

Copy-CAV: V2X-Enabled Wireless Towing for Emergency Transport[★]

Constantine Ayimba^{a,*,1}, Valerio Cislighi^{b,2}, Christian Quadri^{b,3}, Paolo Casari^{c,4} and Vincenzo Mancuso^{a,5}

^aIMDEA Networks Institute, Avenida del Mar Mediterráneo 22, Leganés, 28918, Madrid, Spain

^bComputer Science Department, Università degli Studi di Milano, Via Celoria, 18, Milano, 20133, Italy

^cUniversity of Trento, Via Sommarive, 9, Povo, 38123, Trento, Italy

ARTICLE INFO

Keywords:

Reinforcement Learning
Emergency Transport
MEC
Connected Autonomous Vehicles
V2X

ABSTRACT

As smart connected vehicles become increasingly common and pave the way for the autonomous vehicles of the future, their ability to provide enhanced safety and assistance services has improved. One such service is the emergency transport of drivers in medical distress: as a positive solution of the distress is typically more likely after timely response, an autonomous vehicle could cut on emergency response times, and thus play a key role in saving the life of its driver.

In this paper, we show how such an autonomous emergency transport service can be run from a wireless cellular network, and discuss the importance of having a human in the loop in order to expedite driving. We present a Monte-Carlo-based driver assessment system that the network can use when selecting the most suitable candidate to wirelessly tow an autonomous vehicle with an incapacitated driver. We show that this mechanism results in a selection policy that ensures better cohesion between the vehicles, thereby significantly improving service reliability by reducing the chances of disruptions by intervening traffic.

1. Introduction

The favorable outcome of a medical emergency largely depends on the timely delivery of professional or specialist care [1]. The latter is however complicated when the emergency occurs in inconvenient situations, such as while a person drives alone. In this case, not only is the person typically unable to properly summon help, but can possibly represent a danger to him/herself and a safety hazard for other road users if s/he loses control of the vehicle as a consequence of critical medical conditions. As Connected Autonomous Vehicles (CAVs) become progressively prevalent, their capabilities can be leveraged to provide much needed emergency transport services in such situations.

A major stumbling block for this service is that it will take a very significant time before CAVs represent the largest majority of the circulating vehicles. In the same vein, support from roadside infrastructure may be insufficient in some areas, requiring CAVs to make decisions based only on local information, rather than coordinating broadly to improve the quality of connected driving [2]. In the meantime, CAVs will share road infrastructure with

unpredictable actors, such as human-driven vehicles and pedestrians, that may individually decide to relax or break road circulation rules, and may thus represent an additional source of danger. Moreover, it may not always be feasible for CAV sensors to correctly read road signs in all conditions [3]. These aspects greatly complicate the decision-making processes of a CAV, and represent a significant obstacle towards promptly solving an emergency scenario. Whereas human drivers can safely bend the rules when a situation so requires, such intuitions are examples of “corner cases” that are not adequately incorporated into the training of CAVs [4].

However, if these more unpredictable decision-making aspects could be at least partly assigned to a human driver, it becomes possible to exploit the communication and self-driving capabilities of a CAV to provide emergency transport, while following a human-driven car that proceeds along a desirable path leading to emergency care services. To this end, we propose Copy-CAV, a wireless towing service of CAVs by a human-driven connected vehicle. By having a human in the loop, Copy-CAV essentially simplifies autonomous driving to imitation driving, thereby avoiding the pitfalls of the former, and especially enabling some form of “rule bending” as CAVs imitate a flexible human driver.

With Copy-CAV, the lead, human-driven vehicle acts as a guide to direct the CAV through complicated driving scenarios which would otherwise be overwhelming for the AI agent driving the vehicle with a driver in distress. For the towed vehicle, it would become possible to expedite the delivery of its occupant to emergency care by best following the route and trajectory of its human-driven tower. Given the centrality of the latter, a proper choice of the candidate human driver/vehicle to be used is crucial to the overall performance of this service. To support

[★]Corresponding author

✉ constantine.ayimba@imdea.org (C. Ayimba);
valerio.cislighi@studenti.unimi.it (V. Cislighi);
christian.quadri@unimi.it (C. Quadri);
paolo.casari@unitn.it (P. Casari);
vincenzo.mancuso@imdea.org (V. Mancuso)

🌐 <https://networks.imdea.org/team/imdea-networks-team/people/constantine-ayimba/> (C. Ayimba); (V. Cislighi);
<https://www.unimi.it/it/ugov/person/christian-quadri> (C. Quadri);
<https://webapps.unitn.it/du/it/Persona/PER0221586> (P. Casari);
<https://networks.imdea.org/team/imdea-networks-team/people/vincenzo-mancuso/> (V. Mancuso)

ORCID(s): 0000-0003-4032-9735 (C. Ayimba);
0000-0002-3608-8142 (C. Quadri); 0000-0002-6401-1660
(P. Casari); 0000-0002-4661-381X (V. Mancuso)

the service with effective decisions, we thus propose a reinforcement learning driver/vehicle evaluation system, so that the combined choice of driver and car improves over time.

The introduction of multi-access edge computing (MEC) has unleashed the potential of supporting connected vehicles through centralized coordination services running close to the network edge, typically in small-scale data centers serving one or multiple 5G base station (gNBs). The availability of computational resources at the network edge along with the low latency provided by beyond 5G/6G NR have proven suitable for cooperative vehicle applications such as platooning [5, 6]. Therefore, we exploit MEC resources to host and manage our Copy-CAV solution. Compared to the vehicle-to-vehicle (V2V) communication paradigm, the vehicle-to-NR (V2N) approach solves uncoordinated channel access issues,¹ shadowing effects and limited communication range problems.

Our contributions in this paper can be briefly summarised as follows:

1. we implement and test a vehicle following system that relies on the trajectory created by a lead, human-driven vehicle;
2. we introduce a driving behavior metric to characterise human driving;
3. we implement a driving behavior/vehicle selection algorithm leveraging Monte Carlo Reinforcement Learning;
4. we test the feasibility and robustness of our algorithm using the established 5G network simulator Simu5G [7] and the robust autonomous driving simulator CARLA [8].

Copy-CAV aligns with the principles of Trustworthy AI [9] by having a human in the loop, by relying on transparent features and on a clearly defined reward function—without black-box function approximations—to drive learning. This work enhances our previous study presented in [10] with the following additional contributions: (i) we consider the influence of vehicle power on driver behavior, (ii) we simulate the impact of network communication performance (goodput and latency) on the wireless towing service via a non-trivial interfacing between the CARLA framework and the OMNeT++-based Simu5G framework, (iii) we augment the reward process to include the time taken to get the patient to the required destination.

Notably, to the best of our knowledge, we are the first to synchronize these two powerful simulators via a custom API to effectively reproduce both the autonomous driving component (through the high degree of realism found in CARLA) and the connected vehicle communication aspects

of CAVs (through the Simu5G/OMNeT++ framework and its capability to realistically model 5G cellular networks).

In the remainder of this paper, we discuss related work (Section 2), describe our foreseen wireless car towing service (Section 3), and explain the design of the reinforcement learning-enabled lead driver selection (Section 4). We then detail our simulation setup (Section 5) and results (Section 6), before drawing conclusive remarks in Section 7.

2. Related work

Recent studies [11, 12] have investigated the use of smart vehicles to continuously monitor their driver's health, so that mitigating actions can be taken quickly. The type of ailments typically considered in these studies include cardiac arrests and strokes, which tend to occur suddenly and require immediate or as-fast-as-possible medical attention. Along the same line, driver fatigue and the consequent lapse in focus is also considered as a road safety hazard by the authors of [13]. These studies explore the feasibility of leveraging connected vehicles to ensure timely responses in the event of health emergencies. Our work provides an example of how such monitoring systems can be exploited to enhance health services by using capabilities of 5G and beyond networks.

