Scheduling Out-of-Coverage Vehicular Communications Using Reinforcement Learning

Taylan Şahin ⬥, Ramin Khalili, Mate Boban ⬥, *Senior Member, IEEE*, and Adam Wolisz ⬥, *Senior Member, IEEE*

Abstract— **Performance of vehicle-to-vehicle (V2V) communications depends highly on the employed scheduling approach. While centralized network schedulers offer high V2V communication reliability, their operation is conventionally restricted to areas with full cellular network coverage. In contrast, in out-of-cellular-coverage areas, comparatively inefficient distributed radio resource management is used. To exploit the benefits of the centralized approach for enhancing the reliability of V2V communications on roads lacking cellular coverage, we propose VRLS (Vehicular Reinforcement Learning Scheduler), a centralized scheduler that proactively assigns resources for out-of-coverage V2V communications *before* vehicles leave the cellular network coverage. By training in simulated vehicular environments, VRLS can learn a scheduling policy that is robust and adaptable to environmental changes, thus eliminating the need for targeted (re-)training in complex real-life environments. We evaluate the performance of VRLS under varying mobility, network load, wireless channel, and resource configurations. VRLS outperforms the state-of-the-art distributed scheduling algorithm in zones without cellular network coverage by reducing the packet error rate by half in highly loaded conditions and achieving near-maximum reliability in low-load scenarios.**

Index Terms—**Out of coverage, reinforcement learning, resource management, scheduling, V2V.**

I. INTRODUCTION

COOPERATIVE awareness, defined as knowledge of the location, speed, and bearing of surrounding vehicles, forms a basis for most traffic safety and efficiency applications [1], [2]. Such awareness can be best achieved in real time using vehicle-to-vehicle (V2V) communication utilizing a dedicated set of radio resources. However, achieving a collision-free transmission under a highly dynamic vehicular environment is a challenging task.

Recent work in the 3rd Generation Partnership Project (3GPP) defines a solution where resources for V2V communication are efficiently coordinated by a base station (BS) of a cellular network, resulting in highly reliable transmissions [3], [4].

However, V2V transmissions should not rely completely on ubiquitous availability of the cellular network coordination, as there will always exist coverage gaps, either because of physical impediments (e.g., tunnels, blockage of a link by large objects such as buildings, etc.) or limited infrastructure deployment.

Existing solutions for scheduling V2V communications outside the network coverage are based on distributed methods, where vehicles select resources autonomously. 3GPP defines such a resource allocation method (named "mode 4" in Long Term Evolution (LTE) [5], and "mode 2" in the fifth-generation (5G) New Radio (NR) [6]), which is based on sensing of the radio resources by the vehicles. Another solution is defined by the IEEE standard 802.11p [7] (and its successor 802.11bd [8]) based on carrier-sensing multiple access. Although distributed solutions do not require a network infrastructure, they suffer from the hidden node problem due to the limited local view, which results in reduced quality of service [9]–[11]. Given that the use cases require consistently high reliability and low latency of V2V communication to be maintained irrespective of coverage [4], the intermittent coverage poses one of the key problems in assuring proper V2V connectivity.

In this paper, we study an alternative approach to the V2V resource scheduling in the out-of-coverage (OOC) zones. We propose a centralized scheduler residing in the network, comprising a reinforcement learning (RL) *agent* that *pre-assigns* resources to the vehicles for their V2V transmissions in an OOC area, before the vehicles enter such an area. We are motivated to use an RL-based approach due to its successful applications for resource allocation problems in general [12], and for vehicular communications in particular [13]. The scheduler, which we call VRLS (Vehicular Reinforcement Learning Scheduler), was proposed and demonstrated to be potentially efficient in [14]. VRLS design addresses two practical challenges. First, training and deploying RL solutions in the real world is costly in terms of training. In particular:

- Training an RL agent in a real-world setting is considerably slower than training it in a simulated environment, because of the limited availability of data samples.
- Collecting data from an actual vehicular network is expensive, or might not be even possible considering the additional signaling and processing overhead it incurs.
- Any undesirable outcomes of an RL agent still under training might threaten the safety-critical V2V use cases.

Second, in real-world problems, it is likely that the conditions in a given environment, such as road traffic mobility or data traffic load in a V2V communication network, would change

over time. It would be impractical to re-design, re-train, and re-evaluate a new RL solution every time the environment changes, even if this change is substantial. Therefore, a single RL-based solution should be applicable to varying conditions in the environment.

By defining a unified state representation, we have designed VRLS to be applicable to road sections without network coverage, which can have arbitrary size and number of lanes, any number of vehicles, utilizing any resource pool configuration with different number of resources in time and frequency. VRLS consists of a deep neural network, which processes the state of the environment and decides on the resource allocation for vehicles as an outcome of its trained policy. The reward provided to the RL agent during its training is designed to maximize the reliability of V2V transmissions. In this paper, we extend the basic VLRS design with the following key contributions:

1) We train VRLS in simplified and simulated vehicular environments, and show that it can be deployed without further training in realistic, complex environments, varying in terms of mobility, wireless channel characteristics, OOC area size, network load, and traffic.

2) We evaluate the performance of VRLS in terms of reliability, user fairness, packet inter-reception time, and latency, as well as the impact of network quality of service on V2V applications, in comparison to the state-of-the-art distributed scheduler mode 4. In terms of reliability, VRLS reduces by half the packet loss of mode 4 in highly loaded conditions, and performs close to the theoretical maximum in low-load scenarios. Further, VRLS does not compromise on fairness across the vehicular users, while achieving similar latency and higher "mutual awareness" (defined in Section V-A) as compared to mode 4.

3) Considering that the network might need to operate differently configured resource pools in terms of the number of resources in time and frequency, e.g., to support different V2V services, we show that VRLS can be trained across multiple predetermined resource configurations at once to support any of them by learning a single policy.

In the rest of the paper, we first present the background and the related work in Section II. Section III defines our system model and the problem we address. We elaborate on our VRLS design in Section IV. After describing our evaluation methodology in Section V, we present our performance evaluation and analysis in Section VI, and the training performance of VRLS in Section VII. Finally, Section VIII concludes the paper.

II. BACKGROUND AND RELATED WORK

We first present the challenges of V2V communications in terms of performance requirements in Section II-A. To satisfy these requirements, radio resource allocation plays a critical role, as we elaborate in Section II-B. The centralized resource allocation mode introduced in cellular networks by 3GPP shows clear performance benefits over the distributed methods. We review the distributed resource allocation mechanism specified by 3GPP in Section II-C, which we take as a baseline in our evaluations. Finally, we introduce the centralized, RL-based

approach to resource allocation for V2V communications in areas delimited by cellular network coverage, and identify its difference from the prior art in Section II-D.

A. Intelligent Transport Systems and V2V Communications

V2V communications enable road safety and traffic efficiency via intelligent transport system (ITS) applications: autonomous driving and vehicle platooning, to name a few [1], [2]. To achieve this, the V2V network has to satisfy the performance requirements of target use cases, most notably the reliability, latency, and throughput. The reliability requirements indicate the maximum tolerable packet loss rate between vehicles, together with a maximum application-level latency to be supported. LTE V2V use cases typically require 80–90% reliability with 100 ms latency, targeting various basic road traffic safety applications [1], [3]. On the other hand, advanced use cases to be supported by 5G NR, such as platooning vehicles with a high degree of automation, require up to 99.99% of reliability with 10 ms latency [2]. Such requirements assume an abundance of radio resources available for V2V communication. However, under realistic conditions, where the availability of radio resources is limited, achieving high reliability becomes difficult. Under such conditions, resource allocation plays a vital role in the performance of V2V applications.

B. Radio Resource Allocation for V2V Communications

In the context of the 3GPP standard, V2V communications utilize a *resource pool*, which consists of a limited number of time-frequency resources configured by the network. Assignment of these resources to the vehicles in a reliable, timely, and efficient manner becomes a critical task in order to satisfy the demanding requirements of vehicular applications.

Conventionally, the resource allocation task can be performed either in a centralized or distributed way. The authors in [9] and [15] show that centralized resource allocation enables higher road traffic efficiency and safety, as compared to distributed scheduling mechanisms. To illustrate, the centralized scheduling mode in LTE (called "mode 3" [6]) can achieve a shorter inter-vehicle distance than LTE mode 4 and IEEE 802.11p in a high-density platoon with a guaranteed crash rate $\leq 1\%$ [9]. This is enabled by higher reliability and shorter latency of the V2V transmissions via efficient resource reuse that cannot be achieved by distributed scheduling. Despite providing very low latency, 802.11p is shown to suffer from the increasing collisions with the load [9], and mode 4 is prone to collisions due to re-selection or persistent usage of the same resources by the vehicles in proximity [16], [17]. Irrespective of the underlying technology, the reduced reliability degrades the performance of the V2V use cases (e.g., a platoon can either have a larger inter-vehicle spacing or can support fewer vehicles) [18].

C. Distributed Scheduler LTE Mode 4

If the centralized scheduling of V2V resources with the help of the cellular network fails due to the loss of network connectivity, vehicles need to switch to the distributed resource
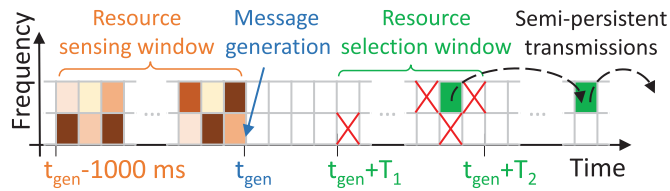
Fig. 1. Illustration of the scheduling mechanism of LTE mode 4. Resources sensed with large power (represented with darker colors) are excluded from the selection (crossed). The selected resource (green) among the remaining ones is utilized to transmit semi-persistently.

allocation mode. Specifically, upon experiencing a connection interruption long and often enough, vehicles stop using their network-scheduled resources and resort to a random resource selection procedure until the sensing results needed to operate LTE mode 4 become available [6], [19].

In mode 4, vehicles autonomously make semi-persistent resource (re-)selections from the configured resource pool for V2V communications, based on sensing [20], [21]. Specifically, as illustrated in Fig. 1, upon a message generation at time t_{gen}, the vehicle selects a single resource from an upcoming resource selection window between $t_{\text{gen}} + T_1$ and $t_{\text{gen}} + T_2$. The vehicle can transmit using the selected resource on a periodic basis for C_{resel} times, i.e., semi-persistently, where C_{resel} takes a random value from a predefined interval. After C_{resel} transmissions, the vehicle makes a new resource re-selection with probability $1 - P_{\text{keep}}$, otherwise keeping the same resource by setting a new random value for C_{resel}. Each resource (re-)selection is based on the sensing results of the past 1000 ms from t_{gen}, excluding the subframes the vehicle transmitted, as no sensing was conducted due to the half-duplex radio constraint (vehicles can either transmit or receive, but not both at the same time). The vehicle further excludes the resources where it sensed an average received power larger than a predefined threshold $\text{Thr}_{\text{sense}}$. It sorts the remaining resources with respect to their average Received Signal Strength Indicator (RSSI), and selects a resource randomly from the top 20% (the lowest RSSI).

The sensing mechanism enables vehicles to find a "free" resource, or, in case of heavy resource use, a resource with less interference. On the other hand, the randomization aims at mitigating the persistent resource conflicts due to multiple vehicles continuously selecting the same resource. Nonetheless, since sensing measurements are limited in time and space, vehicles in mode 4 are prone to the well-known hidden-node problem. To illustrate, if vehicles far apart cannot sense each other's transmissions and select the same resource, their transmissions can interfere on a receiver located between them.

Several works have focused on improving the performance of mode 4. Most of them target the persistent resource collision problem. The authors in [22] and [23] propose reserving resources at each resource selection instance and alternately using them to mitigate the collision probability. Other works propose exchange of information among the vehicles, such as channel measurements [24], location [25], or status and reservation information of resources [26]–[28]. Revisions to the

sensing mechanism are also proposed in several works, mainly by considering different weighting strategies for selecting the resources [29], [30]. Overall, the proposed extensions to mode 4 either require additional signaling or increase resource occupancy, hence making them less efficient from resource utilization point of view. Further, the parameters of the mode 4 algorithm, as well as the extended methods, require careful tuning in order to achieve the desired performance.

