`Edge-V`: Vehicular Edge Intelligence Through Multi-Band Unlicensed Spectrum Access

Francesco Raviglione, Claudio Casetti, *Senior Member, IEEE,*
and Francesco Restuccia, *Senior Member, IEEE*

***Abstract*—Technological advances in the automotive field are driving the development of smarter, greener, and more autonomous vehicles. These vehicles will need to communicate via Vehicle-to-Everything (V2X) wireless communications and perform advanced Deep Learning (DL) tasks while handling large data volumes with low latency and high reliability. Although 5G is frequently viewed as a comprehensive solution for addressing the demanding environment of next-generation autonomous vehicles and of Vehicular Edge Intelligence (VEI), relying solely on cellular networks poses challenges like spectrum congestion, delays in edge offloading, and poor coverage in certain areas. Current unlicensed spectrum technologies also fall short of the VEI requirements. On this basis, we propose `Edge-V`, a novel framework combining unlicensed spectrum technologies to provide low-latency, high-throughput connectivity with reliable task offloading. `Edge-V` uses a Dedicated Short-Range Communications (DSRC) link for exchanging standardized messages, traditional Wi-Fi for connecting on-board devices and sensors, and mmWave for high-speed, low-latency connectivity. With the aim of optimally allocating tasks, an Offloading Manager module is included, based on a system model which is mathematically formulated, and used to propose a sample greedy strategy within `Edge-V`. Our laboratory and field tests, thanks to an open and low-cost Proof-of-Concept, show that `Edge-V` can reduce latency by up to 65% when compared to cellular/cloud-based solutions.**

***Index Terms*—Vehicular Edge Computing, Vehicular Edge Intelligence, VEI, mmWave, Vehicular Networks, unlicensed spectrum**

I. Introduction

An essential step in making autonomous vehicles smarter is making them more aware of their surroundings. In addition to the exchange of standard-compliant messages for safety and non-safety applications, future vehicles are expected to share their on-board sensor data – coming from LiDARs, radars and especially cameras – to enable technologies such as advanced *See-Through* with high-quality video feedback, or real-time sharing of high-definition maps among self-driven vehicles for accurate localization [1], [2]. Furthermore, the next generation of connected vehicles are expected to support interactive entertainment systems for passengers based on V2V communication [3]. These crucial applications have stringent requirements, and they may need real-time multimedia transmission and execution of complex DL tasks, such as *object detection* or *image segmentation* starting from a raw camera input. This requires the execution of computationally expensive tasks, with strict latency requirements and the need for high throughput links.

F. Raviglione is with the Department of Electronics and Telecommunications, C. Casetti with the Department of Control and Computer Engineering, Politecnico di Torino, Italy.

F. Restuccia is with the Institute for the Wireless Internet of Things, Northeastern University, United States.

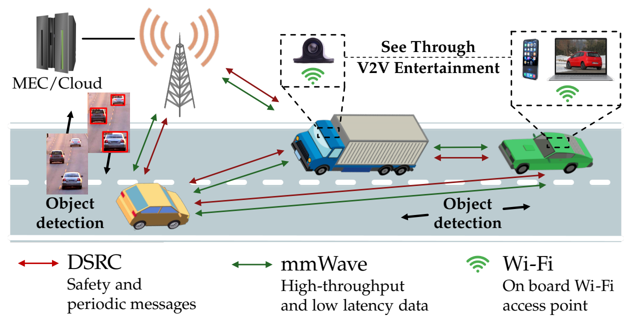

Fig. 1. Vehicular Edge Intelligence (VEI) will enable transforming applications such as "See-Through" vehicular vision, real-time exchange of 3D maps and V2V entertainment.

Due to the limited resources available on vehicles, when compared to the computing power of Multi-Access Edge Computing or cloud servers, task offloading is often looked as a solution to enable complex applications while keeping the on-board resource usage and energy consumption within reasonable limits. This technique can be used to offload computation to the infrastructure and/or to other vehicles that provide free resources, and then receive a lightweight version of the results, containing only the relevant information (e.g., a vehicle can offload an entire frame from a camera output, and receive back the list of detected objects and their position in the frame). Task offloading requires ultra-low latency and very high throughput, and thus it faces several challenges. These challenges include the transmission of data from sensors, such as LIDARs that can generate up to Terabytes per hour of data [4]–[6], and the choice of a proper technology, or a combination of them, to guarantee at the same time high reliability, high throughput and low latency. Reliability is indeed crucial to guarantee that a sufficient number of tasks is properly executed and the results are properly delivered to the vehicles.

With the aim of enabling the execution of such tasks, it may become pivotal to deploy the intelligence at the edge of vehicular networks (i.e., in the vehicles themselves and at infrastructure edge nodes), giving birth to the concept of *Vehicular Edge Intelligence* (VEI).

5G (and, in the future, 6G) is often considered as the solution to the challenges brought by VEI, as its deployment is progressing across Europe and North America. However, relying solely on 5G for this kind of V2X use cases would significantly stress already overloaded licensed spectrum bands, expected to support up to 64 billion subscribers by 2025 [7]. Furthermore, there is currently a lack of open frameworks combining different technologies for V2X to enable practical VEI and task offloading. We thus propose `Edge-V`, the first open framework combining different technologies (DSRC, mmWave and standard Wi-Fi) to enable VEI and on-board AI/ML by leveraging only *unlicensed spectrum bands*. `Edge-V` is designed to be open and it provides several advantages over the usage of 5G only: (i) reduces the usage of the expensive cellular spectrum; (ii) reduces task offloading latency thanks to nearby vehicle offloading and V2V cooperation; (iii) it could be crucial in areas where 5G coverage is scarce or unavailable [8], which, at the time of writing, is often the case in rural areas and small towns.

Additionally, VEI requirements cannot be guaranteed by current vehicular technologies such as Cellular-V2X Mode 4 and IEEE 802.11p due to their limited peak data rates, which are typically below 30 Mbit/s [9]. Although mmWave vehicular networking has been proposed [9], [10], existing work is mostly based on simulations. Furthermore, due to relatively high path loss and potential blockages [11], mmWave technology may experience connectivity issues. Therefore, it is essential to integrate the capabilities of different wireless technologies, with different characteristics, to concretely realize the much-needed VEI.