Our approach extends previous work on vehicle following systems, such as those on Adaptive Cruise Control (ACC) [14, 15, 16] and platooning [17, 18, 19, 6]. These works focused on maintaining a safe distance from a preceding vehicle, and optimizing collision avoidance aspects in small as well as large coordinated groups of vehicles. In contrast to platooning, in our approach the lead and follower vehicles are not as tightly coupled by a control law. This is crucial to ensure that the service is robust to interruptions in road traffic by intervening vehicles and/or pedestrians. In fact, the operating environment is not restricted to open road scenarios, and may likely be in a busy urban area.

Recent progress in the standardization of 5G-and-beyond/6G New Radio (NR) are projected to enable a slew of services requiring low latency and high reliability [20]. The latter, coupled with the rise of CAVs [21], enable the creation of our emergency towing service that can effectively help an incapacitated driver.

In this work, we opt for V2N as the primary communication paradigm. Although it would be possible to use V2V instead of V2N, or V2N and V2V in combination (e.g., by relying on V2V as a backup or fallback solution as done for platooning control in [19]) here we restrict the investigation to V2N as the natural means to connect vehicles and edge computing resources needed by the proposed AI approach. V2N is also inherently safer by design, as it is more robust and controllable by the network in light of the network operator's more accurate take on the operational context of the wireless towing service. However, we plan to consider V2V variants and extensions for future work.

¹With the exception of LTE-V2V mode-3, which relies on cellular infrastructure for managing radio resources.

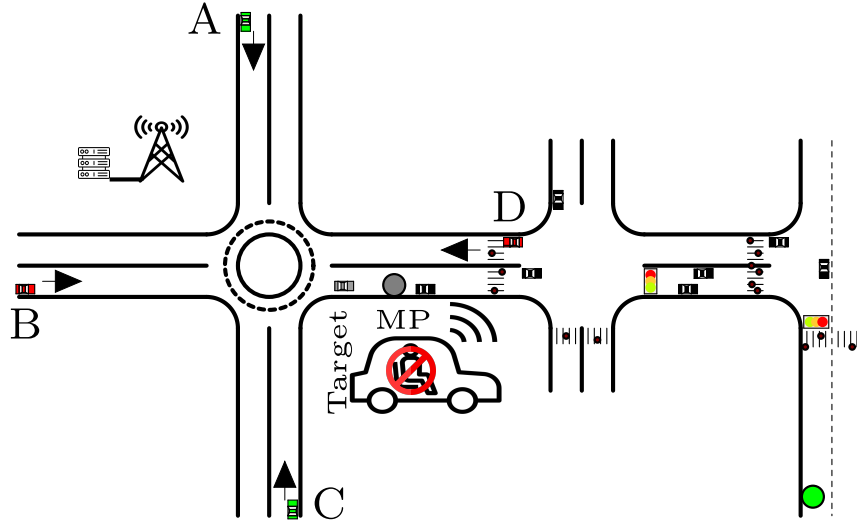


Figure 1: Illustration of wireless towing. The CAV sends an alert to the network that the driver is incapacitated. The network alerts candidate lead vehicles in the vicinity (marked A, B, C, D). As they move towards the meeting point MP, the network service evaluates the driving behavior and the engine power of the candidate lead vehicles, then selects one vehicle to perform the tow. When the towed vehicle arrives at the destination (green point), the network selection agent updates its learning algorithm.

3. Wireless towing scenario and service

3.1. Wireless towing definition

Similar to the classical notion of a leading vehicle pulling another vehicle mechanically joined to it, we define wireless towing as the autonomous movement of a CAV following another vehicle that precedes it along a given route on the road. The CAV, in this case, may receive directives from the lead vehicle itself, the network or both. While comparable to platooning, the key distinctive aspect of wireless towing is that the spacing between the lead and tail vehicle is not maintained relatively constant, and in the same vein the relative speed of the two vehicles is allowed to depart from zero, even by a significant amount.

This fact is of great importance, because the emergency towing service should be versatile enough to work in crowded urban environments, with intervening traffic as well as stops due to other vehicles, traffic lights, and pedestrians. Unlike platooning, where all intervening vehicles must necessarily adhere to the platoon control law (in other words, they should be *part of* the platoon), in the case of wireless towing the intervening vehicles move independently. The service is designed to tolerate such occurrences, since the tail vehicle is only informed about the path it has to take, and not about how much it needs to accelerate or brake. This choice represents a different tradeoff point that privileges flexibility, as required to accommodate a variety of road situations that may occur due to the presence of unpredictable human actors. In fact, the movement dynamics efficiency sought in platooning are not the main issue in wireless towing, which concentrates instead on the timely delivery of a driver in distress to emergency services.

In our wireless towing approach, we employ a towing controller (agent) that resides on the edge of the network.

This aspect is key to reduce latency and thus limit the transmission of alerts to geographical locations close to the CAV with the incapacitated driver.

3.2. Scenario description

The use case for emergency towing is depicted in Fig. 1. The CAV's sensors signal to the network that the driver is unresponsive in a manner similar to the process defined in [11, 13]. Upon such a detection, the vehicle sends out an alert to the network, which reacts by establishing and running the emergency service. In the alert, the vehicle specifies its geographic information, model, and engine data. The network then polls available drivers in the vicinity of the target vehicle. As there may exist a number of drivers with different capabilities, vehicles, and driving styles, it becomes central to choose the most appropriate driver and car to take the lead in the towing process. For this reason, the towing service management consists of two phases: (i) driver/vehicle evaluation phase and (ii) performance evaluation phase.

Driver/vehicle evaluation phase: We choose to have a short driving evaluation phase at the start, given that:

- driving behavior like any human activity is highly variable,
- the set of responding drivers in the vicinity of the vehicle to be towed will typically be different,
- maintaining the integrity of a highly dynamic, persistent database of drivers across all edge nodes of a network adds undesirable overhead and bookkeeping to maintain the service.

The candidate lead drivers/vehicles that respond to the poll start moving towards the target. The network continuously monitors the motion of these drivers/vehicles to

collect metrics related to their driving style and dynamics. After a short time interval (lasting no more than a total of 30 s) the agent at the network edge evaluates the behavior of each responding vehicle. The network agent then selects a candidate driver/vehicle deemed the most suitable among the ones available to carry out the task. The rest of the candidates are alerted of this choice and are relieved from the service.

Performance evaluation phase: When the chosen lead vehicle reaches the target tail vehicle, it overtakes it and stops slightly ahead of it after confirmation from the network. The network then immediately alerts the tail vehicle to activate its towing mode, so as to follow the lead vehicle ahead of it. The tail and lead vehicle maintain constant communications with the network. Specifically, both vehicles continuously update the network regarding their geographical locations and speed.

When the towing process is complete, i.e., the tail CAV has reached the intended destination, the network agent gauges the driving experience based on the data it received over the course of towing. The agent calculates the reward based on two components: (i) the evolution of the spacing between the lead vehicle and the CAV; and (ii) the time taken to reach the destination. While the second component is obvious, the rationale behind the first component of the reward is that, for an effective service, the space between the vehicles needs to be maintained small enough. This minimizes the likelihood that other vehicles would position themselves in between, as this would result in delays depending on how the preceding vehicle is being driven. Moreover, a distressed but conscious tail vehicle passenger should not get the feeling that the wireless towing service is leading them astray, as this event would negatively affect the perceived quality of experience (QoE) of the service.

The network agent will then update the corresponding driving behavior profile with the calculated reward in its reinforcement learning repository. As the service is repeatedly used, the learning agent will observe more towing experiences, and will accrue additional observations. Thanks to such experiences, the agent progressively develops a lead driver/vehicle selection policy. Following such a policy, the agent will improve the selection of the most suitable lead driver and vehicle over time. In due course, after the policy stabilizes, the network agent will consistently select the candidate leader with the driving behavior that exhibits the best spacing discipline and transport time, hence the best service quality.