D. RL-Based Approach for V2V Radio Resource Allocation

RL has recently gained significant attention in vehicular networks, with applications ranging from storage of data [31] to controlling of information flow in sensing networks [32], thanks to its ability to handle tasks in dynamic, time-varying environments. The resource allocation task challenges the mainstream approach of formulating an optimization problem and solving it (sub-)optimally, because of the highly dynamic nature of vehicular networks [13]. RL offers an alternative solution, which interacts with, and adapts its actions to the unknown environment. Further, sequential decision-making encountered in resource allocation tasks is a native functionality of RL [33].

The majority of the works applying RL to the resource allocation problem in vehicular networks consider joint optimization of V2V and cellular links [34]–[36]. The authors in [37] utilize V2V links to cooperatively relay the cellular downlink data within a cluster of vehicles, with the aid of RL-based scheduling. RL is also applied to distributed resource allocation mechanisms [38], [39]. While RL-based approaches are shown to solve resource allocation tasks efficiently, the cost of such solutions comes in terms of an additional effort required for training prior to deployment.

We have first applied RL for managing the V2V resources centrally in [40], motivated by the above-mentioned works. We later proposed VRLS in [14], where we demonstrated its ability to avoid resource conflicts, and to reuse resources more efficiently as compared to the state-of-the-art V2V scheduling algorithms, in a basic OOC setting and under different resource pool configurations. Compared to the previous works, our approach differs by proposing a *centralized*, *RL-based approach* for managing the resources for V2V communications *outside the network coverage*, whereby resources are provided to the vehicles before they leave the coverage.

III. SYSTEM MODEL

A. Vehicular Network Environment

We consider a vehicular network where vehicles broadcast V2V messages with periodic and event-triggered, i.e., aperiodic, traffic. A typical example of periodic traffic is the regular broadcast of vehicle information such as position and speed, as in cooperative awareness messages (CAMs) [41], whereas aperiodic traffic is triggered on events to warn vehicles such as of an accident, as in decentralized environmental notification messages (DENMs) [42]. The V2V messages are assumed to have a fixed size of S_{msg}, requiring a predefined number of time-frequency resources based on the modulation and coding
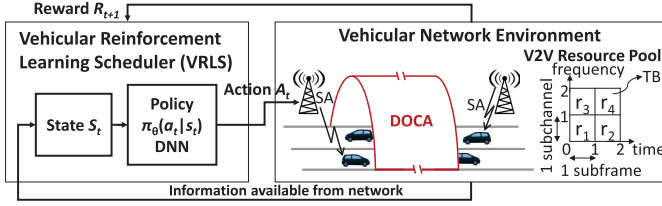
Fig. 2. VRLS applied to our vehicular network environment, which is a delimited out-of-coverage area (DOCA). Vehicles communicate with each other in a DOCA using the resources indicated by the scheduling assignments (SAs) sent by the delimiting BSs before they enter the DOCA.

scheme (MCS). In the case of the periodic traffic, the messages are generated with a fixed periodicity of T_{msg} by all vehicles. For the event-triggered traffic, the event arrivals are assumed to follow a Poisson process with an arrival rate of X_{evt} per s for each vehicle [3], upon which a single message is generated.[1]

We focus on V2V communications in a delimited out-of-coverage area (DOCA). A DOCA can have any size and shape depending on the scenario (e.g., two-dimensional in urban and one-dimensional on highway). We assume a DOCA that consists of a two-way highway segment of length L_{DOCA} outside the coverage of BSs at its two ends, as illustrated in Fig. 2. Vehicles of length L_{veh} are assumed to travel on J lanes per direction. The BSs are assumed to be aware of the existence of the DOCA, as well as its location, length, etc.

B. Time-Frequency Resources

In line with the LTE and NR specifications, V2V communications outside the coverage utilize a dedicated pool of time-frequency resources configured by the network. We denote pool configuration by $C^{K \times M}$, where K is the number of subchannels and M is the number of subframes that the pool contains, over the frequency and the time domains, respectively, repeating with a periodicity of T_{msg}. Fig. 2 illustrates an example resource pool configuration with 2 subchannels and 2 subframes, i.e., $C^{2 \times 2}$.

Following the LTE V2V assumptions [3], a transmission of a single V2V message occupies a single time-frequency resource $r \in C^{K \times M}$, which is referred to as a transmission block (TB). A TB consists of a single subframe of 1 ms, and a single subchannel containing a sufficient number of resource blocks (RBs) to carry the message. No redundant transmissions are considered. Consequently, the latency of a successfully received message is bounded by the time-length of the pool, i.e., the number of subframes, to a maximum of M ms.

We assume half-duplex radios, that is, vehicles transmitting at the same time are not able to receive each other's message (e.g., TBs r_1 and r_3 sharing the same subframe in Fig. 2) [45].

---

[1] According to the ETSI specifications [41], [42], while periodic by default, the periodicity of CAM transmissions can be adjusted depending on vehicular mobility and traffic load, and DENM transmissions might contain bursts of several periodic messages. We evaluate the impact of the different modes of traffic separately, by following previous research work (e.g., [43], [44]) and applying the traffic models proposed in 3GPP [3]. This approach makes the analysis easier and enables us to generalize our results to any type of V2V traffic beyond the CAM and DENM applications.

We refer to the unsuccessful reception of the messages due to the HD limitation as *HD error* or *conflict*, and the relation among the TBs in the same subframe causing this phenomenon, as *HD constraint*, in the paper. When no HD errors occur, successful decoding of a message is further conditioned on the signal-to-interference-plus-noise ratio (SINR) measured for the corresponding TB at the receiver. The SINR of a single transmission at receiver j from transmitter i is:

$$\mathrm{SINR}_{ij} = \frac{P_{\mathrm{Tx}} |h_{ij}|^2}{\sigma^2 + \sum_{l=1, l \neq i}^{L} P_{\mathrm{Tx}} |h_{lj}|^2}, \qquad (1)$$

where P_{Tx} is the transmit power of the transmitter, and $|h_{ij}|^2$ denotes the channel coefficient between the transmitter i and the receiver j, which accounts for the path loss and fading effects of the wireless channel on the transmitted signal. σ^2 is the noise power, and the summation term in the denominator denotes the interference due to the other vehicles $l = 1, \ldots, L$ using the same TB as i. The SINR depends on the interference level of the other transmissions using the same TB, under the path loss and fading effects of the propagation channel. Messages transmitted by different vehicles using the same TB (i.e., the same subframe *and* the same subchannel) may interfere, depending on the propagation conditions, and lead to decoding errors at the receiver, which we refer to as *collision errors*. However, spatial reuse of the same TB or the TBs sharing the same subframe is possible among sufficiently far apart transmitters, without creating any collision and HD errors, respectively. Further, unsuccessful receptions could also result from the channel effects, e.g., due to path loss and shadowing that considerably reduce the received SINR (or SNR), which we refer to as *propagation errors*.

C. Problem Definition

In the absence of a cellular network assigning the resources for V2V communications, as in the case of a DOCA, vehicles can resort to distributed resource scheduling mechanisms based on sensing or using resources randomly [6]. However, such distributed resource allocation methods result in degraded performance of V2V communications as compared to the centralized allocation of resources within the network coverage, given the same limited number of resources available [9], [10].

We argue that the areas without network coverage, i.e., DOCAs, are usually well known to the network (such as via map information providing the location of a tunnel or detecting link interruptions by measured signals at a certain location). Therefore, it is possible to create a centralized scheduler that "pre-schedules" the resources for the DOCA. The centralized scheduling entity is assumed to exchange information with the BSs delimiting the DOCA, where it can gather the information needed for scheduling. In turn, the BSs forward the scheduling decisions to the vehicles via scheduling assignments (SAs) before the vehicles enter the DOCA. An SA indicates the resource to use from the next available resource pool period, every time the vehicle generates a message during its travel in the DOCA. Accordingly, the task is to assign a resource r from the resource pool to each vehicle going into the DOCA,

where the vehicle uses the assigned resource for all its V2V transmissions throughout its journey.

Our objective is to find a centralized radio resource management algorithm that maximizes the reliability of the V2V transmissions in the DOCA. To quantify the reliability, we use the packet reception ratio (PRR) metric as defined in the 3GPP standard [3]. For a single message transmitted from vehicle i, the PRR is calculated by X_i/Y_i, where Y_i is the number of vehicles located within the range (a, b) from the transmitter, and X_i is the number of vehicles with successful reception among Y_i. The average PRR is then calculated for a series of messages consecutively transmitted by all the vehicles in the environment as $(\sum_{i=1}^{V} \sum_{n=1}^{N_i} X_{i,n})/(\sum_{i=1}^{V} \sum_{n=1}^{N_i} Y_{i,n})$ with N_i denoting the number of generated messages by vehicle i, and V is the number of vehicles [3]. Assuming a resource pool configuration of $C^{K \times M}$, we would like to find a resource assignment (k, m) for each vehicle i that maximizes the PRR of all vehicles, which can be formally expressed as:

$$\underset{\{\alpha_{k,m}^i\}_{\forall i}}{\operatorname{argmax}} \sum_{i=1}^{V} \sum_{n=1}^{N_i} X_{i,n}, \text{ where } X_{i,n} = \sum_{j=1}^{Y_{i,n}} \eta_{ij}, \qquad (2)$$

$$\text{s.t.}$$

$$\eta_{ij} = \begin{cases} 1 & \text{if } \frac{P_{\text{Tx}}|h_{ij}|^2 \alpha_{k,m}^i \beta_m^j}{\sigma^2 + \sum_{l=1, l \neq i}^{V} P_{\text{Tx}}|h_{lj}|^2 \alpha_{k,m}^l} \geq \gamma, \\ 0 & \text{otherwise.} \end{cases} \qquad (2a)$$

$$\alpha_{k,m}^i = \begin{cases} 1 & \text{if } i \text{ using subchannel } k \text{ and subframe } m, \\ 0 & \text{otherwise.} \end{cases} \qquad (2b)$$

$$\beta_m^j = \begin{cases} 1 & \text{if } j \text{ using subframe } m, \\ 0 & \text{otherwise.} \end{cases} \qquad (2c)$$

$$\sum_{m=1}^{M} \sum_{k=1}^{K} \alpha_{k,m}^i = 1, \forall i. \qquad (2d)$$

$$0 < P_{\text{Tx}} \leq P_{\max} \qquad (2e)$$

In the above formulation, η_{ij} is a binary variable indicating whether the reception of a single message n at receiver j, transmitted from i is successful or not. Successful reception is conditioned on the SINR_{ij} being above a target threshold γ, subject to collisions and the HD constraint represented by the following variables. $\alpha_{k,m}^i$ is a binary variable denoting whether vehicle i is transmitting using the subchannel k and subframe m, and β_m^j imposes the HD constraint on vehicle j if it is transmitting on the same subframe m that it was supposed to receive another transmission. Constraint (2d) indicates that each vehicle transmits at most using a single TB in a given period of the resource pool. Note that the resource assignments to different vehicles are not necessarily unique, i.e., the same resource could be reused by different vehicles. Finally, constraint (2e) limits the transmit power of vehicles.

Solving the above optimization problem is a challenging task given the limited number of resources to be managed for tackling the HD and collision constraints in a highly dynamic environment beyond the network coverage. As a solution, we propose an RL-based scheduling approach called VRLS. VRLS is a logical entity, deployed at the network infrastructure, whose function is to allocate resources to vehicles entering the DOCA with the goal of maximizing the reliability of their V2V transmissions in the DOCA. While it is trained for maximizing the PRR in the DOCA, which can be conducted off-line, in operation, VRLS solely utilizes the information already available at the network.

IV. VRLS: Vehicular Reinforcement Learning Scheduler

We formulate the centralized resource pre-allocation problem for the DOCA as a single-agent RL problem, where VRLS acts as the agent on the vehicular network environment. Based on the observed *state* S_t of the environment at each discrete instant t in which a new vehicle arrives at the DOCA, VRLS takes an *action* A_t, which is to assign a single time-frequency resource to that vehicle. The actions of VRLS are based on its trained *policy* π, which we model as a *deep neural network* (DNN). The agent is trained with a *reward* signal R_{t+1} provided upon each action, indicating how "good" the action was. In turn, the training goal of the agent is to maximize the total reward it receives in the long run. We train VRLS using the state-of-the-art *actor-critic* RL algorithm.