Summary of novel contributions

- This paper introduces `Edge-V`, the first open framework enabling practical Vehicular Edge Intelligence (VEI) exclusively using unlicensed spectrum and combining the strengths of mmWave, 5.9 GHz ITS DSRC and Wi-Fi.
- VEI includes an Offloading Manager (OM) which manages task offloading based on a formulation of the system model, i.e., the Vehicular Edge Intelligence Problem (VEIP). VEIP, proven NP-Hard, prioritizes offloading tasks to nearby vehicles, minimizing latency; to solve VEIP, we propose a polynomial-time greedy algorithm (DG-VEIP).
- We present a full proof-of-concept using off-the-shelf hardware and open-source software, including an enhanced Local Dynamic Map (LDM) that integrates channel quality and computational load data for VEI.
- `Edge-V` is evaluated via MATLAB simulations, laboratory tests, and on-road experiments. Key results include: significantly reduced latency (up to 65% lower than cloud-only methods) at the cost of a limited accuracy loss (18% mAP), and sub-5 ms end-to-end latency, proving the effectiveness of combined mmWave and sub-6 GHz connectivity. This is the first open-source experimental testbed for VEI in unlicensed spectrum.
- This work significantly extends the previous work presented at VTC2023-Spring [12], by including: (i) a more comprehensive and detailed presentation of the background context, motivations, and up-to-date related works; (ii) a refined version of the POC, with a more complete description of its innovations and a discussion on the use cases that can be enabled by `Edge-V`; (iii) a novel mathematical model of the system, with its NP-hardness proof and the proposal of a greedy scheme for optimizing offloading decisions; (iv) a new evaluation through MATLAB simulation with real-world traces [13]; and, finally (v) a practical experimental validation obtained through field tests and on-road experiments.

II. Related Work

With the advent of dedicated access technologies like IEEE 802.11p and C-V2X, as referenced [14], V2X communications have undergone extensive research in recent years. On the contrary, VEI approaches are still at an early stage of development, given the swift advancement of DL-based algorithms and other use cases that are specifically tailored for vehicular applications [15], [16].

As mentioned earlier, VEI is characterized by the critical need of establishing high-bandwidth and low latency communication among vehicles, which can be potentially moving at high speed. This makes both VEI and task offloading particularly challenging in vehicular scenarios. Indeed, both IEEE 802.11p and C-V2X may be unable to satisfy their stringent requirements if employed alone [9]. Therefore, it becomes crucial to combine different protocols and consider promising high throughput technologies such as mmWave. Even though, with the advent of IEEE 802.11ax, the throughput gain achievable by mmWave may no longer be a decisive factor, the latter still provides several advantages. These include spatial reuse, as technologies working over the sub-6GHz spectrum do not provide a much higher aggregated throughput in a given location.

The application of mmWave to vehicular networks is currently being investigated by a number of research works. Among them, some works focus on IEEE 802.11ad [17], working in the 60 GHz unlicensed spectrum. This technology uses beamforming and can provide ranges up to 100 to 200 m in Line-of-Sight conditions, and 30 to 40 m in Non-Line-of-Sight conditions, guaranteeing very low delays and high throughput, up to more than 1 Gbit/s [18]. Therefore, it appears to be very promising for the application in vehicular networks, although several challenges need to be faced, including beam management, beam blockage recovery [19] and impacting NLOS (Non-Line-of-Sight) conditions, that can reduce the mmWave range by more than 30 times [20].

In addition to mmWave, existing work has also proposed Terahertz-based communications to establish high-bandwidth links [9], [21]. Giordani *et al.* [22] analyzed the application of mmWave to V2X scenarios, while Molina-Galan *et al.* [10] propose to decouple the mmWave data and control plane in vehicular scenarios, and suggest the use of sub-6 GHz C-V2X Mode 4 as control plane to schedule the data transmission over mmWave. As opposed to our proposal, these works are fully simulation-based. Raviglione *et al.* [18] presented one of the first field-test campaigns with mmWave applied to Vehicle-to-Infrastructure (V2I) communications. However, a VEI scenario is not considered and the focus is only on wireless performance. The closest to our work is [9], where Li *et al.* propose a combination of Software-Defined Networks (SDN) and mmWave links, with backhauling based on IEEE 802.11ad and C-V2X, and DSRC to carry control plane signals. `Edge-V` focuses instead on unlicensed spectrum technologies, and does not involve cellular-based communications.

From the point of view of task offloading, instead, a significant amount of work, e.g., [8], [23], [24] has focused on developing algorithms for offloading tasks in the vehicular edge. For instance, Tang *et al.* [25] proposed a cache enabled task offloading system, focusing on offloading tasks to nearby Road Side Units or base stations. Similarly, Higuchi *et al.* [26] define virtual edge servers composed by vehicles, to which jobs can be efficiently offloaded. This work is only evaluated through simulations, without considering the limitations of real hardware.

A key contribution of `Edge-V` is the *combination* of different unlicensed spectrum technologies to enable advanced V2X use cases. As described in more detail in Section III, VEI cannot be enabled by single technologies, and it becomes thus fundamental to incorporate different technologies with different characteristics. Some works in literature propose systems that rely on a single technology, often considering only cellular connectivity. For instance, both [27] and [28] demonstrate the feasibility of real-time offloading with cellular networks. `Edge-V` , on the contrary, improves offloading by considering only unlicensed spectrum and overcoming the limited bandwidth of DSRC with the integration of 60 GHz mmWave, reducing dependence on the cellular network. Other works, like [29] explore offloading with DSRC only, taking traffic priority into account. A pure C-V2X alternative is instead the cluster-based offloading scheme by Bute *et al.* [30], which relies on PC5 sidelink transmissions to distribute tasks among vehicles, together with infrastructured communications to reach remote MEC servers. However, offloading to other vehicles solely over DSRC or Sidelink C-V2X works well only for smaller tasks that do not require a high bandwidth. To this aim `Edge-V` also integrates mmWave, and facilitates the connection of on-board devices thanks to internal standard Wi-Fi.