4. Design of lead driver/vehicle selector

As stated earlier, we leverage Monte-Carlo reinforcement learning [22] to create our driving behavior evaluation and selection algorithm. The objective of the selector is to choose a lead vehicle and driving behavior that best matches the response of the CAV. The latter will ensure a smoother towing performance, a higher likelihood that

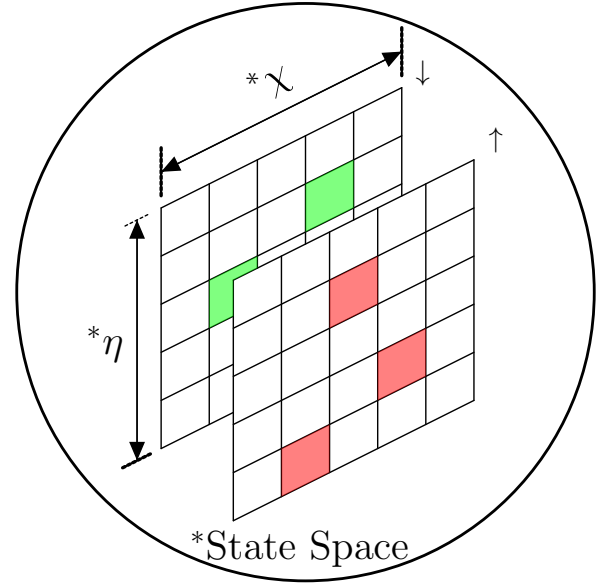


Figure 2: Tow controller(agent) State and Action Spaces. The driving behavior indicator (χ) and vehicle power ratio, (η), constitute the state space. The action space is represented by the red and green colored cells. The green cells represent candidate lead vehicles (labelled \downarrow) which are less powerful than the vehicle to be towed. The red cells are candidate lead vehicles (labelled \uparrow) with the same or higher engine power as the vehicle to be towed.

the lead and tail vehicle move cohesively throughout the required route, ideally with no intervening vehicle getting between them, and will reduce the time required to get the tail vehicle to the intended destination. In particular, we characterize the environment through two state variables: a driving behavior indicator χ that captures the behavior of the candidate lead driver, and the ratio η between the engine power of the lead and tail vehicles. The state and action spaces are shown in Fig. 2. With respect to this figure, in the presence of several options for driver behaviors and engine powers, those where $\eta < 1$ (i.e., the engine power of the lead vehicle is less than that of the towed vehicle) are represented as green cells, whereas those for $\eta \geq 1$ are represented as red cells. The action space includes all colored cells, corresponding to prospective lead drivers that are available in the proximity of the vehicle in distress.

When the alert is received, a candidate lead vehicle sets up a path to the position of the target vehicle, and starts reporting data that the decision agent can use for motion monitoring and lead vehicle selection. The path that the candidate lead vehicle follows is constituted by many waypoints at regular intervals, which dictate the movement of the vehicle to the meeting point. Let \dot{x}_j be the sample speed at waypoint j , $\sigma_{\dot{x}}^2$ the empirical variance of the set of speed values evaluated across all waypoints, $\bar{\dot{x}}$ the mean speed between waypoints and W the number of waypoints until the meeting with the vehicle to be towed. Then,

$$\sigma_{\dot{x}}^2 = \frac{1}{W} \sum_{j \in W} (\dot{x}_j - \bar{\dot{x}})^2, \quad (1)$$

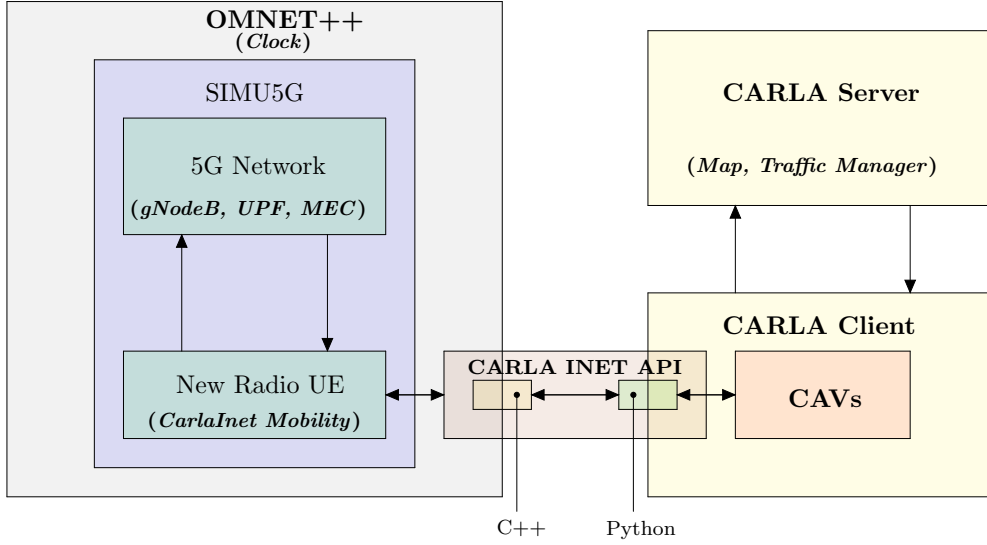


Figure 3: Integrated simulator setup with OMNeT++ and CARLA. We make use of the Simu5G framework for network simulation and implement a CARLA-OMNeT++ API that bridges the C++ environment based on OMNeT++ with the CARLA client. This enables the two simulators to work in tandem, and introduce Simu5G’s realistic networking aspects into the driving environment of CARLA.

and, χ is given as:

$$\chi = \frac{(\sum_{j \in W} \dot{x}_j)^2}{W \sum_{j \in W} \dot{x}_j^2} \cdot \frac{\sqrt{\sigma_x^2}}{3}. \quad (2)$$

The first term of Eq. (2) is the fairness index [23] of speed across all waypoints, and the second term accounts for the standard deviation of the speed waypoints normalised according to the 3σ empirical rule [24]. To obtain η , we use the ratio between the maximum revolutions per minute (RPM) that can be reached by the candidate lead vehicle and the target vehicle to be towed. Specifically, we define

$$R_{\text{rpm}} = \frac{\text{MaxRPM}_{\text{candidate}}}{\text{MaxRPM}_{\text{target}}}, \quad (3)$$

$$\eta = \begin{cases} R_{\text{rpm}}, & \text{if } R_{\text{rpm}} < 1.0, \\ \frac{1}{R_{\text{rpm}}}, & \text{otherwise.} \end{cases} \quad (4)$$

We remark that the above definition is not ambiguous because we preliminary subdivide the state space in two portions including more powerful and less powerful lead vehicle choices than the towed vehicle (see also Fig. 2). This makes it possible to make Eq. (4) bijective across each of the sub-spaces of the state space.

In order to keep the state space small, we quantize χ to 10 levels $0, \dots, 9$. We similarly quantize η to 5 levels $0, \dots, 4$. For this, we employ a uniform quantizer: other choices are possible (e.g., with finer sampling over the interval of behaviors and engine power ratios that represent most of the drivers and vehicle). However, these settings would be location-dependent (e.g., more powerful

vehicles may be more common in rich cities than in rural neighborhoods), as well as population- and time-dependent (e.g., at nighttime both very prudent and very aggressive driving behaviors may appear). By way of contrast, we seek a general approach, that works independent of the context. We update the state values using the following modified update:

$$V(S) \leftarrow \frac{1}{k} [\Delta + (k-1)V(S)], \quad (5)$$

where $\Delta = R + (\gamma - 1)V(S)$, $\gamma \leq 1.0$ is a discount parameter, $k \geq 1$ is the number of times the agent was in state S , and R is the reward at the end of the episode. Note that we used a modified update formula in Eq. (5), because it would be impossible to consider the simple reward R otherwise. In fact, the state values are updated at the end of a learning episode, and the lead driver and vehicle selected in a previous episode have no bearing on the current episode. The reward value, R , is calculated as

$$R = w_t \frac{t_{\text{ideal}} - t_{\text{tow}}}{t_{\text{ideal}}} - w_x \frac{\text{median}(|x_i - x_{\text{ideal}}|)}{x_{\text{ideal}}}, \quad (6)$$

where x_i is the spacing between the lead car and target tail vehicle taken at regular intervals, x_{ideal} is the desired spacing that should be maintained between the lead and tail vehicle, t_{ideal} is the empirical optimum emergency transport time [1] that provides for favorable outcomes and t_{tow} is the actual towing time. Additionally, the weights w_t and w_x are used to nudge the selection algorithm to choose options that perform better as regards towing time or spacing respectively. We use equal weights to give equal importance to both in the eventual policies. Thus greater rewards are assigned to towing outcomes where both the

vehicle spacing and the time taken to complete the towing are closest to the prescribed ideal value. While it may be tempting to reward faster towing realizations we remark that this would promote excessively aggressive behavior, which may yield a very uncomfortable experience for the driver in distress. In any event, different settings of t_{ideal} and x_{ideal} are still possible based on the readings of the vehicle about the distressed driver's state of health (e.g., heart issues may mandate shorter towing times than less critical losses of consciousness). The controller also notifies the lead vehicle, to slow down, when its spacing goes above $2x_{\text{ideal}}$ thereby limiting the adverse effects of aggressive driving behaviours.