Typically, RL solutions in the literature are designed, trained, and evaluated in the same environment that has a specific distribution of (or even fixed) parameters. However, particular design choices tailored for a specific setting may not work as well or even be applicable when the parameters of the environment change significantly [46]. We have observed the same problem in our earlier work [40], where we had to re-design the components of our solution (thus, also re-train) in order to target different vehicular environments. This turns into a serious limitation, considering the diverse environments that V2V communications need to support. Considering that the RL agent is trained "off-line," i.e., before its deployment, it is desirable to learn a policy that is applicable to different environments of interest without further training. This would eliminate the need of training a new agent from scratch every time an unseen (yet similar) condition arises in the deployed environment. Furthermore, it would offer the possibility to train the agent in a simpler, simulated environment, saving from the burdens of real-world training.

As we elaborate in the following, we design the state, action, reward, and training of VRLS to be applicable to any DOCA environment having an arbitrary size and number of lanes, with any number of vehicles inside, utilizing any resource pool configuration with an arbitrary number of resources in time and frequency. These design features aim at making VRLS a practical solution that is deployable in a variety of realistic OOC environments with different mobility, network load, and wireless conditions, and further facilitate efficient and practical training over simpler and simulated ones.

A. State Representation

We devise the state S_t to provide the agent with information on *how* the resources are utilized at each instant t a vehicle is entering the DOCA. Formally, S_t is a matrix (shown in Fig. 3)
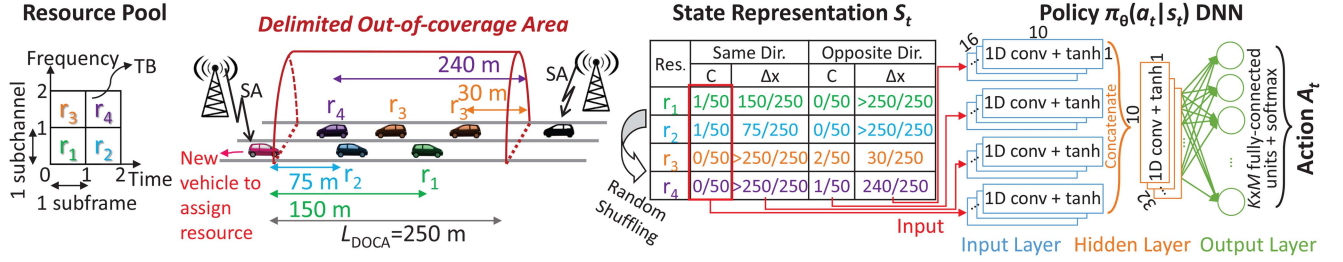
Fig. 3. State representation of a simple exemplary scenario in the DOCA provided to the DNN of VRLS.

with each row representing a resource (TB) in the resource pool, and the columns providing the following information:

- C: number of vehicles each resource is assigned, normalized to the maximum number of vehicles that the DOCA can accommodate (derived by $J \times L_{\text{DOCA}}/L_{\text{veh}}$, considering the case where all lanes are fully occupied). C represents whether the resources are free and, if not, how much loaded. The vehicle density in the DOCA could also be obtained from C by accounting for the sum of the allocated resources in proportion to the calculated maximum number of vehicles.

- Δx: distance from the entrance point of the DOCA to the latest vehicle the resource was assigned to, normalized to L_{DOCA}. The distance is estimated by multiplying the amount of time passed since the vehicle went out of coverage by the average speed v_{avg} of the vehicles in the DOCA, as their speed might vary over time. Δx represents how far the potential interferers are, hence facilitating spatial reuse of each resource.

- The *order* of the columns represents the direction of the vehicle entering the DOCA. The first pair of columns provides C and Δx for the vehicles traveling in the same direction as the vehicle entering the DOCA, while the second pair provides the information from the opposite direction of the DOCA.

Algorithm 1 details how S_t is calculated. The following variables are input for each vehicle i inside the DOCA: time of its entry t_i, assigned resource r_i, traveling direction (i.e., east or west) d_i; as well as the average speed of the vehicle traffic v_{avg} and the current time t_{now}.

S_t is applicable to any number of resources and vehicles, and any DOCA size, thanks to the normalized state variables. An example S_t is illustrated in Fig. 3 for a simple scenario with 4 resources and a DOCA of $L_{\text{DOCA}} = 250$ m, where a maximum of 50 vehicles of $L_{\text{veh}} = 5$ m can fit per lane.

B. Action Definition

The agent takes an action A_t at each instant t a vehicle is about to enter the DOCA. In case multiple vehicles enter at the same instant, the corresponding actions are taken in random order. Action denotes assigning a single time-frequency resource, i.e., a TB, which the vehicle uses for its V2V transmissions through the DOCA. Accordingly, the action-space is a vector of $K \times M$ TBs in the resource pool configured in the network. VRLS gives

---

Algorithm 1: Calculation of the State Representation S_t.

    Require $t_i, r_i, d_i \; \forall i, v_{\text{avg}}, t_{\text{now}}$
1:    Update vehicle distances: $\Delta x_i \leftarrow v_{\text{avg}}(t_{\text{now}} - t_i) \; \forall i$
2:    Remove vehicle if it left the DOCA ($\Delta x_i > L_{\text{DOCA}}$)
3:    **for** each resource in the pool $r = 1, 2, \ldots, K \times M$ **do**
4:        **for** each road direction $d = \{\text{east, west}\}$ **do**
5:           Find vehicles using the same resource in the same direction (check if $d_i == d$ && $r_i == r$)
6:           Update C of the respective d and r with the number of found vehicles
7:           Sort distances of found vehicles to find $\min(\Delta x_i)$
8:           Update Δx of the respective d and r with $\min(\Delta x_i)$
9:        **end for**
10:    **end for**
11:    Order the columns of S_t w.r.t. the direction of the entering vehicle
12:    **return** S_t

---

the decision on which TB to be assigned at time t by its policy π. The policy is a mapping $\pi(a_t|s_t) \rightarrow [0,1]_{K \times M}$ from the state S_t of the environment at t, to a probability distribution over the set of possible actions (the TBs in the resource pool). The TB to be assigned is selected at random according to this distribution.

C. Deep Neural Network Architecture

The large space of possible combinations of vehicles and resources makes tabular RL methods infeasible for this problem [47]. This leads us to apply approximate solution methods by utilizing a DNN to represent the policy. DNN consists of a set of adjustable parameters θ, i.e., $\pi_\theta(a_t|s_t)$ that maps a given state to action probabilities.

We utilize a convolutional neural network (CNN) to model π_θ. At the input layer, we utilize 4 sets of convolutional filters, each processing a different column of S_t, as illustrated in Fig. 3. Each set contains 16 1D convolutional filters of length 10 and applies a $\tanh$ nonlinearity. The output of these filters is then concatenated and input to the hidden layer of the CNN, which is another convolutional layer with 32 1D filters of length 10. The output layer of the CNN is a fully-connected layer with the number of units equal to the number of actions, i.e., $K \times M$ TBs available in the configured resource pool. The softmax activation function

is applied at the output to produce a probability distribution over the actions, from which the TB to be assigned is selected at random.

D. Data Augmentation

The output of convolutional layers is variant to the *order* of the input data they process, due to the convolution operation. Although this is useful for their most common applications, such as processing images or audio that present naturally ordered data (e.g., ordered pixels in space), this feature poses a limitation in our case. The policy of VRLS should not depend on the order of resources presented in S_t, but rather on the information provided about them. That said, the HD constraint depends on the order of resources in the resource pool, as the HD error is caused by using the resources in the same subframe. For example, the two pool configurations $C^{4\times5}$ and $C^{2\times10}$ have different HD constraints, although both have the same number of resources, and there is no information in the state representation to differentiate between them.

To address this challenge, we resort to data augmentation methods. Data augmentation is commonly utilized in deep learning, e.g., for image classification tasks, where the agent is made to *learn* becoming invariant to modifications of the data, e.g., image rotation, clipping, etc., by providing the agent with such modified inputs during the training [48]. In our case, we apply data augmentation by randomly shuffling the order of resources first in time and then in frequency. The rows of S_t and the resource selection probabilities at the output layer of the DNN follow this order. To illustrate our method, consider the example in Fig. 3. A raster-scan ordering of the resources in the resource pool is $[r_1, r_2, r_3, r_4]$. We first group the resources sharing each subframe (corresponding to "columns" of the resource pool), and randomize the order of these groups. This yields a raster-scan ordering of, e.g., $[r_2, r_1, r_4, r_3]$. Then, we group the resources sharing each subchannel, i.e., the "rows" of the pool, and randomly shuffle the order of the "rows". This way, the convolutional network becomes invariant to the order of resources in time or frequency, while being able to infer the HD constraints among the resources.

E. Reward Definition

We incorporate the reliability metric PRR into the reward signal R_{t+1} as a linear function of it: $R_{t+1} = -10 \times (1-\text{PRR})$. PRR is computed at a certain range of interest for all transmissions within the DOCA *since the last action*, i.e., in between each vehicle arrival to the DOCA. The range at which the PRR is measured for the reward could be determined by several factors, such as the distance at which a target PRR value needs to be satisfied; additionally, it can also be limited by the transmission power of the vehicles. In case no transmissions take place between consequent actions, e.g., when two vehicles enter the DOCA almost at the same time, we provide the reward of the previous action to the agent.

F. Training Algorithm

In any RL task, the goal of the agent is to maximize its *return* G_t, i.e., the future rewards it receives in the long run. In its simplest form, $G_t = \sum_{l=1}^{L_{\text{epoch}}} R_{t+l}$, given an *epoch* of experience that consists of L_{epoch} sequential tuples of state-action-reward. The return is estimated via a *value function* $v(s_t)$, indicating how "good" it is for the agent to be in a state and to follow the policy π onwards. Given the large state space of our problem, we also parametrize the value function with a DNN, with a set of parameters w. $v_w(s_t)$ utilizes the same architecture as the policy DNN in Section IV-C, except for its final layer, which is a single fully connected neural unit that outputs the value of the given state.

We employ a state-of-the-art RL algorithm called *policy-gradient actor-critic* [47] to train both DNNs. "Actor" refers to the component that learns the parameterized policy $\pi_\theta(a_t|s_t)$, and "critic" is the component that learns the parameterized value function $v_w(s_t)$ in order to evaluate the actor's policy, i.e., to "criticize" it. The actor updates the policy parameters θ by applying gradient ascent (with step-size α) in the direction the critic indicates as "good". The critic evaluates the actor's policy by measuring $\delta = G_t - v_w(s_t)$, i.e., the error between $v_w(s_t)$ that it estimated and the actual return received from the environment. The parameters w of the critic are also trained in the direction indicated by $\alpha \times \delta$.

To improve training efficiency, we utilize parallel training. Specifically, we employ multiple learning actors, referred to as *workers*, as proposed in [49]. A total of N_{worker} workers are executed in multiple instances of the training environment in parallel. Each instance is simulated with a different random seed and initialized with a random resource assignment to the vehicles, with the first action being also randomized. Each worker asynchronously gathers epochs of experience (of length L_{epoch}) in its own environment. After every epoch, the worker updates the parameters of a single global policy and a value function that are shared by all the workers. In the end, we obtain the global policy that is trained using multiple workers, to be deployed in the target environment. This approach is beneficial in terms of speeding up the training thanks to a more efficient exploration of the state-space of the environment under distinct policies of multiple actors, and the added diversity helps to learn a general policy applicable to similar environments [49].