Our work differs from existing literature as it does not focus exclusively on a specific task offloading architecture, but it aims at providing a complete, open and flexible framework to test VEI and advanced automotive use cases in the field. Indeed, `Edge-V` includes not only the *Offloading Manager*, but several other modules such as an enhanced wireless stack, for which we propose a novel container for Cooperative Awareness Messages (CAMs) [31], that are disseminated with a variable frequency between 1 Hz and 10 Hz depending on the vehicle dynamics, to share VEI-critical information between road users. Concerning task offloading, we also present our own solution and system model starting from the VEIP problem, which is evaluated on both trace-driven simulations as well as on-road experiments, and that can be integrated as a task offloading solution in the *Offloading Manager*.

III. THE EDGE-V FRAMEWORK

The `Edge-V` framework, with its different modules, is represented in Figure 2.

As can be seen, `Edge-V` is designed to be deployed, with different modules, on both connected vehicles and infrastructure nodes (i.e., RSUs, with a connection to a MEC server or cloud data center). `Edge-V` , as a key advancement, combines computation and communication in unlicensed spectrum to deliver V2X and VEI capabilities to existing vehicular networks.

The core of `Edge-V` are its three radio interfaces. Two of them are external for inter-vehicle communication (DRSC and directional mmWave), and one of them is internal to the vehicle (standard Wi-Fi). The latter comprises a Wi-Fi AP bridged with the mmWave interface (the green line represented in Figure 2), to provide the on-board devices with access to the Internet, through the RSU, or to centralized infrastructure services. This enables seamless communication between the on-board devices as well as between different vehicles equipped with mmWave. These devices may include: (i) cameras on-board of different vehicles, enabling use cases such as See-Through with high quality video feedback, (ii) laptops and smartphones for interactive entertainment, and (iii) HMI devices for the reception of warnings, for the provision of road signage information, and for displaying in real-time the position of other road users and/or obstacles (e.g., through the results of DL tasks offloaded to other nodes).

Additionally, the DRSC slot may also host an alternative PC5 Sidelink C-V2X transceiver working in the unlicensed 5.9 GHz spectrum, as no change is required to the higher layers of the ITS stack.

The aim of `Edge-V` is to *combine* these three radio technologies to realize use cases that would not be properly supported by single technologies, but which can benefit from the combined use of these technologies and their strengths. mmWave is ideal for short-range, high-throughput task offloading and multimedia data exchange. DSRC, specifically designed for automotive communications, is suited to safety-critical and standardized messages as defined by ETSI C-ITS standards, thanks to its longer range, but cannot provide data rates higher than 30 Mbit/s [9]. Guaranteeing a level of compliance and integration with standards, through technology specifically designed for the automotive domain, is indeed essential. IEEE 802.11ac/ax Wi-Fi operates in a typical Access Point-client configuration within vehicles, and it is therefore very well suited to connect onboard devices to `Edge-V` , being a commonly supported technology (e.g., by smartphones, unlike DSRC or mmWave).

Furthermore, the presence of the RSU enables data exchange between the on-board devices through the RSU itself, realizing, if needed, a mmWave V2I2V (Vehicle-to-Infrastructure-to-Vehicle) relayed communication. This approach enables the exchange of data between vehicles even in case of noticeable NLOS blockage or distances higher than a few hundred meters. For instance, as shown in [18], [20], mmWave can reach a range of just 30-40 m in NLOS. With a step through an RSU that is in LOS with two vehicles, the range can increase up to 200 to more than 400 m (double the single LOS range measured in [18]).

Relayed communication is an approach that can help improving Vehicle-to-Vehicle communications in presence of NLOS and when far away nodes need to be reached. ETSI has foreseen, for its C-ITS stack, different approaches to perform multi-hop routing, thanks to information available at the GeoNetworking layer. Specifically, both area forwarding (for dissemination of messages inside a target area) and non-area forwarding (for dissemination of messages that still need to reach a destination area) algorithm are foreseen and described in [32]. There are also a number of works in literature that investigate the applicability of multi-hop mmWave V2V communications, such as [33]. These algorithms, however, mostly focus on V2V communication.

On the contrary, `Edge-V` proposes not only an enhanced stack supporting GeoNetworking with forwarding capabilities for the exchange of DSRC messages, but also a forwarding mechanism as part of the RSU, to increase the communication range through mmWave infrastructure nodes. This mechanism can exploit the information received through both dedicated messages and standard-compliant messages, such as the ETSI messages with the GeoNetworking layer.

The combination of the three radio interfaces enables several demanding applications, including (i) DL-task offloading such as object detection from a camera input, as detailed in Section III-A, (ii) raw GNSS data and sensor data exchange with sensors transmitting their data via Wi-Fi to `Edge-V` , and both raw data being transmitted via mmWave, and standard-compliant messages via DSRC, (iii) interactive entertainment and See-Through, with multimedia streams being transmitted by the on-board devices via Wi-Fi and mmWave, and safety messages being broadcasted via DSRC, (iv) centralized and decentralized automated maneuvers, in which safety-critical maneuver management messages are exchanged via DSRC and raw sensor information via mmWave.

It would be possible to argue that combined unlicensed spectrum bands, despite providing several advantages, are also affected by security considerations and possibly additional interference. Nevertheless, `Edge-V` already considers these challenges in three complementary ways. First, two of our three radio interfaces (DSRC and internal Wi-Fi) either work in the ITS band at 5.9 GHz, that is dedicated to vehicular communications, or remain confined to the in-car environment, while the 60 GHz mmWave link combines a short propagation range with beamforming, which drastically limits interference. Second, as described later, Edge-V includes an enhanced wireless stack where security services can be implemented "on top" of Edge-V (e.g., standardized ETSI C-ITS security). Third, in the case of impaired links due to interference, `Edge-V` can take them into account and try to offload to other nodes, as explained in Section IV.