For the present setup, we set $x_{\text{ideal}} = 10$ m, as proposed in [6], and $t_{\text{ideal}} = 240$ s as specified in [1]. Note that, for the second term of Eq. (6), we use the median value of the deviation, and not the mean as in our previous work [10]. This choice makes the reward value more robust to outliers, which are likely to occur under the ever-changing and highly variable conditions of typical road traffic. This choice also helps speed up policy convergence.

At the start of each episode, we use ϵ -greedy selection to choose the lead car and driving behavior option. Since a given driving behavior and vehicle combination may occur with greater frequency than others, we prevent bias in the policy by setting a threshold for the number of significant visits a state can have. If the number of visits to a state exceeds the threshold, further visits to the state do not reduce the corresponding ϵ value, thereby ensuring that randomness persists in the selection process. If for a given set of available options (candidate behavior/vehicles) all states have reached this threshold, then $\epsilon = \epsilon_{\min}$ for all states, and the selection agent acts according to policy with probability $1 - \epsilon_{\min}$.

We also employ the weighted fair exploration mechanism presented in our previous work [25] to improve the trade-off between exploration and exploitation. The mechanism can be briefly summarized as follows. At the k^{th} episode, if the option yielding the highest reward(after $k - 1$ episodes) is not selected, the probability $P_l^{(k)}$ of selecting any of the L options available depends on their value function $V_l^{(k)}(S)$ and the number of times $n_l^{(k)}$ that it has been previously selected. Specifically, we set

$$P_l^{(k)} = \begin{cases} \frac{1}{L}, & \text{for } \Psi_l^{(k)} = 0 \\ \frac{\Psi_l^{(k)}(1 - \tanh \phi_l^{(k)})}{\sum_{j=1}^L \Psi_j^{(k)}(1 - \tanh \phi_j^{(k)})}, & \text{for } \Psi_l^{(k)} > 0, \end{cases} \quad (7)$$

where

$$\begin{aligned} \Psi_l^{(k)} &= V_l^{(k)}(S) + \sum_{i=1}^L |V_i^{(k)}(S)|, \\ \phi_l^{(k)} &= \frac{n_l^{(k)}}{k}. \end{aligned} \quad (8)$$

5. Simulation setup

We implement the towing service as a coordinated discrete event simulation with complementary simulations on OMNeT++ and CARLA running simultaneously and concurrently, as shown in Fig. 3. CARLA simulator is designed for autonomous driving research and provides developers with a full-fledged set of APIs for modeling, sensing and controlling detailed physical environments. It is built on top of the Unreal Engine² which offers a remarkable level of physical and graphical realism. Unlike SUMO [26], which only provides standard car-following models and basic engine-braking system simulation, CARLA includes sophisticated models of a vehicle's physical dynamics and a wide set of realistic and customizable driver profiles. Furthermore, it is specifically designed for autonomous driving simulation, supporting the creation of a realistic context and providing onboard sensors (e.g., cameras and LIDAR) suitable for testing AI-assisted autonomous driving. However, CARLA does not model any kind of networking technology. On the contrary, OMNeT++ provides a complete set of features and frameworks to simulate networks in detail but lacks built-in realistic vehicular mobility models. We thus aim to extract the best that CARLA and OMNeT++ can offer by coordinating their execution through some custom Application Programming Interfaces (APIs) specifically designed for this purpose. As a preliminary step, we configure CARLA to work in synchronous mode. This makes it possible for OMNeT++ to control CARLA's progress by enforcing OMNeT++'s own principal clock. We also use standard 5G network elements provided by the Simu5G [7] framework to faithfully mirror how the service would work in the real world.

The interaction between the two simulators is realized through a message-oriented middleware in charge of dispatching JSON-formatted messages³ between OMNeT++ and CARLA. This solution is inspired to the approach adopted by the Veins [27] framework, which implements the binary Traffic Control Interface (TraCI) of SUMO and the communication between OMNeT++ and SUMO is realized through TCP sockets. However, we design the communication relying on high-level messaging libraries such as ZeroMQ, rather than basic TCP sockets, and use the JSON message format, which allows us to easily extend and customize the set of APIs according to the specific scenarios. This solution decouples the implementation of the two simulators, and gives us the opportunity to develop the CARLA counterpart of the cross-simulator interface using a more flexible programming language such as Python.

In CARLA, we instantiate three types of actors: candidate lead vehicles; the tail vehicle; and other road traffic. For this purpose, we create a custom mobility module (dubbed CarlaInetMobility) that works with our custom APIs. The

²<https://www.unrealengine.com/>

³In the context of the co-simulation framework, the messages exchanged between OMNeT++ and CARLA are meant for event clock synchronization and for reading data/sending commands from/to CARLA. Application layer messages are entirely modeled in OMNeT++.

latter provides updates on the positions of these elements as they move in the CARLA environment so that their movements can be mirrored in the OMNeT++ environment.

For the lead vehicle, we implement behavior agents which build on CARLA's basic driver agent. This allows us to emulate different driving styles for different simulated human drivers. We supply the destination coordinates to a method of the behavior agent, which leverages the CARLA world object (with its global view of the map) in order to generate waypoints to the specified location. The lead vehicle sends its instantaneous position and bearing to the network at regular intervals. These data constitute waypoints, which must be transmitted to the tail vehicle so that it can follow them and thus trace the path of the lead vehicle to the final destination of the wireless towing service. To this end, we implement a custom follower navigation agent that tracks the motion of the lead vehicle. In order for the tail vehicle to smoothly follow the lead vehicle, we leverage the trajectory pruning algorithm proposed in [28]. We remark that our model of application layer messages is very general and more focused on payload size and sending frequency, rather than a specific message format. However, given the characteristic mentioned above, the application layer messages can be implemented using standardized ETSI Cooperative Awareness Messages (CAMs). The source code of the co-simulation library is available⁴

For each candidate lead vehicle, we specify a behavior agent exhibiting a given driving behavior: Extra-cautious (EC), Cautious (CS), Normal (NL) and Aggressive (AG). The characterisation of each is given by the parameters in Table 1: a more aggressive driver is more likely to drive near the speed limit, brake harder and more suddenly, as well as keep smaller safety distances. We remark that these are basic behavioral characteristics from which a driving behavior indicator χ can be derived based on external observations of speed evolution and variance, as explained in Section 4.

Specifically, the value of χ is determined for each candidate vehicle using Eq. (2), where data readings come from a 30 s behavior evaluation period. Conversely, η is readily available by comparing the engine power of the vehicle in distress and of the candidate lead vehicle. The model of the lead vehicles are randomly selected from a set of 10 real world vehicle models available in the CARLA library. We choose Tesla Model 3, from this set, to be the tail CAV. Based on χ and η , the decision agent applies its learned policy to select one of the cars/driving behaviors.