G. On Real-World Implementation of VRLS

In this study, we train and evaluate the performance of VRLS in simulative environments. Yet, the proposed methods might as well be implemented in a real-world vehicular network. In a real-world scenario, network vendors or operators would implement VRLS as an intelligent controller deployed at the edge of the network and integrated into the radio access network (RAN), thanks to the enabling architecture envisioned for 5G and beyond networks [50]. Within this architecture, the BSs deployed at the entrance/exit of the DOCA can be realized as remote radio units. While these radio units serve physical layer functions, they are connected to a centralized entity that is responsible for

TABLE I
ALGORITHMIC COMPLEXITY OF VRLS

	Time complexity	Space complexity
Stage (i): Calculation of S_t (Steps of Algorithm 1)	Step 1: $\mathcal{O}(V)$ Step 2: $\mathcal{O}(V)$ Step 5: $\mathcal{O}(KMV)$ Step 7: $\mathcal{O}(KMV \log V)$ Overall: $\mathcal{O}(KMV \log V)$	$3V + 2 + 4KM$ variables $(t_i, r_i, d_i \; \forall i = 1, ..., V,\ v_{\text{avg}},\ t_{\text{now}},$ and S_t) Overall: $\mathcal{O}(V + KM)$
Stage (ii): CNN processing	Input layer: $\mathcal{O}(KM)$ Hidden layer: $\mathcal{O}(KM)$ Output layer: $\mathcal{O}(K^2 M^2)$ Overall: $\mathcal{O}(K^2 M^2)$	Input layer: $4 \times 16 \times 10$ variables Hidden layer: 32×10 variables Output layer: $32 \times (4 \times 16(KM{-}10+1){-}10+1) \times KM$ variables Overall: $\mathcal{O}(K^2 M^2)$
Total	$\mathcal{O}(K^2 M^2 + KMV \log V)$	$\mathcal{O}(K^2 M^2 + KM + V)$

V: number of vehicles within the DOCA; K, M: number of subchannels and subframes of the resource pool configured for V2V communications.

resource allocation and other higher-layer functionalities, where VRLS can be implemented. By implementing VRLS, operators would aim at ensuring seamless quality of V2V communications when vehicular users experience coverage losses. This would in turn ensure safer and more efficient road traffic. With regards to deployment and operation costs of VRLS, since being a logical entity, it can be implemented as software and can make use of the processing hardware available at the network infrastructure. Additional processing power is necessary to train the RL agent (with high processing requirements and the possible need to pre-train on the simulated environments), and to operate it (with relatively low processing requirements). VRLS can again benefit from the fact that in 5G RAN, there are more computational resources deployed at the network edge to train learning algorithms [50].

To train and operate VRLS, the BSs delimiting the DOCA would collect and report the required data constituting the state information input to VRLS as described in Section IV-A. The BSs can easily keep track of the time of entry t_i and assigned resource r_i for each vehicle, autonomously, thus not requiring any additional signaling between the vehicles and the BS. Further, the BSs can obtain the information pertaining to d_i and v_{avg} from the regular V2V traffic such as CAMs that vehicles transmit, thus again not requiring any additional signaling. The collected information at the BSs is forwarded to the centralized agent when an action is required. In turn, the actions of VRLS, namely the resource allocation, will be signaled to the vehicles via the BSs, before they enter the DOCA from the respective direction. Considering a pool configuration of $C^{K \times M}$, signaling of an assigned resource would consist of $\log(K + M)$ bits of information. Assuming a vehicle traffic of 0.4 vehicles/s/lane arriving at a DOCA with 3 lanes per direction [3], and a pool of $C^{2 \times 100}$, this would correspond to ~ 8 bits/s of downlink data traffic per BS.

In Table I, we provide the algorithmic complexity of VRLS during its real-time operation (i.e., online inference phase), by decomposing it into two stages: i) the calculation of S_t as described in Algorithm 1; and ii) the processing of S_t by the trained CNN to select the resource as described in Section IV-C. Stage (i) yields a time complexity of $\mathcal{O}(KMV \log V)$, which is due to Step 7 that sorts at most V vehicles in the DOCA, for each resource and direction, i.e., $2KM$ times. At stage (ii), the time complexity of the CNN is dominated by the multiplications at its fully-connected output layer, which yields $\mathcal{O}(K^2 M^2)$.

Altogether, when both stages are combined, VRLS has a time complexity of $\mathcal{O}(K^2 M^2 + KMV \log V)$ during its real-time operation, thereby allowing a practical implementation.

In terms of memory requirements, the algorithm in stage (i) stores $3V + 2 + 4KM$ variables (3 per vehicle, v_{now}, t_{now}, and S_t having $4KM$ entries), which results in a space complexity of $\mathcal{O}(V + KM)$. At stage (ii), CNN stores a total of $4 \times 16 \times 10$, 32×10, and $32 \times (4 \times 16(KM - 10 + 1) - 10 + 1) \times KM$ parameters at its input, hidden, and output layer, respectively, hence yielding a space complexity of $\mathcal{O}(K^2 M^2)$. Overall, the space complexity of VRLS is $\mathcal{O}(K^2 M^2 + KM + V)$, which is practical from the implementation point of view.

For training VRLS, it is possible to collect the reward signal from the network also in a real-world implementation. For example, vehicles could keep track of sent/received V2V message IDs with time and location stamps, which they report to the BSs after going back to the coverage. In turn, the BS calculates the PRR using this information to derive the reward. Such report sent by each vehicle would consist of the IDs and time/location stamps of the messages it has transmitted and received during its past travel within the DOCA. We illustrate the incurred overhead with an example setting as follows. Assuming an average vehicle speed of 50 km/h with an arrival rate of 0.4 vehicles/s/lane, there will be 173 vehicles in a DOCA of 1000 m with 3 lanes/direction at a given time, on average. It would take 72 s on average for a vehicle to travel through the DOCA, where it transmits 720 V2V messages, and receives at most 123 840 messages from other vehicles, assuming a message transmission rate of 10 MHz (i.e., $T_{\text{msg}} = 100$ ms [3]) per vehicle, and all transmissions being successfully received by all vehicles. Further assuming that vehicle IDs are represented with 10 bits of information, and it takes 16 bits to represent the timestamp [41] and 64 bits to represent the location stamp of each message [51], each vehicle would then collect and report 1.40 MB of information to the BS. This would correspond to around 1.69 MB/s of uplink traffic per BS on average. The delay in gathering the information does not pose a limitation for training since the agent acquires experience (the sequence of state-action-reward tuples) in batches before each training step.

On the other hand, training VRLS in simulative environments (and, if needed, re-training during a real-world usage) would circumvent numerous challenges associated with real-world

training from scratch. By simulation, it is easier to create and collect sufficient data; hence the training becomes more flexible and less time-consuming. Besides, the costs of additional signaling and processing overhead at the network and at the vehicles required to collect data would be avoided.

V. EVALUATION METHODOLOGY

A. Key Evaluation Metrics

In addition to the reliability metric PRR defined in Section III-C, we also consider the following performance metrics:

- *Mutual awareness:* We use the mutual awareness metric [52], [53] to study the impact of PRR on the performance of applications running over V2V links. The authors in [53] propose *awareness probability* P_A as an intermediate metric that relates the network quality of service to the application performance. P_A is defined as "probability of successfully receiving at least n packets from a transmitter within the application tolerance time window T," i.e., $P_A = \sum_{n}^{k} \binom{k}{n} p^n (1-p)^{k-n}$, where p is the PRR at the transmitter-receiver range of interest, and k is the number of packets sent during T [53]. Thus, P_A reflects the network performance in the form of PRR, i.e., reliability, and can be used to evaluate its impact on the performance of V2V applications. Each application can set requirements on P_A, as well as on n and T. Requirements of several V2V applications are exemplified in [53], which we provide in Section VI-B.

- *Fairness:* Since the PRR is reported for all vehicles and all V2V transmissions, this does not indicate whether all vehicles experience the same (or similar) reliability. Therefore, the following fairness metric is additionally used in this paper: the average PRR_j is computed separately for each vehicle j, and afterward, the standard deviation of these per-user averages is estimated.

- *Packet inter-reception time (PIR):* PIR is defined as the time elapsed between two successive successful receptions at a certain vehicle, transmitted from another one [3]. It is used to evaluate the "situational awareness" of the vehicles at the V2V application layer [54].

- *Latency:* An important scheduling metric is the latency of V2V messages. The latency is measured between the time a V2V message is generated at the transmitter and successfully received by the receiver, at their application layers, respectively.

B. Training Environment Model and Methodology

The training environment (denoted as "E0") has basic vehicular mobility and wireless channel characteristics, which enables an efficient training thanks to reduced simulation time. The communications in the training environment is abstracted by the *protocol model* [55], i.e., a transmission is assumed to be successful if: i) no other transmitter is using the same TB within a transmission range of $R_{Tx} = 120$ m from the receiver; and ii) the receiver is not transmitting at that time (HD constraint). The impact of path loss and fading in the training environment is

simplified by assuming correct decoding of the received packets within a range of R_{Tx}, and unsuccessful reception beyond that distance. The mobility is simple with 30 vehicles having the same constant speed of 50 km/h, initially placed uniformly at random inside a DOCA of length $L_{DOCA} = 500$ m with $J = 1$ lane/direction of 4 m width. Upon exiting the DOCA, the vehicles are returned back from the opposite direction after a time offset $\sim \text{Exp}(0.4)$, leading to an average inter-vehicle gap of 2.5 s [3]. The V2V resource pool in the network is assumed to be configured with $C^{2 \times 10}$, i.e., 2 subchannels by 10 subframes, to generate loaded conditions with a V2V message traffic that has a fixed periodicity of $T_{msg} = 100$ ms.

The training is conducted using $N_{worker} = 16$ workers in parallel, each interacting with a different instance of the environment in epochs of length $L_{epoch} = 60$. Considering the transmission range of $R_{Tx} = 120$ m and the usage of the protocol model, PRR is measured at a range of 0–100 m away from the transmitters when calculating the reward signal R_{t+1}. Such a range is crucial to avoid imminent crashes between vehicles, where communication reliability needs to be ensured according to V2V service requirements [56].

C. Evaluation Environment Models and Methodology

We evaluate the performance of VRLS trained on the simple environment E0, over realistic environments accommodating various mobility, density, wireless channel conditions, and message traffic in the DOCA. To simulate the vehicular network in the evaluation environments, we use a full-stack LTE V2V protocol in ns-3 [57], based on our own implementation that extends the openly available D2D [58] and LTE [59] modules. The V2V channel model consists of a realistic path loss and shadow fading according to the 3GPP evaluation methodology [3] (cf. Table II for details). Further, we simulate the vehicular mobility in the evaluation environments using the realistic traffic simulator SUMO [60]. Mobility traces from the simulated scenarios we describe below are first generated using SUMO; then, the traces are input to ns-3 to simulate the location of the vehicular nodes. In SUMO, new vehicles are created at different times, which continuously enter and leave the DOCA during the simulated time. In ns-3, however, all nodes need to be created at the beginning of a simulation. Therefore, we turn off the radio functionality of any node outside the DOCA during the network simulations in ns-3.

In our evaluations, we consider two realistic environments denoted as "E1" and "E2". E1 has a single lane per direction, which obliges vehicles to drive in an ordered manner, thus representing a use case similar to platooning. Whereas, E2 has two lanes per direction, which yields more dynamic mobility due to the second lane allowing overtaking. We consider two DOCA lengths of $L_{DOCA} = 500$ m and $L_{DOCA} = 1000$ m for both environments. The vehicle arrivals to the DOCA follow a Poisson distribution with rate 0.4/s (mean of 2.5 s inter-arrival time) per direction as per the 3GPP evaluation assumptions [3]. The vehicles follow a stochastic driving behavior by randomly varying their speeds based on the utilized car-following and lane-changing models [61], [62], which depend on, e.g., average