In addition to the wireless technologies, the main modules and components of the framework can be summarized as follows:

- **Computing module.** Each vehicle and RSU has computing resources in terms of RAM, CPU and GPU. These resources can be used to execute tasks and on-board services locally, or can be exploited by other nodes to offload their tasks.
- **Local Dynamic Map.** `Edge-V` includes a standard-compliant LDM [34], which has been enhanced to store information about available resources on nearby vehicles (and, possibly, infrastructure/edge nodes), together with other VEI-specific information. These include computational load and channel load metrics for each vehicle stored in the map. Vehicles running `Edge-V` can exploit such information to determine dangerous road conditions, or to optimize decisions such as where to offload data. The LDM is populated thanks to the exchange of standard-compliant vehicular messages, exchanged through the DSRC link, and it provides enhanced awareness to vehicles, also thanks to the storage of both real-time and historical data. Other than the direct reception via DSRC, the LDM can also receive messages from other vehicles through both multi-hop and from the RSU. This allows the Offloading Manager to have situational awareness, letting it compute the number of mmWave hops which may be needed to reach each node. As the LDM is mainly updated with standard-compliant vehicular messages, they are usually disseminated periodically at frequencies high enough to provide a real-time view over the road and other connected entities.

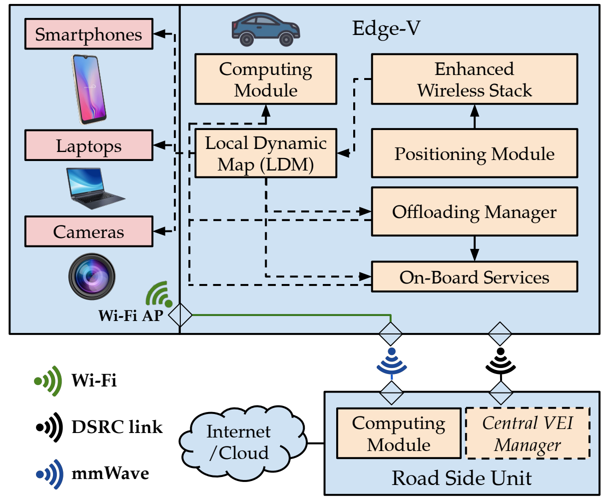

Fig. 2. High-level overview of the `Edge-V` framework.

- **Enhanced Wireless Stack.** Each vehicle is equipped with a standard compliant ITS wireless stack, i.e., an ETSI C-ITS stack for Europe or an IEEE WAVE stack for the US. This stack is enhanced to include an optional container in periodic messages (e.g., CAMs or BSMs) for the transmission and reception of channel and node load data. The broadcast exchange of standard-compliant messages occurs through the DSRC link. This stack should also support forwarding for multi-hop communication (e.g., area and non-area GeoNetworking if an ETSI C-ITS stack is employed). To guarantee QoS, the enhanced wireless stack may rely on MAC-layer traffic prioritization, depending on the needs of on-board services, that can request a given priority, and on the channel load. More specifically, when considering Wi-Fi-based technologies for DSRC and mmWave, 8 user priorities can be set, following the IEEE 802.11D standard. However, Edge-V is designed to support other kinds of prioritization mechanisms, that can be implemented depending on the selected versions of the radio technologies.
- **Positioning Module.** Positioning represents one of the key enablers of vehicular networks. Each vehicle running `Edge-V` is thus equipped with an external or embedded GNSS receiver, providing localisation and kinematic data to the ITS stack for the generation of standard-compliant mes-

sages, which are in turn used to populate the LDMs of the other vehicles. The positioning information is acted upon by the Offloading Manager to perform the decision process. As it represents a quite critical information for the performance of most V2X algorithms (including task offloading), Real-Time Kinematics (RTK) Global Navigation Satellite System (GNSS) receivers can be used to reach lane-level accuracy, and corrections can be received through the DSRC link (using Radio Technical Commission for Maritime Services Extended Messages – RTCMEM – [35]), via raw GNSS data [13], or thanks to the mmWave link towards the RSU. It should be mentioned that future connected vehicles are likely all expected to integrate lane-level GNSS receivers, or at least devices with an accuracy lower than 1 m. Therefore, to consider a real-world scenario, a precise receiver has been used to perform the on-road tests presented in Section VI.

- **On-board Services.** They represent the core on-board services and applications running on the vehicle OBU. They may include, for instance, collision avoidance, object detection through task offloading, See-Through, automated maneuver management, indication of nearby parking spots and road works, and many more. As depicted in Figure 2, these services can retrieve data from the enhanced LDM when needed, including dynamic, node and channel load and network data for each nearby connected vehicle. Additionally, they can also retrieve useful data from (i) the ITS stack, (ii) the on-board devices and sensors, (iii) the high capacity mmWave link, and (iv) the Offloading Manager output. Furthermore, they can use the on-board computing resources to perform computations.
- **Offloading Manager (OM).** This component represents the core of `Edge-V`. It oversees selecting which are the best vehicles to perform task offloading and to communicate with, and deciding which is the best link to use. The decision can be based on the information available in the LDM and on mathematical optimization or AI-based algorithms. The main information used by the OM includes the channel quality towards the destination vehicles, their distance from the source, and their available computing resources. It should be noted how the channel quality, for instance measured via the received signal strength, already takes into account the vehicle's mobility and speed. Indeed, a vehicle that is moving fast will be likely affected by a fast-varying channel quality, and by a distance that increases or decreases quickly and these are all factors that can be taken into account by the OM algorithm. This component is also in charge of managing the local computing resources and determining the best route towards any destination in case multiple mmWave hops are required. Indeed, when performing high-speed data exchange between on-board devices, it can determine which is the best route and provide it to the related On-Board Services. Even though it is flexible enough to accommodate any optimization algorithm, depending on the target scenario, we propose in Section IV a system model, which can be used as a baseline for the algorithm for task offloading and routing.
- **Road Side Unit (RSU).** As mentioned earlier, `Edge-V` can be deployed on an RSU with a smaller and different set of components. As for vehicles, the RSU is equipped with two external interfaces, i.e., mmWave and DSRC, while it does not include any internal Wi-Fi interface. The RSU includes a Computing Module, and, optionally, a Central VEI Manager, and it is connected to the Internet. Indeed, the RSUs are used by `Edge-V` for connection to the broader Internet, enabling further offloading of DL-based tasks to cloud or MEC servers in case no vehicles have enough on-board resources to reach the desired results. They can also execute tasks locally, if needed, as they can host a Multi-Access Edge Computing (MEC) server and use the internal available resources (CPU, RAM and GPU). Furthermore, a Central VEI Manager can be optionally deployed, to centrally manage groups of vehicles in a centralized VEI approach. It should be highlighted that this component is optional, and `Edge-V` is designed to work independently of a centralized unit.