The selected car is the signaled to proceed to an agreed meeting point (MP), usually corresponding to the location of the vehicle to be towed, and shown in Fig. 1 as a grey circle. Given that the random number generator is seeded differently in each simulation run, behaviors are not completely deterministic, and may vary from round to round. Therefore, one behavior may map into a different indicator in a subsequent simulation. This reflects the fact that human drivers do not drive exactly in the same way

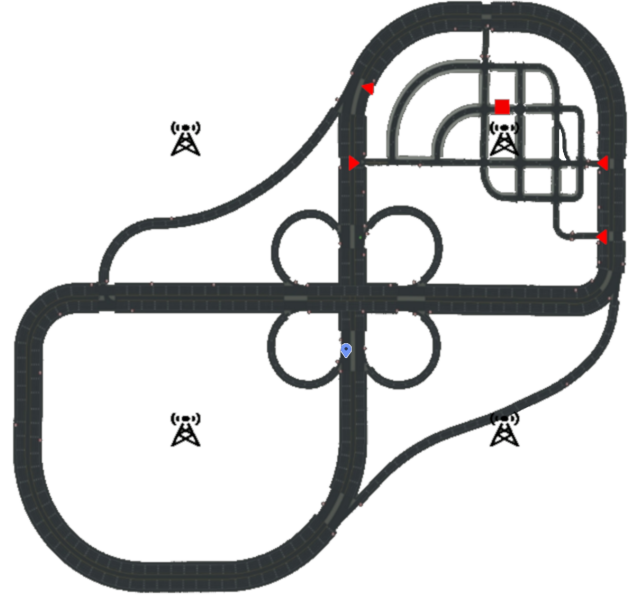


Figure 4: Simulation scenario with the map of the road and the location of the main communication network items. The red triangles represent the initial positions of the candidate lead vehicles before they respond to the towing alert from the network. The square represents the tail CAV with a driver in distress at the meeting point. The blue pin placeholder represents the intended towing destination. The sizes of the gNodeBs and vehicles have been exaggerated and the rest of the intervening traffic present in the simulation has been omitted for clarity.

Table 1
Driving behaviours of the lead vehicle

	EC	CS	NL	AG
Max. Speed [km/h]	30	40	50	70
Deviation from Speed limit [km/h]	10	6	3	1
Braking level [km/h]	15	12	10	8
Sudden braking time to collision [s]	5	3	3	3
Distance to object before avoidance [m]	15	12	10	8
Emergency braking distance [m]	10	6	5	4

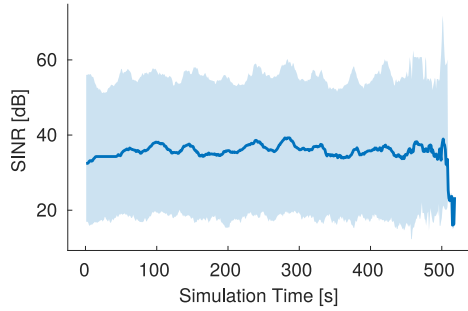
each time. The indicator χ still makes it possible to cover the resulting ample spectrum of behaviors. Table 2 gives an overview of other key simulation parameters.

The overlay of the 5G network in OMNeT++ over the road network in CARLA is shown in Fig. 4. During the simulation, the vehicle positions, speed and power in CARLA are sent to the corresponding OMNeT++ car elements through our custom API. The OMNeT++ vehicle elements then communicate to the 5G network elements and receive packets from the network which are then passed back to the CARLA actors through the API. In this way, the network service in OMNeT++ performs driver evaluation, and the decision regarding the selected lead vehicle can be communicated back to CARLA. Similarly, for the performance evaluation phase, the server in OMNeT++

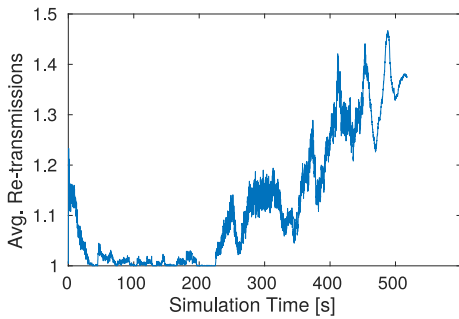
⁴<https://github.com/carlanet>

Table 2
Simulation Parameters

Parameter	Value
Path loss model	ITU Rural macro cell
Car UE transmit power	26 dBm
Car UE antenna gain	+0 dB
gNodeB antenna gain	+18 dB
gNodeB height	25 m
gNodeB transmit power	43 dBm
gNodeB scheduling discipline	Max CI
Size of uplink and downlink packet	1024 Bytes
Block error rate	1%
Handover latency (per vehicle)	$X \sim \mathcal{U}(0 \text{ ms}, 10 \text{ ms})$
Lead Vehicle reporting frequency	0.1 Hz
Number of intervening traffic (n)	30
Ideal spacing between lead and tail x_{ideal}	10 m
Ideal emergency towing time t_{ideal}	240 s



(a) Uplink SINR. The bold plot represents the median value. The ribbon around the plot indicates the interval between the 5th and the 95th percentiles of the values observed over 50 simulations.



(b) Moving average of HARQ triggered uplink packet re-transmissions for 50 simulation runs.

Figure 5: Network performance metrics.

evaluates the performance of the selected driving behavior and vehicle using data received from the actors in CARLA.

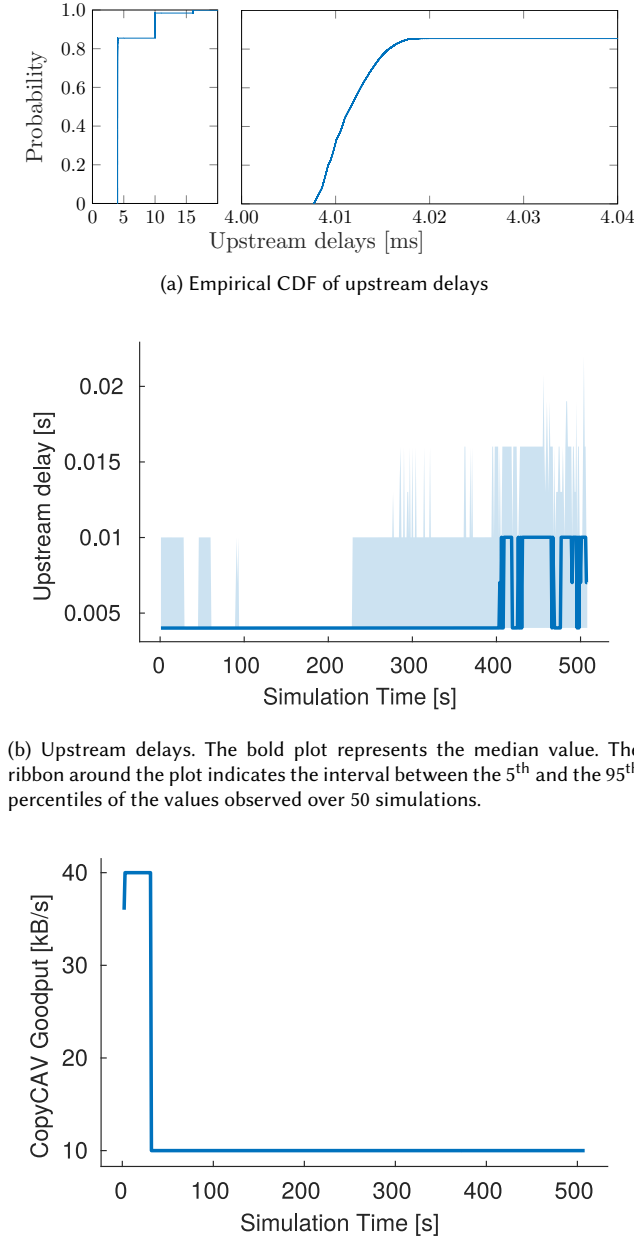
6. Results

We begin by considering network performance metrics resulting from 50 simulation runs presented in Fig. 5. Although the instantaneous values of the Signal to Interference and Noise Ratio (SINR) as shown in Fig. 5a exhibit wide variance ($< 20\text{dB}$ to 60dB), the overall median SINR remains fairly stable. This aligns with expectations for services running on the network edge. However, these instantaneous shifts in SINR impact the packet re-transmission rate as a result of forward error correction mechanisms using Hybrid ARQ. The latter is clearly depicted by Fig. 5b which combines the results of all 50 simulation runs and uses a window of 1000 samples to show the moving average of transmission attempts for a packet over the course of the simulations.