TABLE II
SIMULATION PARAMETERS

	Training Environment	Evaluation Environments			
	E0	E1-L	E1-HL	E2-L	E2-HL
Mobility model	Straight highway section with 4-m lane-width; vehicle length of $L_{\text{veh}} = 5$ m				
– Highway scenario	$J = 1$ lane/direction; no overtaking $L_{\text{DOCA}} = 500$ m	$J = 1$ lane/direction; no overtaking $L_{\text{DOCA}} = \{500, 1000\}$ m		$J = 2$ lanes/direction with overtaking $L_{\text{DOCA}} = \{500, 1000\}$ m	
– Vehicle speeds	50 km/h (constant)	$\sim \mathcal{N}(120, 12)$ km/h	$\sim \mathcal{N}(50, 5)$ km/h	$\sim \mathcal{N}(120, 36)$ km/h	$\sim \mathcal{N}(50, 15)$ km/h
– Dynamics	30 vehicles with wrap-around $\sim$ Exp(0.4) [3]	Poisson arrival per direction with $\sim$ Exp(0.4) [3] Realistic SUMO mobility [60]			
Network model	TB interference/HD model	Complete LTE V2X protocol stack in ns-3 [57] [58] [59] Bandwidth = 10 MHz (50 RBs) with 32 RBs active; Carrier frequency = 5.9 GHz 1 subchannel = 16 RBs, 1 subframe = 1 ms, MCS index = 9			
V2V channel model	Protocol model [55] with Tx range $R_{\text{Tx}} = 120$ m No path loss and shadowing	3GPP channel model [3] with path loss: LOS model in WINNER+B1 with antenna height = 1.5 m; path loss at 3 m is used for distances < 3 m Shadowing fading: log-normal distr. with 3 dB std. dev. and 25 m decorr. distance $P_{\text{Tx}} = \{-5, 23\}$ dBm; thermal noise level = -174 dBm/Hz 1 Tx and 2 Rx omni-directional antennae with 3 dBi gain and 9 dB Rx noise figure			
V2V message traffic model	$S_{\text{msg}} = 190$ B; $T_{\text{msg}} = 100$ ms for the periodic [3], and $X_{\text{evt}} = 1/s$ for the aperiodic traffic.				
V2V resource pool	$C^{2 \times 10}$ (2 subchannels by 10 subframes) and $C^{2 \times 50}$ (2 subchannels by 50 subframes), periodically repeating with 100 ms				
Mode 4 Configuration Parameters					
$T_1 = 4$ ms [3], $T_2 = \{14, 54\}$ ms, $C_{\text{resel}} \sim$ Unif[5, 15] [21], $P_{\text{keep}} = 0$, $\text{Thr}_{\text{sense}} = -120$ dBm					
VRLS Training Parameters					
$N_{\text{worker}} = 16$; $L_{\text{epoch}} = 60$; $\alpha = 10^{-3}/(1 + 0.01 \times \#ep^{1.1})$					

speed, road length, etc., hence making the mobility even more realistic in the evaluation environments. We vary the mean and variance of the vehicle speeds in both environments to create different loads of the vehicular traffic over time and space. Specifically, we consider two scenarios in terms of vehicle density, denoted as loaded ("L") and highly loaded ("HL"), both in E1 and E2, where the mean speed of the vehicles is set to 120 km/h and 50 km/h, respectively (i.e., the slower, the denser). Further, the speeds among the vehicles are normally distributed, where we set the variance to 10% and 30% of the mean speed values in E1 and E2, respectively. The higher variance of speeds in E2 increases the occurrence of vehicle take-overs across the two lanes.

Unless otherwise stated, the vehicles generate a periodic V2V traffic with $T_{\text{msg}} = 100$ ms and $S_{\text{msg}} = 190$ B (as common to CAMs [3]). We set the MCS index as 9 and the number of RBs per subchannel as 16 to fit the transmission of a single message of 190 B into a single subchannel. In order to simulate loaded (and highly loaded) channel conditions in our evaluations, we assume that the resource pool consists of 2 subchannels in the frequency domain (within an overall V2V bandwidth of 10 MHz) and 10 subframes in the time domain (hence denoted by $C^{2 \times 10}$) unless otherwise stated, considering the number of vehicles and their V2V message generation rate. We accordingly set the length of the resource selection window of the mode 4 algorithm to 10 ms with $T_1 = 4$ [3] and $T_2 = 14$ ms. T_1 is to allow a processing time for the vehicles before they transmit their V2V messages, and T_2 sets a limit on the maximum latency of the transmissions. P_{keep} is set to 0, which leads to a dynamic re-selection of resources as much as possible, and has been shown to improve the reliability by avoiding persistent collisions especially under (highly) loaded channel conditions as in [63]. To enable multiple collision domains within smaller DOCA lengths (500 m), we set the V2V transmit power as -5 dBm, which yields a maximum communication range of around 200 m. This allows us to simulate environments with fewer

vehicles, thus taking shorter simulation times. Nevertheless, we evaluate the performance of the algorithms also with the transmission power set to its allowed maximum value of 23 dBm [64] in Section VI-D. The further parameters related to the environment models, training of VRLS, and configuration of mode 4 are as listed in Table II.

VI. VRLS PERFORMANCE

A. Reliability Performance

In Fig. 4, we compare the reliability of VRLS and mode 4 in E1 and E2 with loaded (L) and highly-loaded (HL) traffic with two DOCA sizes of $L_{\text{DOCA}} = 500$ m and 1000 m, using different subfigures. The plots provide the mean (solid curve) and the standard deviation (shaded region) of the average PRRs calculated in 10 s intervals, for a simulation duration of 1000 s (excluding the initial warm-up phase of 200 s due to the initial random assignment of resources).

Fig. 4 shows that VRLS achieves better performance than mode 4 in all of the considered scenarios. VRLS is typically able to maintain a higher PRR over larger transmission ranges in both E1 and E2. The performance of mode 4 degrades more with the increasing distance between the transmitters and the receivers, mainly caused by the hidden-node problem leading to packet collisions. Beyond 100 m, the path loss effect of the wireless channel becomes dominant, and inevitably reduces the PRR of both algorithms.

To isolate the errors due to scheduling, in Table III, we numerically show the percentage of the packet losses due to the scheduling of the algorithms. The percentages at each transmitter-receiver (Tx-Rx) range are calculated as the difference between the achieved mean PRR and a reference value giving the maximum possible mean PRR in the environment. The reference values represent an ideal scenario, which assumes that there are always sufficient resources for all transmissions, and the packet losses are only due to propagation errors. For
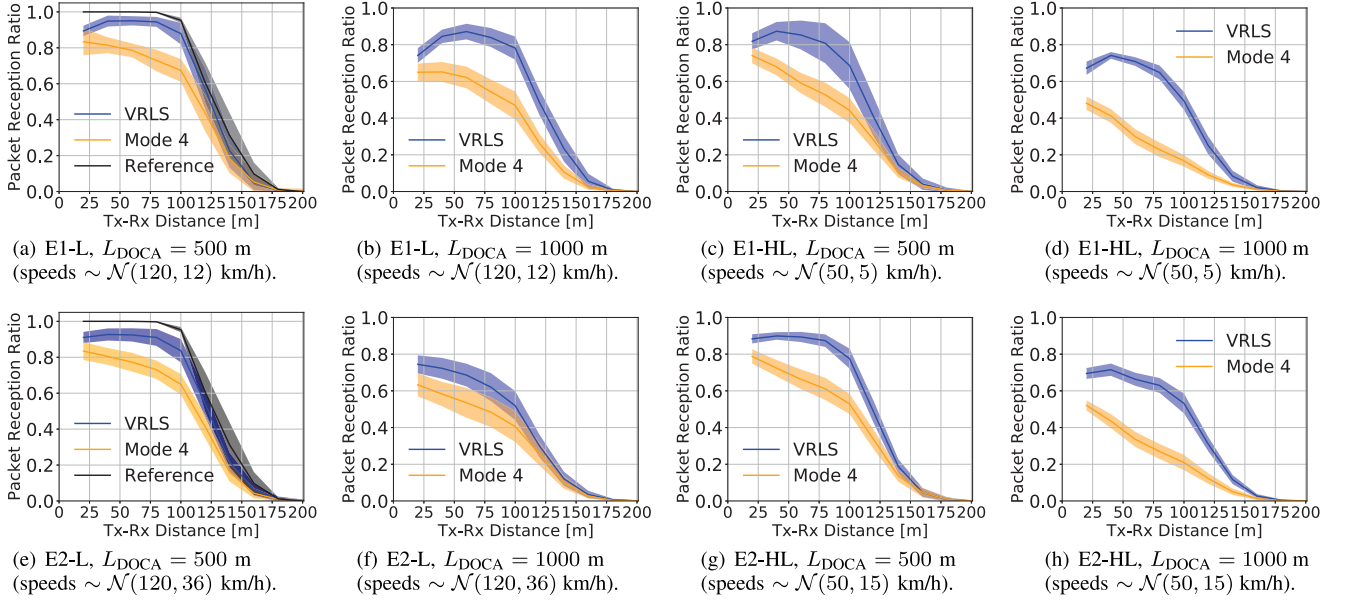
(a) E1-L, $L_{DOCA} = 500$ m (speeds $\sim \mathcal{N}(120, 12)$ km/h).

(b) E1-L, $L_{DOCA} = 1000$ m (speeds $\sim \mathcal{N}(120, 12)$ km/h).

(c) E1-HL, $L_{DOCA} = 500$ m (speeds $\sim \mathcal{N}(50, 5)$ km/h).

(d) E1-HL, $L_{DOCA} = 1000$ m (speeds $\sim \mathcal{N}(50, 5)$ km/h).

(e) E2-L, $L_{DOCA} = 500$ m (speeds $\sim \mathcal{N}(120, 36)$ km/h).

(f) E2-L, $L_{DOCA} = 1000$ m (speeds $\sim \mathcal{N}(120, 36)$ km/h).

(g) E2-HL, $L_{DOCA} = 500$ m (speeds $\sim \mathcal{N}(50, 15)$ km/h).

(h) E2-HL, $L_{DOCA} = 1000$ m (speeds $\sim \mathcal{N}(50, 15)$ km/h).

Fig. 4. Performance of VRLS and distributed scheduling algorithm mode 4 in the DOCA environments E1 and E2 with different vehicular mobility scenarios. PRR vs Tx-Rx distance shown with mean (solid curve) and standard deviation (shade).

TABLE III
PERCENTAGES OF PACKET LOSS DUE TO SCHEDULING, AND MEAN LATENCY IN E1 AND E2

Tx-Rx[m]	E1-L 500m		E1-L 1000m		E1-HL 500m		E1-HL 1000m		E2-L 500m		E2-L 1000m		E2-HL 500m		E2-HL 1000m	
	VRLS	Mode 4	VRLS	Mode 4	VRLS	Mode 4	VRLS	Mode 4	VRLS	Mode 4	VRLS	Mode 4	VRLS	Mode 4	VRLS	Mode 4
0-20	10.6	16.6	25.9	35.0	18.2	26.0	32.8	51.9	9.0	16.6	25.6	36.7	11.7	21.3	30.5	47.9
20-40	5.1	18.6	15.4	34.8	12.6	32.1	25.8	58.8	7.3	19.5	27.7	41.8	10.1	27.6	28.5	56.5
40-60	5.0	21.4	12.8	37.8	14.7	40.9	29.2	70.1	7.6	22.7	31.4	46.5	10.6	33.7	33.7	66.4
60-80	5.3	26.8	15.8	45.4	18.9	47.0	34.9	77.0	8.7	26.7	37.7	51.4	12.4	38.7	36.8	72.8
80-100	7.4	27.9	17.2	48.4	26.7	51.1	46.0	78.6	11.7	30.4	43.4	54.8	17.8	42.5	42.2	74.4
100-120	3.6	17.1	12.2	34.5	20.4	32.8	35.6	51.6	6.5	19.4	30.8	35.2	12.6	27.4	29.7	48.4
120-140	8.5	13.1	7.9	20.6	16.6	21.0	22.9	27.6	9.5	15.3	19.2	21.2	12.5	17.6	19.6	26.3
Latency [ms]	9.43	7.80	9.40	7.89	9.40	8.79	9.41	7.96	9.39	8.25	9.30	8.04	9.44	8.80	9.36	7.92

convenience, we also plot the maximum possible PRR as a *reference curve* in Fig. 4(a), 4(e), and in Fig. 6.

From Table III, we observe that VRLS has superior performance compared to mode 4 in all scenarios. Within a 100 m of Tx-Rx range, VRLS maintains a higher rate of successful packets. Beyond this range, the packet losses are predominantly caused by the propagation loss rather than the scheduling, given the low transmit power. In scenarios E1-L and E2-L with $L_{DOCA} = 500$ m, VRLS shows a performance close to the ideal scenario. The PRR for both algorithms is degraded considerably with the increased vehicular density, as well as the increased DOCA size that impact the interference conditions. In the highly-loaded scenarios, the collisions increase due to the allocation of the same resources to different vehicles. In such cases, although both algorithms perform sub-optimally given the limited number of resources, VRLS results in half the packet losses compared to mode 4.