A. An `Edge-V` Walk-Through

After describing the modules and components of `Edge-V`, this Section presents a brief walk-through, in five steps, of the main operations performed by `Edge-V`, focusing on a low-latency, DL-task offloading use case in a VEI scenario. In our example, **(1)** multimedia sensors generate DL tasks (e.g., *object recognition* from the output of a HD camera), which are captured through the Wi-Fi interface. **(2)** At the same time, `Edge-V` is receiving enhanced ITS messages which are used to populate the enhanced LDM, thanks to the enhanced wireless stack. **(3)** As DL tasks are being generated, `Edge-V` checks the available on-board resources thanks to the OM, which directly gathers this information from the Computing Module. **(4)** If the on-board resources, on the vehicle, are not enough, the OM selects the best remote vehicles to offload the task to. If no vehicles with enough resources and stable enough connection are available, task offloading will occur in the MEC/cloud thanks to the communication to one or more RSUs. To compute the needed information, the OM uses the information available in the LDM, solves an optimization problem and provides the results to the On-board service which needs the results of the DL task execution. **(5)** Finally, the On-board service can offload the tasks thanks to the mmWave links.

The procedure herein described is represented in the flow chart in Figure 3.

IV. `EDGE-V`: THE VEIP PROBLEM

To practically enable VEI, optimized task offloading to other vehicles and infrastructure nodes with free resources should be fully supported and integrated with DL-based and other innovative use cases. For this reason, the OM can employ a mathematical model of the whole system to perform optimal decisions on where and how to offload tasks, at each time slot. This problem is called Vehicular Edge Intelligence Problem (VEIP).

The main features of our model for VEIP can be summarized as follows: first, our model prioritizes offloading to other vehicles to reduce the load on the remote edge or cloud servers. Second, it retains generality and does not make any strong

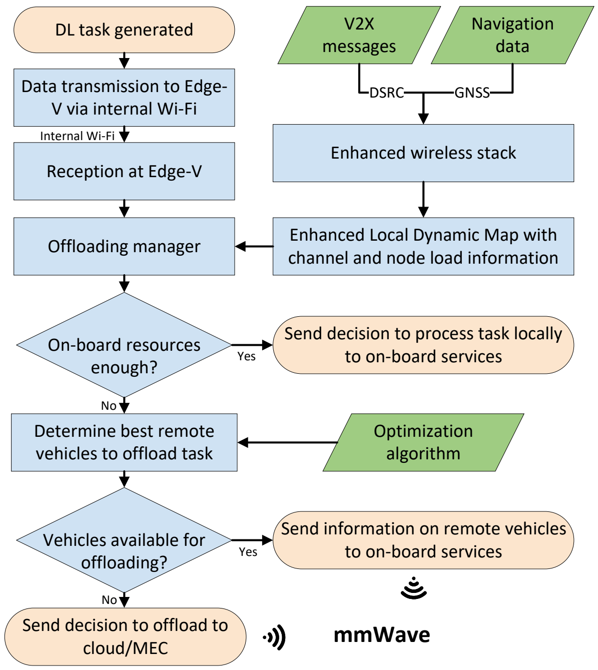

Fig. 3. Flow chart of the operations performed by `Edge-V`, focusing on a DL-task offloading use case.

assumption on the underlying channel model, as opposed, for instance, to [36] which focuses only on IEEE 802.11p. Third, it considers the possibility of multi-hop routing via mmWave thanks to graph modeling. Finally, it is worth mentioning how energy constraints can be neglected, since a system like `Edge-V` is expected to have negligible power consumption when compared to the typical output capacity of a car battery. For instance, our Proof-of-Concept, described in Section V, has a maximum and worst-case power consumption of around 25 W [37]–[39] (less than halogen headlight bulbs), but it is expected to have a much lower average consumption during normal operations. Additionally, with dedicated hardware supporting both DSRC and mmWave, the required power would be even lower. Considering a fuel-powered car battery with a capacity of 50 Ah, providing 12 V and considering a moderate load on our `Edge-V` prototype, with a power consumption of around 10 W, it would require around 60 hours to fully consume the battery when the vehicle is turned off.

Among the related works proposing VEI models and algorithms, it is worth mentioning again the one by Higuchi *et al.* [26]. The authors define task completion probabilities which are shared between vehicles. Tasks are then offloaded based on deadlines. Although this represents a very valid approach, their model differs from VEIP, as our model attempts to explicitly minimize the average task completion latency, keeping the available resources as constraints, considering that, in vehicular networks, most tasks are safety critical and should be executed as soon as possible.

For each DL task, the OM needs to make three main decisions: (i) which location tasks should be offloaded to, i.e., either other vehicles or the cloud/MEC; (ii) how many resources should be reserved on the same vehicles (or requested to the cloud/MEC); (iii) only if the tasks are splittable, which fraction of each task should be assigned to each destination node.

We define as *sources* all vehicles that may offload tasks to other nodes, i.e., other vehicles or the MEC/cloud. Through the shared knowledge built thanks to the DSRC link, each source is supposed to be aware of the full mmWave network topology. At a given time slot, several tasks may need to be fulfilled by each source i.

We define as f_i the number of computations needed for the tasks of each source i with respect to a given reference system (e.g., a given platform with a certain CPU and GPU). Furthermore, we assume no constraints on the available RAM and disk in the destination nodes.