The left panel of Fig. 6a and Fig. 6b show that the upstream delay is bounded to a few tens of milliseconds, and most of the time it is below 20 ms. In fact the right panel of Fig. 6a shows that almost 85% of the values are in the order of 4 ms. Such low delay is achieved because the control of towing occurs in a server running at the edge of the cellular network. This kind of delay is compatible with human reaction times and corresponds to vehicle displacements of the order of a few tens of centimeters at legal speed, which explains how it is possible to automatically control a moving vehicle remotely.

Although the degradation of the channel, cf. Fig. 5b, leads to increases in the upstream delay for some packets with respect to the mainstream delay as shown in Fig. 6b, it has little impact on the expected goodput of the towing service shown in Fig. 6c. This result is explained by the small size of the packets (1 kB) and the low data rate (1 packet every 100 ms) required by CopyCAV. For descriptive purposes, the first 30 s of Fig. 6c includes the driver/vehicle evaluation phase, whereby different candidate vehicles transmit their driving data to the network for assessment. This evaluation phase is responsible for the higher goodput experienced over this period.

We now discuss the performance of the towing operation. The driving behavior of the tail CAV is closely aligned to that of the human-driven lead vehicle. This is clearly observed from Fig. 7a, which shows the median speed profiles of both the lead vehicle and the towed CAV. Speed variations are observed as the lead vehicle strives to maintain the ideal spacing x_{ideal} between itself and the tail vehicle. The spacing profile between the lead and tail vehicle depicted in Fig. 7b is therefore highly correlated with the speed tuning behavior. We observe that while $x_{\text{ideal}} = 10 \text{ m}$ the spacing achieved remains slightly larger. This phenomenon can be attributed to a variety of factors, the first of which is varying delay in communication (depicted in Fig. 6b) between the vehicles and network as a result of transients in channel quality as already discussed with reference to Fig. 5b. It is also due to a combination of the lead vehicle's driving behavior and of road conditions, including intervening traffic, intersections, and traffic lights.

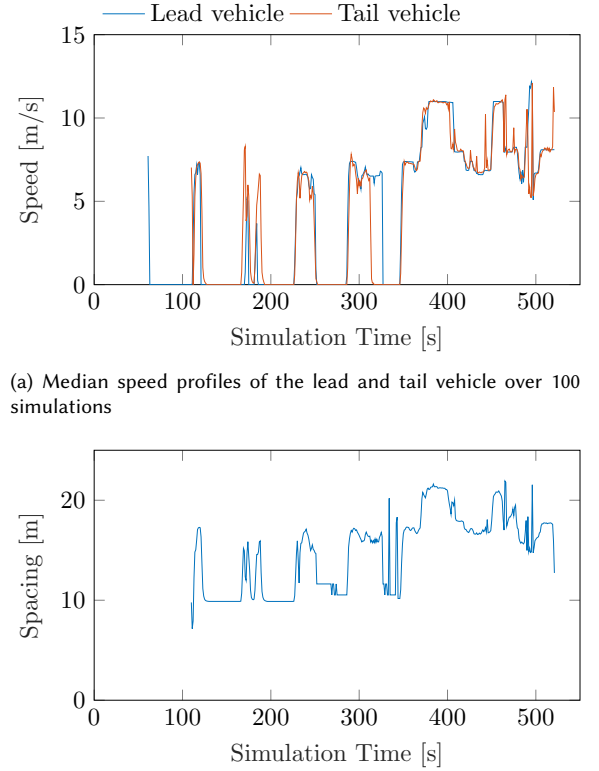


(b) Upstream delays. The bold plot represents the median value of 50 simulation runs. The ribbon around the plot indicates the interval between the 5th and the 95th percentiles of the values observed over 50 simulations.

Figure 6: System performance metrics.

For example, it can occur that the driving style or engine power of the leader are different than the tail vehicle's, so that the leader turns, accelerates or brakes more abruptly than the tail vehicle. In the worst case, the leader may proceed beyond a traffic light, while the tail vehicle is left behind. These conditions compound with intervening traffic, and typically lead to time-varying mismatches in the distance between the two vehicles.

We now comment on the performance of the reinforcement learning agent that selects the lead vehicle to execute the towing operation. We achieve this by showing how many times the policy promotes the selection of a given



(a) Median speed profiles of the lead and tail vehicle over 100 simulations

(b) Median spacing between lead and tow vehicle over 100 simulations

Figure 7: Speed and spacing between the leading and the towed vehicles. The driving behavior of the tail vehicle depends on the lead vehicle. In the wireless towing case, unlike platooning, the spacing between the vehicles varies, given that the tail vehicle simply follows the path paved by the lead vehicle, in place of a deterministic control law.

combination of driving behavior and lead/towed vehicle engine power ratio through the heat maps in Fig. 8. In the initial stages, the network agent has accrued limited experience, and thus it acts more randomly. This results in a more evenly spread out choice of driver behavior and engine power as shown in Fig. 8a. As the policy is progressively developed, see Fig. 8b, the agent chooses certain combinations of driving behavior and engine power with greater frequency and completely avoids others. This reflects the experience that the chosen combinations lead to towing completion time and vehicle spacing closer to the prescribed ideal values, and hence higher rewards.

When the policy is more established, as depicted in Fig. 8c, the agent acts more greedily and exhibits the highest preference in its selection. The weighted fair exploration mechanism [25] allows the agent to already start choosing actions that align to the eventual policy. This becomes apparent considering the high likelihood that the agent selects driving behavior $\chi = 1.0$ and $R_{rpm} \approx 1.67$ with a more developed policy as shown in Fig. 8. In particular, the selection policy leans more and more towards choosing stable driving behaviors ($\chi \rightarrow 1.0$) and more powerful vehicles ($R_{rpm} > 1.0$).