We examined the policy that VRLS learned by observing the course of states and actions. VRLS develops a strategy to divide the resource pool dynamically into two directions of the highway, in proportion to the data traffic demand. Simultaneously, VRLS performs *resource reuse per direction*, hence mitigating the hidden-node problem. Given the loaded

conditions, HD errors in the network become inevitable even though the collisions could be avoided. Namely, vehicles can be allocated to different subchannels, yet sharing the same subframe. To illustrate, with $C^{2\times10}$, when there are 20 vehicles in the DOCA, each of the 10 subframes would be shared by two vehicles using different subchannels in order to avoid any resource collisions. Yet, such an allocation would result in HD errors among these vehicles when they enter within each other's communication range. In fact, in such loaded scenarios, the probability of unsuccessful transmissions due to HD errors would be even larger if the pool consisted of fewer subframes and more subchannels (e.g., $C^{4\times5}$) as more vehicles would be required to use the different subchannels sharing each subframe to avoid any collision errors. We evaluate and observe such different resource pool configurations in Section VI-F. Yet, VRLS learned to assign the resources with HD conflicts, i.e., the subchannels sharing the same subframe, to vehicles in the *opposite* directions rather than to the nearby vehicles in the same direction. Such assignment strategy results in comparatively fewer HD errors, as those vehicles pass by each other for a shorter duration of time. The outcome is especially observable in E1. The single-lane traffic in E1 results in a minimum inter-vehicle distance of about 40 m within a lane. The Tx-Rx distances below 40 m have a
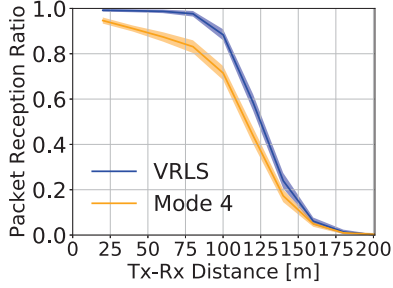
Fig. 5.   PRR evaluation of VRLS and mode 4 with aperiodic V2V traffic in the environment E2-HL with $L_{\text{DOCA}} = 1000$ m. PRR vs Tx-Rx distance shown with mean (solid curve) and standard deviation (shade).
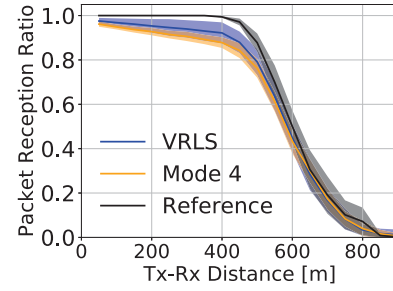

Fig. 6.   PRR vs Tx-Rx distance with mean (solid curve) and standard deviation (shade) for VRLS and mode 4 in comparison to the maximum possible value (reference) in E2-L with $L_{\text{DOCA}} = 1000$ m, $P_{\text{Tx}} = 23$ dBm, and $C^{2\times50}$.

reduced PRR, which occurs only as a result of the vehicles from opposite directions passing by each other. Compared to E1, the second lane in E2 enables the vehicles to overtake each other, hence resulting in a more dynamic environment. Subsequently, the distance between the vehicles in the same direction can take any value, resulting in a smoother decrease of the PRR with the increasing Tx-Rx range, as compared to E1. The trained VRLS policy is efficiently deployable in such a dynamic environment with varying vehicular density and network load over time and space, where it can deliver higher reliability than mode 4.

B. Impact of Network Quality of Service on V2V Applications

We evaluate the impact of PRR on the performance of V2V applications, by utilizing the awareness probability P_{A}. As an illustrative example, the lane-change warning application requires at least $n = 3$ messages to be received within $T = 1$ s with $P_{\text{A}} = 99\%$ to make the neighboring vehicles aware of the intended maneuver [53]. Following our assumption of 10 Hz message frequency, i.e., $k = 10$, and assuming independent message errors, this translates into a PRR requirement (i.e., p given P_{A}) of 61.12%. In our multi-lane environment E2, VRLS can achieve such a PRR at up to a 120 m of range for $L_{\text{DOCA}} = 500$ m (see Fig. 4(e) and 4(g)), and up to around 80 m in $L_{\text{DOCA}} = 1000$ m (Fig. 4(f) and 4(h)). In comparison, mode 4 achieves around 100 m and 80 m of an awareness range in $L_{\text{DOCA}} = 500$ m under the loaded and highly-loaded traffic, respectively. In the case of $L_{\text{DOCA}} = 1000$ m, mode 4 yields an awareness range of 30 m for the loaded scenario, and cannot satisfy the requirement at all for the highly-loaded scenario.

C. Performance Under Aperiodic V2V Traffic

We further evaluate the performance of VRLS under event-triggered V2V traffic. The vehicles are assumed to generate a message upon each event, where the event arrivals for each vehicle follow a Poisson distribution with a rate of 1 event/s ($X_{\text{evt}} = 1/s$). In Fig. 5, we report the PRR performance of the algorithms in scenario E2-HL with $L_{\text{DOCA}} = 1000$ m by considering aperiodic traffic only (not coexisting with periodic traffic). The event-triggered traffic results in less frequent V2V message generation as compared to the periodic traffic, which

effectively creates a lower network load. Accordingly, the performance of both algorithms is increased (observed also in other scenarios), with VRLS achieving a PRR very close to 100% up to a range of 80 m. The results show that the policy learned by VRLS for the periodic traffic is applicable to the aperiodic type of traffic as well. On the other hand, mode 4, which is a solution primarily designed for periodic V2V traffic, underperforms in this setting.

D. Performance Under High Transmit Power and Larger Pool

For all of the results above, the vehicle transmit powers are set to $P_{\text{Tx}} = -5$ dBm, whereas the allowed maximum for V2V transmissions is 23 dBm [64]. To ensure that the performance of VRLS holds for larger and arguably more realistic communication ranges, we evaluate the performance of the algorithms with the transmission power set to 23 dBm for all vehicles in E2-L with $L_{\text{DOCA}} = 1000$ m. In order to compensate for the increased interference caused by the high-powered transmissions, we consider a pool that consists of 2 subchannels and 50 subframes, i.e., $C^{2\times50}$, which is five times larger than the resource pool configuration $C^{2\times10}$ we have considered so far. For this scenario, VRLS is trained in E0 as well, but utilizing the resource pool $C^{2\times50}$.

Results of the algorithms are provided in Fig. 6, where we see that both algorithms achieve very high reliability, close to a 100% PRR at shorter Tx-Rx distances, owing to the sufficiently provisioned resources. VRLS delivers a marginally higher PRR than mode 4 at almost all Tx-Rx ranges. The results demonstrate that VRLS is trainable in environments having different resource pool configurations, and that the learned policy is applicable to scenarios with different transmission ranges. The small percentage of the packet losses results mainly from propagation errors, but also due to HD or even collisions to a small extent. In the case of mode 4, vehicles cannot sense the subframes they transmit on (due to the HD constraint); thus there exists a probability of selecting the resources used by other vehicles that might interfere or collide. In the case of VRLS, although the learned policy avoids allocating the same resource to more than a single vehicle, it is challenging for the agent to learn the HD constraints in such a large and sparse state-space, where some of its assignments lead to HD errors.
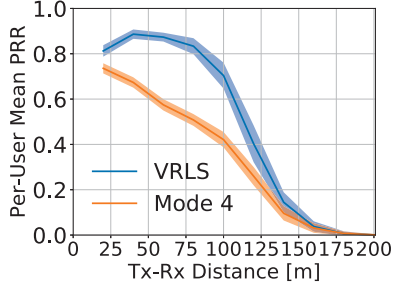
Fig. 7.   Per-user mean PRR of VRLS and mode 4 in E1-HL with $L_{\mathrm{DOCA}} = 500$ m; shown with mean (solid curve) and standard deviation (shade).
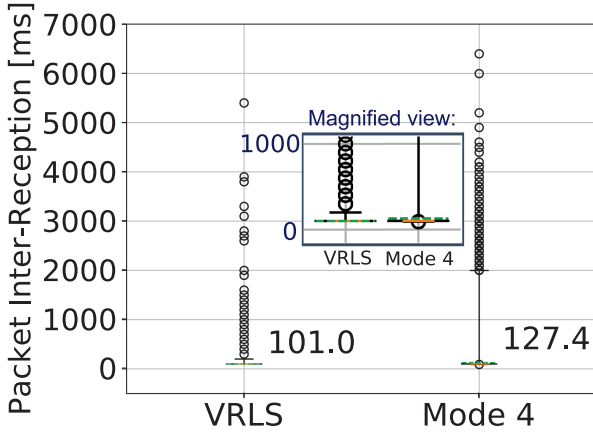

Fig. 8.   PIR performance of the algorithms in E2-HL with $L_{\mathrm{DOCA}} = 500$ m, shown with mean (green, dashed, denoted): 101.0^V, 127.4^M; median (orange): 100.0^V, 100.0^M; 25th and 75th percentiles (black): both 100.0^V, 100.0^M; 0.1st and 99.9th percentiles (whiskers): 100.0^V, 100.0^M and 200.0^V, 2000.0^M; and outliers (rings), measured at Tx-Rx distances of 0–50 m (V:VRLS, M:Mode 4, values in ms).

E. User Fairness, Packet Inter Reception, and Latency

Although our solution gives an equal opportunity to all vehicles to transmit in the DOCA by allocating resources, the PRR results do not provide the information on whether fairness is ensured, i.e., the PRR of a certain group of users is not sacrificed in favor of system-wide performance. In Fig. 7, we provide the mean and the standard deviation of the per-user average PRRs, to evaluate the variation of reliability across the users. The results are presented for the highly-loaded environment E1-HL with $L_{\mathrm{DOCA}} = 500$ m. We observe that VRLS is able to deliver its performance without sacrificing user fairness. Both VRLS and mode 4 achieve a similar variation of mean PRR across the users, where the standard deviation is around 0.025 considering all Tx-Rx ranges.

In Fig. 8, we report another *per-user* performance metric, PIR, in the environment E2-HL with $L_{\mathrm{DOCA}} = 500$ m, measured within a 50 m of Tx-Rx range, in terms of mean and percentiles. We observe that VRLS achieves better performance than mode 4, the latter resulting in at least 25% larger PIR on average. Note that for both algorithms, all quartiles of the PIR are at 100 ms, which is equal to T_{msg}. We have observed that the relative PIR performance of the algorithms in the other scenarios are also similar, where VRLS achieves a mean PIR close to 100 ms, at

most reaching 106 ms, and mode 4 resulting in mean values ranging from around 115 ms up to 200 ms that mainly increase with the network load. For the high-transmit-power scenario in Fig. 6, as the best case, VRLS and mode 4 can achieve a mean PIR of 100.4 and 100.6 ms, respectively. Combined with our analysis in Section VI-B, it is evident that VRLS is able to provide superior awareness to the vehicular users while maintaining fairness, which benefits the V2V applications.

We report the mean latency measured in all evaluation scenarios in the last row of Table III. Note that the messages are at least delayed by the processing time across the communication layers, which is assumed to be 4 ms [3], and at most delayed by the time-length of the resource pool, which is 10 ms, plus the processing time, as all messages are scheduled within the utilized resource pool by both algorithms. Both VRLS and mode 4 yield a similar latency of around 9 ms, on average.

F. Handling Different Resource Pool Configurations

The network might need to operate different resource pool configurations that may vary in terms of the number of resources in time and frequency, i.e., K and M, so as to serve different V2V services with different traffic requirements, or depending on the availability of resources. Therefore, we are interested in studying whether and how VRLS can handle a *set* of different resource pools configured with different number of resources in time and frequency, with a *single* training. Note that the pool configurations are usually a part of network planning; thus the configuration information would be available before operating the scheduler. Given this information, having a single policy that can achieve an appropriate performance level under all different configurations would be desirable, instead of training multiple ones that can only operate on a specific configuration. That said, it is a challenge to learn and solve the HD and collision constraints of different pools at once by a single policy, as each has a different impact on the performance. Such an approach requires careful consideration of the state and the other RL components.