Several input quantities are then defined, all referring to a single time slot:

1) $\mathcal{N} = \{1, ..., n, k\}, |\mathcal{N}| = \nu$, set of all the available n connected nodes; beside the connected vehicles, k is a special node modelling the access to the cloud (or to a MEC server), which can be accessed from all the vehicles.
2) $\mathcal{S} \subseteq N, |\mathcal{S}| = \xi$, set of all the *source* vehicles generating (and possibly performing) tasks at a given rate.
3) $E = \{(w, z) : d_{wz} < d_{lim} \wedge RSSI_{wz} \geq RSSI_{lim}\} \cup \{w, k\}, 1 \leq w \leq n, 1 \leq z \leq n$, set of edges between nodes, i.e., active mmWave links with a good-enough signal. Furthermore, d_{lim} is the distance limit above which any mmWave link is considered unstable, while $RSSI_{lim}$ is the RSSI limit below which any mmWave is considered unstable. The RSSI also considers real-time fluctuations in the wireless channel, due to weather and blockage. Thanks to this, when a degraded channel quality occurs due to weather or blockage, the RSSI will decrease, and the OM will be able to select other nodes for offloading, if available, mitigating as much as possible the effect of channel quality variations.
4) $G = (\mathcal{N}, E)$, graph of the current network topology, whose edges are represented by valid and stable mmWave links.
5) $\mathcal{C} = \{c_1, ..., c_\nu\}$, set of the currently available computational capacity of each node, in terms of computations per second, with respect to a given reference system, as defined for f_i. The computational capacity is a function of the available CPU and GPU for each node j: $c_j = \alpha(CPU_j, GPU_j)$.
6) $\mathbb{R}_{ij} = \{(i, h_1, ..., h_z, j) : (i, h_1) \in E, (h_z, j) \in E, (h_i, h_{i+1}) \in E, 1 \leq i \leq z - 1\}$, set of all routes from each source i to any possible destination $j \in \mathcal{N}$ in the network.
7) $L(i, j)$, latency of the route from source i to destination j. Assuming a symmetric channel, the Round Trip Time (RTT) can be represented by $2 \cdot L(i, j)$. This term depends on the amount of data D_{ij} which needs to be transmitted from each source vehicle to each destination node, on the MAC-layer priority of the traffic P_{ij}, and on the average wireless channel data rate b_{ij}. In turn, D_{ij} depends on the actual task fraction assignment to each destination node: $D_{ij} = \beta(\tau_{ij})$, where β maps the task fraction to each destination node to the data which needs to be sent to that node.

8) o_j, overhead time of each destination node j. This time accounts for the overhead due to the message reception in the target operating system, data encoding and decoding and all the operations which do not depend on the size of the actual task.

We also define a set of output quantities for each time slot, coming from the solution to the optimization problem:

1) $\mathcal{Y}^* = \{y_{11}, ..., y_{ij}, ...y_{\nu\nu}\}$, set of binary variables equal to 1 if node j has been selected as destination node by source node i, 0 otherwise.
2) $\mathcal{A}_i^* = \{a_{i1}, ..., a_{i\nu}\}$, set of optimal resource assignment to each destination node in the network for the current source i. All nodes j such that $y_{ij} = 1$ will be destination nodes towards which the tasks are offloaded, and are expected to have some resources assigned to perform the tasks ($a_{ij} \geq a_{min}$, where a_{min} is the minimum amount of resources than can be reserved). Each a_{ij} is defined in terms of computations per second.
3) $\mathcal{T}_i^* = \{\tau_{i1}, ..., \tau_{i\nu}\}$, set of optimal task subdivision to each node in the network for the current source i. Each τ_{ij} represents the fraction of task f_i assigned to the destination node j. Each destination node $j \in \mathcal{D}$ should have at least one non-zero τ_{ij}, while each non-destination node j should have all its $\tau_{ij} = 0$.
4) $\mathcal{D} \subseteq \mathcal{N}$, set of selected destination nodes to which the tasks should be offloaded. Each node j that have at least one $y_{ij} = 1$ is a destination node.
5) $\delta = |\mathcal{D}|$, cardinality of the set $\mathcal{D}$, i.e., the total number of destination nodes.
6) $\mathbb{R}_{ij}^*$, set of selected optimal routes from each source i to the chosen destinations j.

The VEIP problem is then modelled as follows:

$$\text{Find} \quad \mathcal{A}_i^*, 1 \leq i \leq \xi \tag{1a}$$
$$\mathcal{T}_i^*, 1 \leq i \leq \xi \tag{1b}$$
$$\mathcal{Y}_i^*, 1 \leq i \leq \xi \tag{1c}$$
$$\mathcal{D} \subseteq \mathcal{N} : \{1 \leq j \leq \nu : y_{ij} = 1, 1 \leq i \leq \xi\} \tag{1d}$$
$$\mathbb{R}_{ij}^* \subseteq \mathbb{R}_{ij}, 1 \leq i \leq \xi \tag{1e}$$

such that:

$$\text{minimize } i,j \quad \sum_{i \in S} \sum_{j=1}^{\nu} \{2 \cdot L(i,j) + \frac{\tau_{ij}}{a_{ij}} + o_j\} \cdot \frac{y_{ij}}{\delta} \tag{1f}$$
$$\text{subject to:} \quad \delta = \sum_{i \in S} \sum_{j=1}^{\nu} y_{ij} \tag{1g}$$
$$\sum_i a_{ij} \cdot y_{ij} \leq c_j, \forall j \tag{1h}$$
$$a_{ij} \geq a_{min}, \forall i,j \tag{1i}$$
$$\sum_j y_{ij} \geq 1, \forall i \tag{1j}$$
$$\sum_j \tau_{ij} = f_i, \forall i \tag{1k}$$
$$\tau_{ij} \leq M \cdot y_{ij}, \forall i,j \tag{1l}$$
$$\tau_{ij} \geq \tau_{min} \cdot y_{ij}, \forall i,j \tag{1m}$$

The selected destination nodes and routes from sources to destinations are defined by the sets (1d) and (1e). The objective function (1f), instead, aims at minimizing the overall average latency and it is composed of multiple terms: (i) the RTT of the path towards the destination node; (ii) the computation latency, defined as the ratio between the number of computations needed to fulfill tasks from source vehicle i and the resources (number of computations per second) assigned to the destination node j for the tasks of source node i; (iii) the overhead time, as defined earlier.