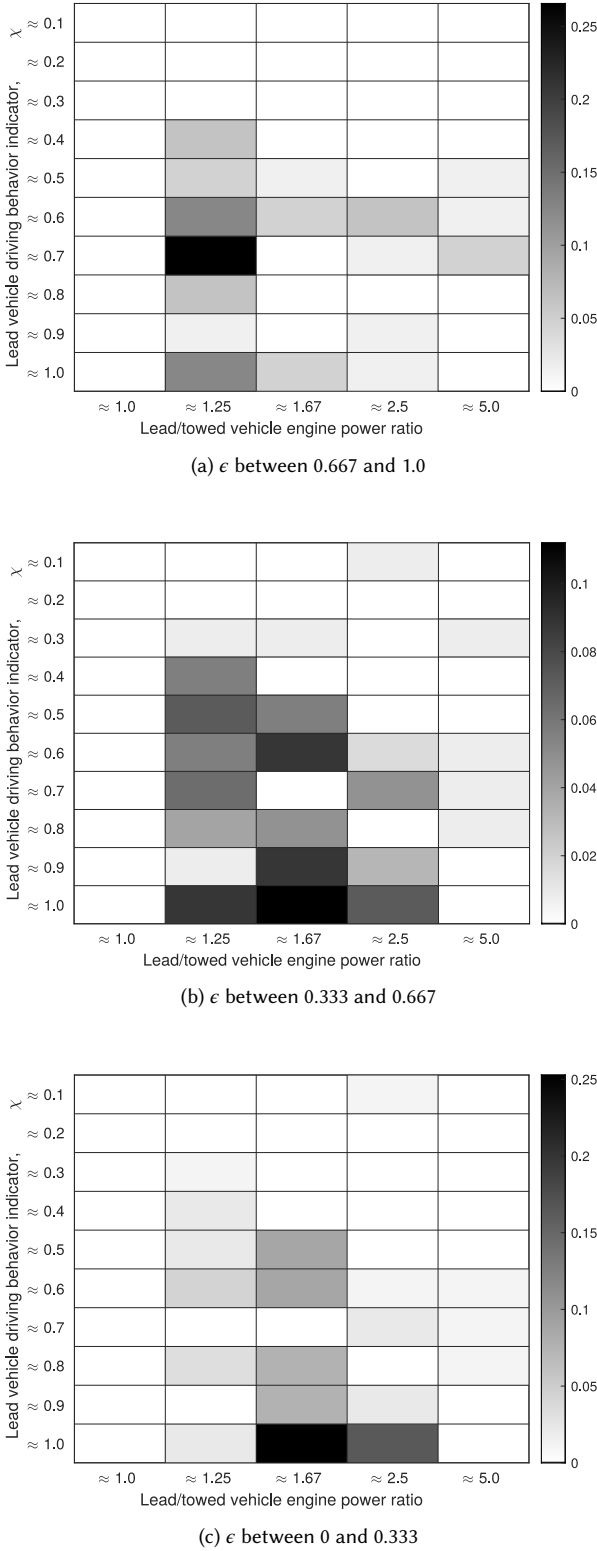


Figure 8: Joint distribution of selection probabilities at different levels of policy convergence.

We remark that during the initial period the agent acts more randomly ($\bar{\epsilon} \rightarrow 1.0$), and selects lead vehicles and driving behaviors that are not well matched to the

towed CAV. We show this occurrence through the speed profile distribution in Fig. 9. Here, panel (a) conveys the evolution of the speed of the lead vehicle, panel (b) that of the towed vehicle, and panel (c) depicts the evolution of the spacing between the two. The solid line represents the median value, whereas the shaded area around it conveys the interval between the 5th and 95th percentiles, both computed across 100 simulations with the selection agent acting randomly. As a consequence of adverse network conditions beyond the 225 s of simulation time (see. Fig. 5b) and compounded by this random choice of the lead vehicle, the speed and spacing of the tail vehicle exhibit significant dispersion, as shown in Fig. 9b and Fig. 9c. For example, Fig. 9a shows that between 300 s and 350 s of simulation time the median speed of the selected lead vehicle remains relatively high even though that of the tail vehicle in Fig. 9b drops to zero more than once. This imprudent choice in driving behaviour and lead vehicle consequently lead to greater dispersion in the spacing between the two vehicles as shown in Fig. 9c.

As the Copy-CAV selection agent gains more experience by reinforcement learning and acts according to policy ($\bar{\epsilon} \rightarrow 0$), it progressively selects the lead vehicle and driving behavior combination that better matches that of the tail CAV. This is illustrated in the panels of Fig. 10. For example, we observe that the tail vehicle is able to keep up with the leading vehicle as the latter better adapts to the autonomous driving style of the CAV. Moreover, in Fig. 10a, we observe a case where the lead vehicle stops at 334 s to allow the tail vehicle to catch up, since it had stopped earlier at 320 s (see Fig. 10b). The two vehicles then start moving again together from 346 s onwards. The latter case, in particular, shows that a better choice of the leading vehicle's power and driving behavior leads to generally better coordination during the towing process. This is further proven by a reduced dispersion in maximum spacing, which is over 100 m less in Fig. 10c when compared to Fig. 9c. These results show the effectiveness of our reinforcement learning algorithm in the selection of the most suitable driving behaviour and lead vehicle for the towing service.

7. Conclusions

In this paper, we have presented the design of an emergency wireless towing service that relies on a human-driven vehicle to help the passenger of a connected autonomous vehicle (CAV) in distress. Similar to a physical towing scenario, here a towed CAV relies on a leading vehicle to decide on the route towards a given destination, and automatically follows the lead vehicle's trail. The management of potential lead vehicles and the consequent lead vehicle as well as the the CAV is reliant on the 5G communications infrastructure and its edge computing fabric.

The presence of a lead human driver ensures that crucial intuitive decisions can be made, as they come naturally to a human driver but may not feature in the training

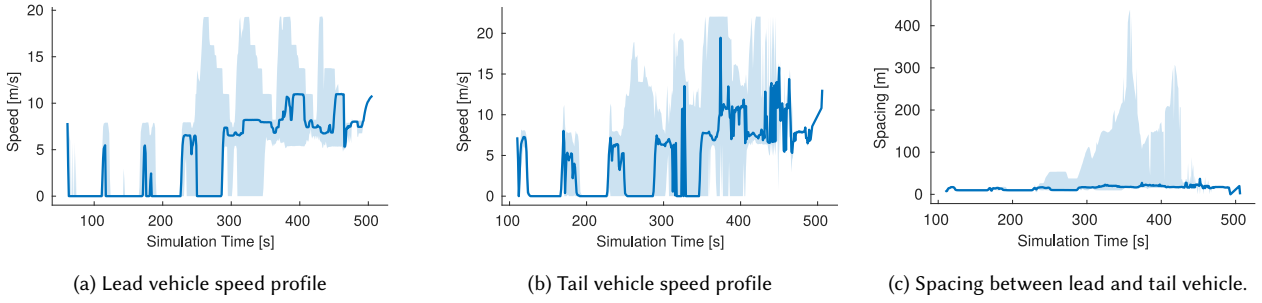


Figure 9: Distribution of speed profiles and spacing for off-policy selection ($\bar{\epsilon} = 0.965$). The bold plot represents the median value. The ribbon around the plots indicates the interval between the 5th and the 95th percentiles of the values observed over 100 simulations.

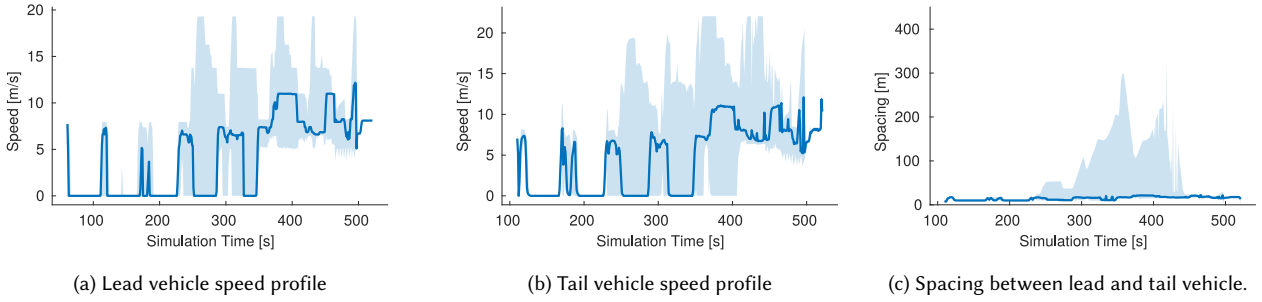


Figure 10: Distribution of speed profiles and spacing for on-policy selection ($\bar{\epsilon} = 0.134$). The bold plot represents the median value. The ribbon around the plots indicates the interval between the 5th and the 95th percentiles of the values observed over 100 simulations.

process of a typical CAV. We have shown that, unlike platooning, the wireless towing service is robust to changing network conditions given its low data rate requirement and the flexible speed and spacing that it can tolerate.

We have also presented a lead vehicle/driving behavior selection algorithm that leverages Monte-Carlo reinforcement learning to improve the performance of the wireless towing service over time. With this scheme, the cohesiveness between the tail CAV and lead vehicle is better maintained reducing the chances of other traffic getting in between the two and disrupting the service.

We also observe that transient drops in quality of the wireless channel to the edge network has a non-trivial effect on the cohesion between the towed and lead vehicle. As part of future work, we intend to incorporate V2V communication between the vehicles as a backup channel to solve this problem. We will also investigate the robustness of our towing scheme when network resources are congested or constrained.