We describe our method to train a single policy for multiple resource pool configurations as follows. Consider a set of different resource pool configurations $\{C_1^{K_1 \times M_1}, C_2^{K_2 \times M_2}, \ldots\}$ to be operated by the network, where each pool C_i consists of a different number of subchannels K_i and subframes M_i. We first determine a superset ("master") pool configuration $C_{\mathrm{ms}}^{K_{\mathrm{ms}} \times M_{\mathrm{ms}}}$, which can accommodate any configuration in the set. Accordingly, the dimensions of C_{ms} are selected as $K_{\mathrm{ms}} = \max(K_1, K_2, \ldots)$ and $M_{\mathrm{ms}} = \max(M_1, M_2, \ldots)$. VRLS is provided with a state- and an action-space having the same number of resources as in C_{ms}, i.e., $K_{\mathrm{ms}} \times M_{\mathrm{ms}}$. As illustrated in Fig. 9, considering a case where the network operates four different resource configurations $\{C_1^{2 \times 10}, C_2^{4 \times 5}, C_3^{5 \times 4}, C_4^{10 \times 2}\}$, a master pool of $C_{\mathrm{ms}}^{10 \times 10}$ accommodates all four; thus the state- and the action-space of VRLS consist of 100 resources. We provide the different pool configurations to different groups of workers training the VRLS policy in parallel. Each group of workers i is trained with the pool configuration C_i, where we only "disclose" the resources of C_i within S_t by replacing the rows corresponding to other resources
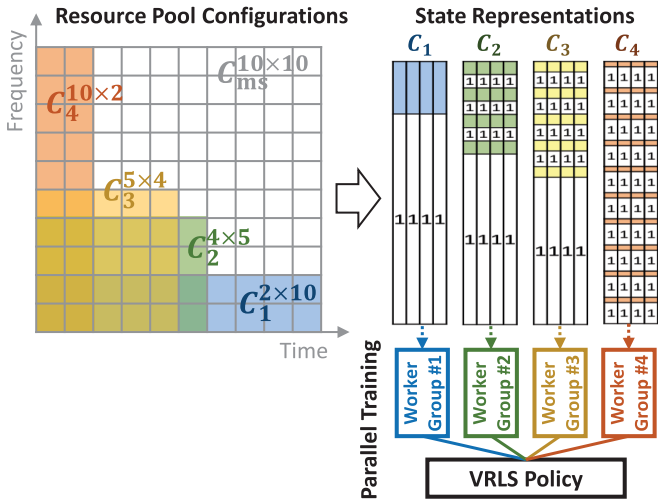
Fig. 9.   Training VRLS with multiple resource pool configurations in parallel.
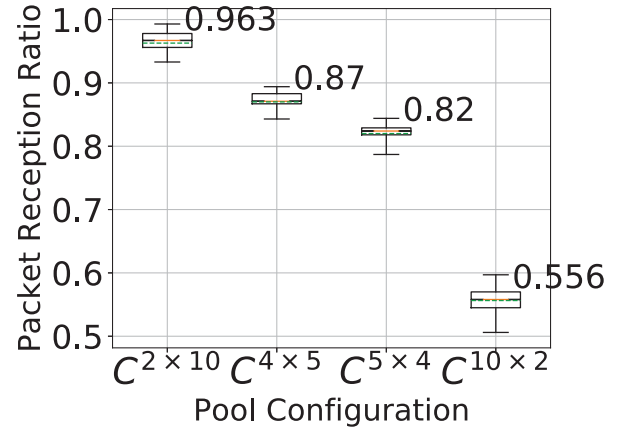

Fig. 10.   PRR performance of VRLS on different resource pool configurations $C^{K \times M}$ having K subchannels and M subframes, with mean (green, dashed, denoted), median (orange), 25th and 75th percentiles (box), and 5th and 95th percentiles (whiskers).

with the row vector $[1, 1, 1, 1]$. Further, if the worker selects such a resource, we provide a large negative reward and execute no actions until the worker selects a resource within its own pool. With such training, we aim at limiting the action selection of the policy only to the represented subset of the resources in S_t.

In the following, we evaluate our solution for four different configurations $\{C_1^{2 \times 10}, C_2^{4 \times 5}, C_3^{5 \times 4}, C_4^{10 \times 2}\}$. VRLS is trained from scratch with a total of 40 workers in parallel, in four groups of 10 workers. Each group is provided with one of the four different configurations. If a worker selects a resource outside its configuration, a reward of $R_{t+1} = -10$ is provided. In turn, to compensate for the higher variance in the rewards, the training epoch length is increased to 200 actions. We evaluate VRLS in a DOCA similar to E0, with 10 vehicles initially placed on the highway with transmission range $R_{\text{Tx}} = 500$ m, resulting in a single collision domain (transmissions using the same TB are assumed to collide). Such a simple setting enables a deterministic calculation of performance bounds and better evaluation of whether the learned policy can deal with the constraints of the different resource pools. Namely, in the case of $C_1^{2 \times 10}$, when all 10 vehicles reside in the DOCA, a 100% PRR would be achievable only if vehicles were assigned to TBs in different subframes. With $C_2^{4 \times 5}$, the allocation for all 10 vehicles would be optimal if all TBs were assigned orthogonally first in time, then in frequency. Every transmission would be received by all other vehicles except the one transmitting in the same subframe, due to the HD constraint. Thus, the best assignment of TBs would result in eight successful receptions out of nine receiving vehicles, yielding an $88.\bar{8}\%$ PRR. Similarly, in the case of $C_3^{5 \times 4}$ and $C_4^{10 \times 2}$, when all of 10 vehicles exist in the DOCA at the same time, orthogonal assignment of TBs first in time and then in frequency would yield $82.\bar{2}\%$ and $55.\bar{5}\%$ PRR, respectively.

In Fig. 10, we report the performance of VRLS when applied to the network with different resource pool configurations, in terms of the PRR measured up to a 500 m of Tx-Rx range. We observe that VRLS yields a mean PRR almost equal to the calculated bounds of $1.0, 0.\bar{8}, 0.8\bar{2}$, and $0.\bar{5}$ for the configurations $C_1^{2 \times 10}, C_2^{4 \times 5}, C_3^{5 \times 4}$, and $C_4^{10 \times 2}$, respectively. Note that the larger
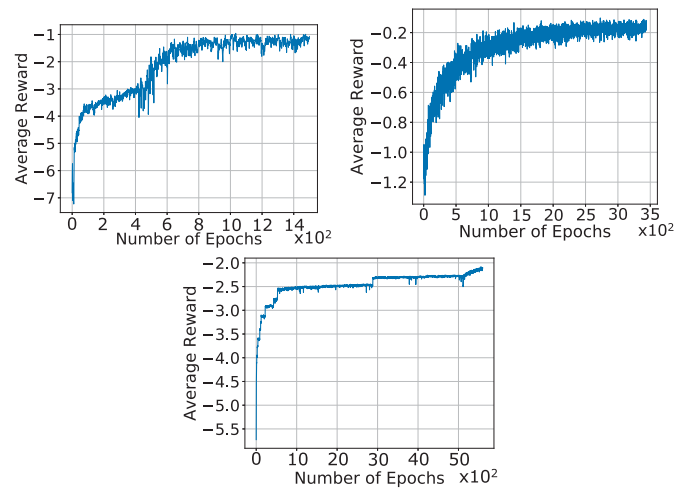

Fig. 11.   Learning curves of VRLS in the training environment E0 with resource pool configuration $C^{2 \times 10}$ (top left), $C^{2 \times 50}$ (top right), and with multiple configurations $\{C_1^{2 \times 10}, C_2^{4 \times 5}, C_3^{5 \times 4}, C_4^{10 \times 2}\}$ (bottom).

PRRs are reached when fewer than the maximum of 10 vehicles reside in the DOCA. VRLS is able to achieve such performance by learning a single policy that can handle distinct constraints of HD and collisions for different pools simultaneously.

VII. LEARNING PERFORMANCE

We provide the learning curves of VRLS in the training environment E0 with different resource pool configurations that we considered throughout our evaluations, in Fig. 11. The curves represent the average reward collected by the trained workers versus the number of training epochs. VRLS converged to a stable performance level after around 1000 epochs when trained with a single resource configuration $C^{2 \times 10}$ as per Section VI-A. With the larger resource pool of $C^{2 \times 50}$ as per Section VI-D, it took around two times longer for the agent to converge. This is because of additional exploration required by the increased state and the action space. VRLS obtained a larger average reward in the case of $C^{2 \times 50}$ owing to the sufficiently

provisioned resources in the network. When VRLS is trained with four resource pool configurations in parallel, i.e., with $\{C_1^{2\times 10},\ C_2^{4\times 5},\ C_3^{5\times 4},\ C_4^{10\times 2}\}$ as per Section VI-F, it took a longer time for the algorithm to converge as compared to the training with a single pool configuration. This is due to the different collision and HD constraints posed by each different resource pool configuration that VRLS needs to learn, as well as the larger state space, which results in slower convergence. The collected reward is smaller as it represents the average of dissimilar performance levels on the different pools reported in Fig. 10. The overall performance is largely converged, which could be yet further optimized, such as via exhaustive training on the desired configuration, or with a larger number of workers, however, calling for increased training time and resources.

VIII. CONCLUSIONS AND OUTLOOK

We designed VRLS, a centralized scheduling approach for V2V communications outside the network coverage, based on RL. VRLS is applicable to a variety of vehicular environments having different sizes, densities, mobility, network load, wireless conditions, and resource configurations. Because VRLS can be pre-trained using simulations, it can be trained on a wider range of environments and resource configurations than what would be practically doable in the real world. In terms of V2V communication reliability, VRLS outperforms the state-of-the-art mode 4 scheduler by reducing the packet loss by half in case of overloaded network conditions, and performing very close to the maximum possible level under low load. Furthermore, while achieving similar fairness and V2V communication latency as mode 4, VRLS provides higher awareness among the vehicles.

Nevertheless, the actions of VRLS, namely assigning a single resource to each vehicle going outside the coverage, might become unfeasible under certain conditions. There could occur some situations such as road congestion due to an accident, or cases for which the agent might not be trained. Our complementary work [65] addresses such issues by encompassing a *hybrid* solution that combines the centralized RL-based approach with the distributed sensing-based scheduling for areas outside the network coverage. In any case, if the performance degradation is not tolerable, i.e., the communications cannot satisfy the requirements of a given V2V use case, the network could over-provision the resources, as we have shown, or congestion control mechanisms could be applied, considering such unforeseeable conditions.

In another line of our work [66], we proposed *iVRLS* (in-coverage VRLS) for scheduling *in-coverage* V2V communications, which extends the VRLS design by taking advantage of resource assignments that are possible at all times and making use of instantaneous and exact knowledge of vehicular mobility. iVRLS is shown to improve the V2V transmission reliability under high traffic load, with less frequent scheduling as compared to a state-of-the-art scheduling algorithm. Further work remains to evaluate the network efficiency of the proposed algorithms, by measuring the number of resources they demand to satisfy target reliability requirements under varying conditions of network coverage.

REFERENCES

[1] 3GPP, "Service requirements for V2X services; Stage 1," 3GPP Tech. Specification 22.185, Jun. 2018.

[2] 3GPP, "Enhancement of 3GPP support for V2X scenarios; Stage 1," 3GPP Tech. Specification 22.186, Dec. 2018.

[3] 3GPP, "Study on LTE-based V2X services," 3GPP Tech. Rep. 36.885, Jun. 2016.

[4] 3GPP, "Study on NR vehicle-to-everything (V2X)," 3GPP Tech. Rep. 38.885, Mar. 2019.

[5] 3GPP, "Study on evaluation methodology of new vehicle-to-everything (V2X) use cases for LTE and NR," 3GPP Tech. Rep. 37.885, Dec. 2018.

[6] 3GPP, "Overall description of radio access network (RAN) aspects for vehicle-to-everything (V2X) based on LTE and NR," 3GPP Tech. Rep. 37.985, Jun. 2019.

[7] *IEEE Standard for Information Technology - Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, IEEE Standard 802.11, 2010.

[8] G. Naik, B. Choudhury, and J.-M. Park, "IEEE 802.11bd & 5G NR V2X: Evolution of radio access technologies for V2X communications," *IEEE Access*, vol. 7, pp. 70169–70184, 2019.

[9] V. Vukadinovic et al., "3GPP C-V2X and IEEE 802.11 p for vehicle-to-vehicle communications in highway platooning scenarios," *Ad Hoc Netw.*, vol. 74, pp. 17–29, 2018.

[10] G. Cecchini, A. Bazzi, B. M. Masini, and A. Zanella, "Performance comparison between IEEE 802.11p and LTE-V2V in-coverage and out-of-coverage for cooperative awareness," in *Proc. IEEE Veh. Netw. Conf.*, 2017, pp. 109–114.

[11] R. Molina-Masegosa, J. Gozalvez, and M. Sepulcre, "Comparison of IEEE 802.11p and LTE-V2X: An evaluation with periodic and aperiodic messages of constant and variable size," *IEEE Access*, vol. 8, pp. 121526–121548, 2020.