Each term of the sum is divided by δ, that, as defined by equality Constraint (1g), represents the total number of destination nodes. This allows us to consider the minimization of the *average* latency, and not of the sum of all the latency values, that would penalize solutions that split more to achieve a lower delay for each couple of nodes (i, j).

Constraint (1h) is the capacity constraint, ensuring that we cannot assign to a destination node (i.e., a node j for which at least one y_{ij} is 1) more capacity than its current availability, while Constraint (1j) forces each task to be executed. Constraint (1i), instead, forces each destination node to be assigned at least the minimum possible amount of resources that can be reserved (a_{min}). This also ensures the positivity of a_{ij}, that appears at the denominator of the objective function.

Constraint (1k) forces each task from source i to be completely fulfilled by the selected destination nodes j. Finally, Constraints (1l) and (1m) make each destination node j with $y_{ij} = 1$ perform at least a fraction of task τ_{ij}, and each non-destination node j with $y_{ij} = 0$ perform no computations for the current task f_i. In our notation, M represents an upper bound to the value of each τ_{ij}, while τ_{lim} is a lower bound to the value of each τ_{ij}.

After formulating VEIP and its mathematical optimization model, we demonstrate that it is NP-Hard and propose an efficient, lightweight greedy solution that can be employed by the OM thanks to the development of proper open source software.

Finally, it should be noted that VEIP is a Mixed-Integer Quadratically Constrained Program (MIQCP), since it can be reduced to a problem with quadratic terms both in the objective function and in the constraints [40].

Theorem 1. *The proposed problem (VEIP) is NP-Hard.*

Proof. We prove the result by showing that the Knapsack problem (0/1 KP), which is known to be NP-hard [41], can be reduced to an instance of VEIP.

To simplify and reduce the problem, we can make the following assumptions:

1) one destination node only (e.g., the cloud) can be selected for all the source nodes, thus, this allows us to remove the double sum in the objective function and the *j* index; furthermore, this allows us to remove Constraint (1g) and δ from the objective function as it will always be $\delta = 1$;
2) the problem is now supposed to be unsplittable (like in the case of some scenarios, such as single frame inference), enabling us to replace τ_{ij} with f_i and remove constraints from (1k) to (1m);
3) task are allowed to be unfulfilled (this removes Constraint (1j));

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3613703

4) the resource allocation for each source task a_i is now given as a positive number greater than a_{min}, and it does not represent anymore a decision variable.

Under these assumptions, the problem now has only one set of decision variables ($y_i, \forall i$), while the other values are known input quantities. The problem can be thus rewritten as:

$$\text{minimize } i,j \qquad \sum_{i=1}^{\xi}(2 \cdot L(i) + \frac{f_i}{a_i} + o) \cdot y_i = \sum_{i}^{\xi} k_i \cdot y_i \tag{2a}$$

$$\text{subject to:} \qquad \sum_{i}^{\xi} a_i \cdot y_i \leq c \tag{2b}$$

This is an instance of an NP-Hard 0/1 KP problem, thus also VEIP is NP-Hard.

□

A. DG-VEIP: A Greedy Solution to VEIP

As the VEIP problem proved to be NP-Hard, the OM can employ an efficient Greedy Algorithm to solve it. We thus propose a distributed VEIP algorithm, here referred to as *DG-VEIP*. As an assumption, we consider the cloud/MEC as always reachable and available. Furthermore, DG-VEIP requires $L(i, j)$ to be properly set depending on the scenario and on the set of tasks, and it is designed to be executed for each source vehicle i. Finally, we assume that connected vehicles are equipped with lane-level accurate GNSS receivers, as discussed earlier.

The algorithm pseudocode is reported in *Algorithm 1*.

Algorithm 1 Distributed Greedy VEIP Algorithm (DG-VEIP)

1: Define an empty list of vehicles V
2: Define the remaining task fraction to be assigned $\tau_{rem} \leftarrow f_i$
3: **for** $n \in \mathcal{N}$: path between i and n exists in $\mathbb{R}_{ij}$ **do**
4:   **if** $n \neq k$ **then**
5:     Compute $\frac{d_n}{c_n}$
6:   **else**
7:     $\frac{d_n}{c_n} \leftarrow \infty$
8:   **end if**
9:   Add node to list V.
10: **end for**
11: Sort list V by ascending $\frac{d_n}{c_n}$
12: **for** $v \in V$ and $\tau_{rem} > 0$ **do**
13:   **if** $c_v > 0$ **then**
14:     Request all available capacity to the node: $a_{iv} \leftarrow c_v$
15:     **if** $v \neq k$ **then**
16:       $c_v \leftarrow 0$
17:     **end if**
18:     Assign task fraction τ_{iv} such that it is completed before or in a t_i time
19:     $\tau_{rem} \leftarrow \tau_{rem} - \tau_{iv}$
20:   **end if**
21: **end for**

At each time step t_i, each source vehicle i generates a number of tasks to be executed. The number of total computations needed f_i is used to represent these tasks.

Thanks to the enhanced LDM, each vehicle can scan the list of the $\mathcal{N}$ connected nodes (which can include the vehicle itself at $d_n = 0$), and add them to a new list V (line 9), which is sorted in ascending order by the ratio of the distance and the available node capacity (which can be referred to as *cost*). The cloud (or MEC) node is artificially assigned the highest ratio among all other nodes (ideally ∞), so that it will be selected last, only if needed (line 7). As mentioned earlier, the distance already takes into account the mobility pattern followed by the vehicles. Indeed, the speed of the vehicles will directly affect how the distance evolves at each time step t_i for which DG-VEIP is executed, which in turn influences each *cost* term. The algorithm then loops over all the nodes in V (line 11) until all the f_i computations have been offloaded (or performed by itself or by the cloud). The variable τ_{rem} represents the number of remaining computations which need to be offloaded. If a node in V is found with free computation resources (line 12), the source vehicle will require all its capacity (line 13), to try to minimize the latency, and assign a task fraction τ_{iv} such that the task is either completed in a t_i time, meeting the deadline of the next time frame, or before that time, if the node has more computation resources available (line 17). Finally, it should be mentioned that the cloud is supposed with no hard resource limits, and it can thus be always selected as last possibility when no other local vehicles can be selected. DG-VEIP has been implemented in a dedicated MATLAB function, which has been exploited to evaluate its usage within `Edge-V` and which is going to be become publicly available, on GitHub, under the GPLv2 license.