Acknowledgment

This work received support from the Spanish State Research Agency (AEI) under the aegis of PID2019-109805RB-I00/AEI/10.13039/501100011033. It has been partially supported by the Project AEON-CPS (TSI-063000-2021-38), funded by the Spanish Ministry of Economic Affairs and Digital Transformation and the NextGeneration-EU in the framework of the Spanish Recovery, Transformation and Resilience Plan. Christian Quadri was supported by project

SERICS (PE00000014) under the MUR NRRP funded by the European Union NextGenerationEU.

CRedit authorship contribution statement

Constantine Ayimba: Conceptualization, Methodology, Investigation, Software, Data Curation, Visualization, Writing-Original draft preparation, Writing- Reviewing and Editing, Software. **Valerio Cislighi:** Software, Validation, Writing-Original draft preparation. **Christian Quadri:** Software, Validation, Writing- Reviewing and Editing. **Paolo Casari:** Visualization, Conceptualization, Validation, Writing- Reviewing and Editing, Supervision. **Vincenzo Mancuso:** Funding Aquisition, Conceptualization, Validation, Writing- Reviewing and Editing, Supervision.

References

- [1] J. Yonghoon, K. Tae Han, L. Sun Young, R. Young Sun, H. Ki Jeong, S. Kyoung Jun, S. Sang Do, Association of transport time interval with neurologic outcome in out-of-hospital cardiac arrest patients without return of spontaneous circulation on scene and the interaction effect according to prehospital airway management, *Clin Exp Emerg Med* 9 (2) (2022) 93–100. doi:10.15441/ceem.21.074.
- [2] C. Johnson, Readiness of the road network for connected and autonomous vehicles, https://www.racfoundation.org/wp-content/uploads/2017/11/CAS_Readiness_of_the_road_network_April_2017.pdf, accessed: 2023-2-13 (2017).

- [3] A. Ng, Self-driving cars won't work until we change our roads and attitudes, <https://www.wired.com/2016/03/self-driving-cars-won-t-work-change-roads-attitudes/>, accessed: 2022-10-07 (2016).
- [4] A. Ng, When will self-driving cars be available to consumers, <https://www.quora.com/When-will-self-driving-cars-be-available-to-consumers/answer/Andrew-Ng>, accessed: 2022-10-07 (2016).
- [5] A. Virdis, G. Nardini, G. Stea, A framework for mec-enabled platooning, in: 2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW), 2019, pp. 1–6. doi:10.1109/WCNCW.2019.8902910.
- [6] C. Quadri, V. Mancuso, M. Ajmone Marsan, G. P. Rossi, Edge-based platoon control, *Computer Communications* 181 (2022) 17–31. doi:https://doi.org/10.1016/j.comcom.2021.09.021.
- [7] G. Nardini, D. Sabella, G. Stea, P. Thakkar, A. Virdis, Simu5g—an omnet++ library for end-to-end performance evaluation of 5g networks, *IEEE Access* 8 (2020) 181176–181191. doi:10.1109/ACCESS.2020.3028550.
- [8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: An open urban driving simulator, in: Proc. 1st Annual Conference on Robot Learning, Vol. 78 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 1–16.
- [9] High-Level Expert Group on Artificial Intelligence, Ethics guidelines for trustworthy AI, <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>, accessed: 2023-2-09 (2019).
- [10] C. Ayimba, V. Sciancalpore, P. Casari, V. Mancuso, I move u move: V2x-enabled wireless towing, in: IEEE WCCI 2022 - IEEE World Congress on Computational Intelligence (WCCI AI6G WORKSHOP), 2022, pp. 1–9.
- [11] S. J. Park, S. Hong, D. Kim, I. Hussain, Y. Seo, Intelligent in-car health monitoring system for elderly drivers in connected car, in: Proc. 20th Congress of the International Ergonomics Association (IEA 2018), Springer, 2019, pp. 40–44. doi:10.1007/978-3-319-96074-6_4.
- [12] Minea, Marius and Dumitrescu, Cătălin Marian and Costea, Ilona Mădălina, Advanced e-call support based on non-intrusive driver condition monitoring for connected and autonomous vehicles, *Sensors* 21 (24) (2021) 1–29. doi:10.3390/s21248272.
- [13] G. Sikander, S. Anwar, Driver fatigue detection systems: A review, *IEEE Transactions on Intelligent Transportation Systems* 20 (6) (2019) 2339–2352. doi:10.1109/TITS.2018.2868499.
- [14] S. Moon, I. Moon, K. Yi, Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance, *Control Engineering Practice* 17 (4) (2009) 442–455. doi:https://doi.org/10.1016/j.conengprac.2008.09.006.
- [15] S. Magdici, M. Althoff, Adaptive cruise control with safety guarantees for autonomous vehicles, *IFAC-PapersOnLine* 50 (1) (2017) 5774–5781, 20th IFAC World Congress. doi:10.1016/j.ifacol.2017.08.418.
- [16] B. Groelke, C. Earnhardt, J. Borek, C. Vermillion, A predictive command governor-based adaptive cruise controller with collision avoidance for non-connected vehicle following, *IEEE Transactions on Intelligent Transportation Systems* 23 (8) (2022) 12276–12286. doi:10.1109/TITS.2021.3112113.
- [17] M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. L. Cigno, F. Dressler, Toward communication strategies for platooning: Simulative and experimental evaluation, *IEEE Transactions on Vehicular Technology* 64 (12) (2015) 5411–5423. doi:10.1109/TVT.2015.2489459.
- [18] T. Hades, C. Sommer, Dynamic platoon formation at urban intersections, in: 2019 IEEE 44th Conference on Local Computer Networks (LCN), IEEE, 2019, pp. 101–104. doi:10.1109/LCN44214.2019.8990846.
- [19] Constantine Ayimba and Michele Segata and Paolo Casari and Vincenzo Mancuso, Driving under influence: Robust controller migration for MEC-enabled platooning, *Computer Communications* 194 (2022) 135–147. doi:https://doi.org/10.1016/j.comcom.2022.07.014.
- [20] X. Cheng, Z. Huang, S. Chen, Vehicular communication channel measurement, modelling, and application for beyond 5G and 6G, *IET Communications* 14 (19) (2020) 3303–3311. doi:10.1049/iet-com.2020.0531.
- [21] R. Coppola, M. Morisio, Connected car: Technologies, issues, future trends, *ACM Comput. Surv.* 49 (3) (oct 2016). doi:10.1145/2971482.
- [22] R. S. Sutton, A. G. Barto, Introduction to Reinforcement Learning, 2nd Edition, MIT Press, Cambridge, MA, USA, 2020. URL <http://incompleteideas.net/book/RLbook2020.pdf>
- [23] R. K. Jain, D.-M. W. Chiu, W. R. Hawe, et al., A quantitative measure of fairness and discrimination, Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA (1984).
- [24] D. Vysochanskij, Y. I. Petunin, Justification of the 3σ rule for unimodal distributions, *Theory of Probability and Mathematical Statistics* 21 (25-36) (1980).
- [25] C. Ayimba, P. Casari, V. Mancuso, SQLR: Short-Term Memory Q-Learning for Elastic Provisioning, *IEEE Transactions on Network and Service Management* 18 (2) (2021) 1850–1869. doi:10.1109/TNSM.2021.3075619.
- [26] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner, Microscopic traffic simulation using SUMO, in: IEEE International Conference on Intelligent Transportation Systems, 2018. doi:10.1109/ITSC.2018.8569938.
- [27] C. Sommer, R. German, F. Dressler, Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis, *IEEE Transactions on Mobile Computing (TMC)* 10 (1) (2011) 3–15. doi:10.1109/TMC.2010.133.
- [28] M. Potamias, K. Patroumpas, T. Sellis, Sampling trajectory streams with spatiotemporal criteria, in: 18th International Conference on Scientific and Statistical Database Management (SSDBM'06), 2006, pp. 275–284. doi:10.1109/SSDBM.2006.45.