[12] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. 15th ACM Workshop Hot Topics Netw.*, 2016, pp. 50–56.

[13] H. Ye, L. Liang, G. Y. Li, J. Kim, L. Lu, and M. Wu, "Machine learning for vehicular networks: Recent advances and application examples," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 94–101, Jun. 2018.

[14] T. Sahin, R. Khalili, M. Boban, and A. Wolisz, "VRLS: A unified reinforcement learning scheduler for vehicle-to-vehicle communications," in *Proc. IEEE 2nd Connected Automated Veh. Symp.*, 2019, pp. 1–7.

[15] G. Nardini, A. Virdis, C. Campolo, A. Molinaro, and G. Stea, "Cellular-V2X communications for platooning: Design and evaluation," *Sensors*, vol. 18, no. 5, 2018, Art. no. 1527.

[16] A. Bazzi, C. Campolo, A. Molinaro, A. O. Berthet, B. M. Masini, and A. Zanella, "On wireless blind spots in the C-V2X sidelink," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9239–9243, Aug. 2020.

[17] Y. Yoon and H. Kim, "Resolving persistent packet collisions through broadcast feedback in cellular V2X communication," *Future Internet*, vol. 13, no. 8, 2021, Art. no. 211.

[18] G. Giordano, M. Segata, F. Blanchini, and R. L. Cigno, "The joint network/control design of platooning algorithms can enforce guaranteed safety constraints," *Ad Hoc Netw.*, vol. 94, 2019, Art. no. 101962.

[19] 3GPP, "Evolved universal terrestrial radio access (E-UTRA) and evolved universal terrestrial radio access network (E-UTRAN); Overall description; Stage 2," 3GPP Tech. Specification 36.300, Dec. 2019.

[20] 3GPP, "Evolved universal terrestrial radio access (E-UTRA); Physical layer procedures," 3GPP Technical Specification 36.213, Dec. 2019.

[21] 3GPP, "Evolved universal terrestrial radio access (E-UTRA); Medium access control (MAC) protocol specification," 3GPP Tech. Specification 36.321, Dec. 2019.

[22] S.-Y. Jung, H.-R. Cheon, and J.-H. Kim, "Reducing consecutive collisions in sensing based semi persistent scheduling for cellular-V2X," in *Proc. IEEE 90th Veh. Technol. Conf.*, 2019, pp. 1–5.

[23] P. Wendland, G. Schaefer, and R. S. Thomä, "LTE-V2X mode 4: Increasing robustness and DCC compatibility with reservation splitting," in *Proc. IEEE Int. Conf. Connected Veh. Expo*, 2019, pp. 1–6.

[24] S. Heo, W. Yoo, H. Jang, and J.-M. Chung, "H-V2X mode 4 adaptive semipersistent scheduling control for cooperative internet of vehicles," *IEEE Internet of Things J.*, vol. 8, no. 13, pp. 10678–10692, Jul. 2021.

[25] R. Molina-Masegosa, M. Sepulcre, and J. Gozalvez, "Geo-based scheduling for C-V2X networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8397–8407, Sep. 2019.

[26] G. Cecchini, A. Bazzi, B. M. Masini, and A. Zanella, "MAP-RP: Map-based resource reselection procedure for autonomous LTE-V2V," in *Proc. IEEE 28th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2017, pp. 1–6.

[27] S. Sabeeh, P. Sroka, and K. Wesołowski, "Estimation and reservation for autonomous resource selection in C-V2X mode 4," in *Proc. IEEE 30th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2019, pp. 1–6.

[28] Y. Jeon and H. Kim, "An explicit reservation-augmented resource allocation scheme for C-V2X sidelink mode 4," *IEEE Access*, vol. 8, pp. 147241–147255, 2020.

[29] L. F. Abanto-Leon, A. Koppelaar, and S. H. de Groot, "Enhanced C-V2X mode-4 subchannel selection," in *Proc. IEEE 88th Veh. Technol. Conf.*, 2018, pp. 1–5.

[30] S. Sharma and B. Singh, "Context aware autonomous resource selection and Q-learning based power control strategy for enhanced cooperative awareness in LTE-V2V communication," *Wireless Netw.*, vol. 26, no. 6, pp. 4045–4060, 2020.

[31] C. Wu, T. Yoshinaga, Y. Ji, T. Murase, and Y. Zhang, "A reinforcement learning-based data storage scheme for vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6336–6348, Jul. 2017.

[32] J. Wang, C. Jiang, K. Zhang, T. Q. Quek, Y. Ren, and L. Hanzo, "Vehicular sensing networks in a smart city: Principles, technologies and applications," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 122–132, Feb. 2018.

[33] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, Feb. 2020.

[34] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[35] R. Hu, X. Wang, Y. Su, and B. Yang, "An efficient deep reinforcement learning based distributed channel multiplexing framework for V2X communication networks," in *Proc. IEEE Int. Conf. Consum. Electron. Comput. Eng.*, 2021, pp. 154–160.

[36] H. Yang, X. Xie, and M. Kadoch, "Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency IoV communication networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4157–4169, May 2019.

[37] Y. Xia, L. Wu, Z. Wang, X. Zheng, and J. Jin, "Cluster-enabled cooperative scheduling based on reinforcement learning for high-mobility vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12664–12678, Nov. 2020.

[38] A. Pressas, Z. Sheng, F. Ali, D. Tian, and M. Nekovee, "Contention-based learning MAC protocol for broadcast vehicle-to-vehicle communication," in *Proc. IEEE Veh. Netw. Conf.*, 2017, pp. 263–270.

[39] Y. Su, M. LiWang, Z. Gao, L. Huang, S. Liu, and X. Du, "Coexistence of cellular V2X and Wi-Fi over unlicensed spectrum with reinforcement learning," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.

[40] T. Şahin, R. Khalili, M. Boban, and A. Wolisz, "Reinforcement learning scheduler for vehicle-to-vehicle communications outside coverage," in *Proc. IEEE Veh. Netw. Conf.*, 2018, pp. 1–8.

[41] *Intelligent Transport Systems (ITS), Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, ETSI Standard EN 302 637–2, Rev. 1.3.1, 2014.

[42] *Intelligent Transport Systems (ITS), Vehicular Communications; Basic Set of Applications; Part 3: Specification of Decentralized Environmental Notification Basic Service*, ETSI Standard EN 302 637–3, Rev. 1.2.0, 2013.

[43] A. Bazzi, G. Cecchini, A. Zanella, and B. M. Masini, "Study of the impact of PHY and MAC parameters in 3GPP C-V2V mode 4," *IEEE Access*, vol. 6, pp. 71685–71698, 2018.

[44] A. Mansouri, V. Martinez, and J. Härri, "A first investigation of congestion control for LTE-V2X mode 4," in *Proc. 15th Annu. Conf. Wireless On-Demand Netw. Syst. Serv.*, 2019, pp. 56–63.

[45] Z. Ding, I. Krikidis, B. Rong, J. S. Thompson, C. Wang, and S. Yang, "On combating the half-duplex constraint in modern cooperative networks: Protocols and techniques," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 20–27, Dec. 2012.

[46] S. Gamrian and Y. Goldberg, "Transfer learning for related reinforcement learning tasks via image-to-image translation," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 2063–2072.

[47] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[49] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[50] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, programmable, and virtualized 5G networks: State-of-the-art and the road ahead," *Comput. Netw.*, vol. 182, 2020, Art. no. 107516.

[51] *Intelligent Transport Systems; Users and Applications Requirements; Part 2: Applications and Facilities Layer Common Data Dictionary*, ETSI Standard EN 102 894–2, Rev. 1.2.1, 2014.

[52] M. Boban and P. M. d'Orey, "Exploring the practical limits of cooperative awareness in vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3904–3916, Jun. 2016.

[53] N. An, T. Gaugel, and H. Hartenstein, "VANET: Is 95% probability of packet reception safe?," in *Proc. Int. Conf. ITS Telecommun.*, 2011, pp. 113–119.

[54] M. E. Renda, G. Resta, P. Santi, F. Martelli, and A. Franchini, "IEEE 802.11 p VANets: Experimental evaluation of packet inter-reception time," *Comput. Commun.*, vol. 75, pp. 26–38, 2016.

[55] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[56] 3GPP, "Study on LTE support for vehicle-to-everything (V2X) services," 3GPP Tech. Rep. 22.885, Sep. 2015.

[57] The ns-3 network simulator. 2017, [Online]. Available: www.nsnam.org

[58] R. Rouil, F. J. Cintrán, A. B. Mosbah, and S. Gamboa, "Implementation and validation of an LTE D2D model for ns-3," in *Proc. Workshop ns-3*, 2017, pp. 55–62.

[59] G. Piro, N. Baldo, and M. Miozzo, "An LTE module for the ns-3 network simulator," in *Proc. 4th Int. Conf. Simul. Tools Techn.*, 2011, pp. 415–422.

[60] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2575–2582.

[61] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," DLR, Cologne, Germany, Tech. Rep. DLR-FB-98-08, 1998.

[62] J. Erdmann, "Lane-changing model in SUMO," in *Proc. SUMO2014 Model. Mobility Open Data*, 2014, pp. 77–88.

[63] P. Wendland, G. Schaefer, and R. Thomä, "An application-oriented evaluation of LTE-V's mode 4 for V2V communication," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, 2019, pp. 165–173.

[64] 3GPP, "User equipment (UE) radio transmission and reception," 3GPP Technical Specification 36.101, Dec. 2019.

[65] T. Şahin, M. Boban, R. Khalili, and A. Wolisz, "A hybrid sensing and reinforcement learning scheduler for vehicle-to-vehicle communications," in *Proc. 53rd Asilomar Conf. Signals Syst. Comput.*, 2019, pp. 1136–1143.

[66] T. Şahin, M. Boban, R. Khalili, and A. Wolisz, "iVRLS: In-coverage vehicular reinforcement learning scheduler," in *Proc. IEEE 93rd Veh. Technol. Conf.*, 2021, pp. 1–7.

Taylan Şahin received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 2014, and the M.Sc. degree from the Technical University of Munich, Munich, Germany, in 2016. He is currently working toward the Doctoral degree with the Technical University of Berlin, Berlin, Germany. He was with Huawei Munich Research Center, Germany as a Research Student from 2015 to 2020. Since 2020, he has been with Nokia Standards, Munich. His research interests include resource allocation, positioning, and machine learning in wireless mobile networks. His master's work was the recipient of the Best Paper Award, and he was the Co-Chair and TPC Member in various IEEE conferences and workshops.

Ramin Khalili received the B.Sc. degree from Shiraz University, Shiraz, Iran, the M.Sc. degree from the Sharif University of Technology, Tehran, Iran, and the Ph.D. degree from Pierre and Marie Curie University, Paris, France. He was with the University of Massachusetts Amherst, Amherst, MA, USA, The Ecole polytechnique federale de Lausanne, Lausanne, Switzerland, and the TLabs in Berlin, before joining the Huawei Research Center in Munich, Germany. He has authored or coauthored more than fifty scientific papers on topics related to wireless networking, optimization, and machine learning. He was the recipient of multiple best paper awards during these years.
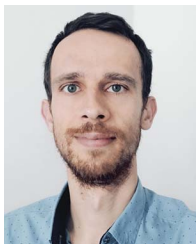
Mate Boban (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA. He is a Principal Research Engineer with Huawei Technologies, Munich Research Center, Germany. He is currently a Co-Chair of COST INTER-ACT WG1 (Radio Channels) and an Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING. His research interests include channel modeling, resource allocation, and machine learning applied to wireless communications.

Adam Wolisz (Senior Member, IEEE) received the M.S., Ph.D., and Habilitation degrees from the Silesian University of Technology, Gliwice, Poland, in 1972, 1976, and 1983, respectively. After research work with the Institute of Complex Control Systems, Polish Academy of Sciences and GMD Fokus (Berlin) he was appointed Professor with Technische Universitaet Berlin, Berlin, Germany, leading the Telecommunication Networks Group (1993–2018) and acted as the Executive Director of the Institute for Telecommunication Systems (2001–2018). He was also an Adjunct Professor with EECS, University of California, Berkeley, Berkeley, CA, USA, (2005–2017) associated with Berkeley Wireless Research Center. Since retirement (2018) he has been active as a fellow with Einstein Center Digital Future, Berlin.