RATIO (PRR)

Approach / Metrics	**CBR**	**PRR**
Baseline	40,25 %	77.83 %
Dynamic-based	36.54 %	78.20 %
CBR-selective	34.12 %	80.32 %
CloudAC-IU	26.34 %	88.11 %

Improving the network's reliability allows CAVs to receive additional useful objects via CPMs, increasing their EA. In this context, EA represents the ratio between the number of unique objects known by a CAV and the total number of objects in its communication coverage. Fig. 4 plots EA as a function of the distance between the detected object and the vehicle receiving it. Compared to the baseline, the dynamic-based has almost attained the same EA at distances larger than 100 m; meanwhile, they lower this metric by around 5% at distances smaller than 100 m. In contrast, the CloudAC-IU improved EA at distances smaller than 100 m as well as in medium and large distances from 100 to 400 m. The improvements obtained by the CloudAC-IU can be expressed because the CAVs send and receive additional CPMs, which seem to be lost or not sent by the remaining approaches.

CONCLUSION

In this paper, we proposed CloudAC-IU as a cloud-based DRL approach to maximize the utility of perception information and reduce information redundancy in the network. We implemented and evaluated the proposed approach using advanced network and road traffic simulators. Simulation results showed that the proposed mitigated redundant objects and improved network reliability without sacrificing environmental awareness. However, further research should be conducted to ensure CP resiliency while addressing the centralized training delay and system failure limitations.

REFERENCES

[1] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, "Enhancements of V2X communication in support of cooperative autonomous driving," IEEE Commun. Mag., vol. 53, no. 12, pp. 64–70, 2015, https://doi.org/10.1109/MCOM.2015.7355568.

[2] ETSI EN 302 663 V1.3.11, Intelligent Transport Systems (ITS); ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency bands, 2020.

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

[4] A. Haydari and Y. Yılmaz, "Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey, IEEE Transactions on Intelligent Transportation Systems," vol. 23, no. 1, pp. 11-32, 2022, https://doi.org/10.1109/TITS.2020.3008612.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015, https://doi.org/10.1038/nature14236.

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.

[7] A. Cassandra., "A Survey of POMDP Applications," AAAI Fall Symposium, 1998.

[8] V. R. Konda, J. N. Tsitsiklis, "Actor-critic algorithms, Proceedings of the Conference on Neural Information Processing Systems," 2000, pp. 1008–1014.

[9] S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, and D. Rus, "Carspeak: a content-centric network for autonomous driving," in Proc. of ACM SIGCOMM, 2012.

[10] S. W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "Multivehicle cooperative driving using cooperative perception: Design and experimental validation," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 663–680, 2015.

[11] Günther, H.J., Mennenga, B., Trauer, O.; Riebl, R., Wolf, L., "Realizing collective perception in a vehicle," Proc. the 2016 IEEE Vehicular Networking Conference (VNC), Columbus, OH, USA, 8–10 2016, https://doi.org/10.1109/VNC.2016.7835930.

[12] G. Thandavarayan, M. Sepulcre, J. Gozalvez, "Analysis of Message Generation Rules for Collective Perception in Connected and Automated Driving," 2019 IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 134-139, https://doi.org/10.1109/IVS.2019.8813806.

[13] G. Thandavarayan, M. Sepulcre, J. Gozalvez, "Redundancy Mitigation in Cooperative Perception for Connected and Automated Vehicles," 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1-5. https://doi.org/10.1109/VTC2020-Spring48590.2020.9129445.

[14] T. Higuchi, M. Giordani, A. Zanella, M. Zorzi, O. Altintas, "Value-Anticipating V2V Communications for Cooperative Perception," Proc. the 30th IEEE Intelligent Vehicles Symposium, pp.1690-1695, 2019. https://doi.org/10.1109/IVS.2019.8814110.

[15] H. Huang, W. Fang, H. Li, "Performance Modelling of V2V based Collective Perceptions in Connected and Autonomous Vehicles," Proceedings of the 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrueck, Germany, 2019, https://doi.org/10.1109/LCN44214.2019.8990854.

[16] H. Huang et al., "Data Redundancy Mitigation in V2X Based Collective Perceptions, IEEE Access, vol. 8, pp. 13405-13418, Jan. 2020. https://doi.org/10.1109/ACCESS.2020.2965552

[17] M. K. Abdel-Aziz, C. Perfecto, S. Samarakoon, M. Bennis and W. Saad, "Vehicular Cooperative Perception Through Action Branching and Federated Reinforcement Learning, IEEE Transactions on Communications, vol. 70, no. 2, pp. 891-903, Feb. 2022. , Jan. 2020. https://doi.org/10.1109/TCOMM.2021.3126650

[18] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation, IEEE Transactions on Automatic Control," vol. 42, no. 5, pp. 674-690, 1997, https://doi.org/10.1109/9.580874.

[19] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, N. de Freitas, "Sample efficient actor-critic with experience replay, ICLR, 2017, https://doi.org/10.48550/arXiv.1611.01224.

[20] R. S. Sutton, D. McAllester, S. Singh, Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," Proceedings of the 12th International Conference on Neural Information Processing Systems (NIPS'99), MIT Press, Cambridge, MA, USA, 1057–1063.

[21] R. Riebl, H. Günther, C. Facchi and L. Wolf, "Artery: Extending Veins for VANET applications," 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015, pp. 450-456, https://doi.org/10.1109/MTITS.2015.7223293.

[22] Krajzewicz, Daniel & Erdmann, Jakob & Behrisch, "Michael & Bieker-Walz, Laura. (2012)," Recent Development and Applications of SUMO - Simulation of Urban Mobility," International Journal On Advances in Systems and Measusrements. 3&4.

[23] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, 2016