DG-VEIP represents a sample, yet effective, greedy algorithm that can be integrated into the OM to provide solutions with a limited complexity of $O(n \cdot log(n))$. Indeed, the first for loop has a complexity of $O(n)$, and the second for loop $O(v)$ with $v \leq n$ (thus, at worst it will also run in $O(n)$). The most complex operation from an algorithmic point of view is instead sorting (line 11), that can be implemented to be executed in $O(n \cdot log(n))$. Combining the complexity of the different phases, we get $O(n) + O(n) + O(n \cdot log(n))$, with the most significant term being $O(n \cdot log(n))$. Therefore, the overall algorithm complexity is $O(n \cdot log(n))$.

It should be mentioned that DG-VEIP represents a baseline algorithm to efficiently offload tasks in `Edge-V`. More advanced versions can be easily implemented as part of the OM, and they are currently being developed, including an enhancement that considers the channel RSSI, in addition to $\frac{d_n}{c_n}$.

Finally, it should be mentioned that a simplified version of DG-VEIP has been employed to demonstrate the effectiveness of `Edge-V` in the field (Section VI-C2), as a possible Offloading Manager algorithm implementation.

V. `Edge-V`: Prototype Design

Figure 4 provides a high-level overview of our Proof-of-Concept. As can be seen, the `Edge-V` prototype is designed to be deployed inside an RSU Road Side Unit (RSU) and multiple OBUs. Specifically, we assembled three OBUs, to be deployed on up to three vehicles, and one RSU, which we then used to evaluate `Edge-V` on the field and in a laboratory environment. As mentioned, `Edge-V` is characterized by its openness. Therefore, all key features of our framework have been implemented using low-cost, commercially-available hardware and with well-documented open source software components. We selected three IEEE 802.11-based standard, i.e., IEEE 802.11ac for the 5 GHz Wi-Fi internal connectivity, IEEE 802.11p for the DSRC link, and IEEE 802.11ad for mmWave at 60 GHz. It should be mentioned how a commercial C-V2X module supporting Sidelink communication at 5.9 GHz could be possibly employed instead of IEEE 802.11p, yielding a similar performance with slightly higher network latency [42]. However, we focus on IEEE 802.11p due to the wider hardware availability and being it the most deployed technology in Europe.

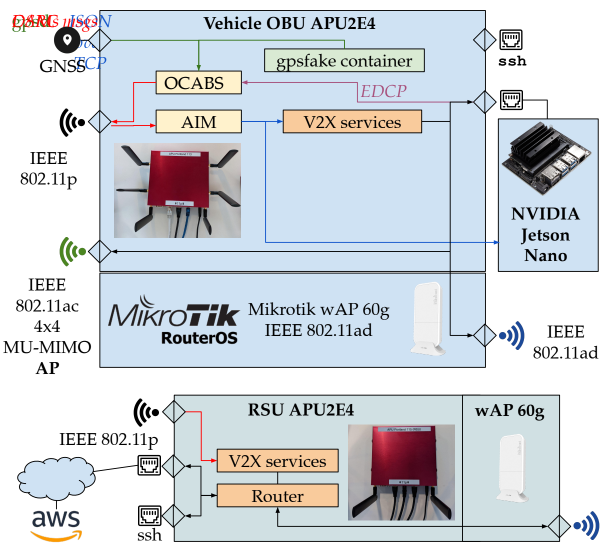

Fig. 4. The proof-of-concept prototype of `Edge-V`.

A. Hardware Components

The prototype hinges upon customizable embedded PC Engines APU2E4 hardware boards [37], equipped with a quad-core AMD GX-412TC 1 GHz embedded CPU and 4 GB of RAM, running the latest version of OpenWrt-V2X (21.02) and with a maximum power consumption of around 12 W. OpenWrt-V2X is a patched version of the OpenWrt embedded Linux distribution, enabling IEEE 802.11p communication and providing additional tools for automotive services [43]. This version of OpenWrt has been successfully evaluated in past research works, together with 5.9 GHz-enabled UNEX DHXA-222 wireless cards, in both static [44] and dynamic scenarios [18].

Each board has been enhanced with the installation of:

- An Atheros AR5B22 mPCIe wireless module for the IEEE 802.11p interface. This module relies on the same chipset (AR9462) as the UNEX DHXA-222 cards, and it is characterized by the same performance and maximum selectable transmission power (18 dBm).
- A Compex WLE1216V5-23 Multi-User Multiple-Input and Multiple-Output (MU-MIMO) 4x4 IEEE 802.11ac mPCIe card, providing a relatively high maximum transmission power (up to 29 dBm with MCS 0). As this module requires a 5V additional power supply, we used mini PCI express extender cards, which have been installed in one of the APU2E4 miniPCIe slots and on which we soldered a jumper cable. The cable has been soldered, in particular, to the reserved pins of the extender cards (45, 47, 49, 51), which can also be used to provide an additional power supply to the Compex chips.

Each APU2 board (both OBUs and RSU) is connected to a MikroTik wAP 60G [39], through one of the available Ethernet ports. These devices are IEEE 802.11ad routers working in the 60 GHz unlicensed spectrum and equipped with 6x6 planar phased antenna arrays, covering an angular range of 60 degrees. According to [18], they are able to reach up to 300 m with an RSSI higher than -70 dBm, in nearly ideal Line-Of-Sight (LOS) conditions. Since these devices are still unable to establish direct peer-to-peer links, we deployed an additional MikroTik wAP 60Gx3 Access Point, to which all the devices in the testbed are connected and acting as relay node. This strategy makes the client devices appear as if they are directly connected together. Finally, since the APU2 boards do not embed a GPU for executing DL tasks, we interfaced each board (except the RSU one) with an Nvidia Jetson Nano Development Kit. The Nvidia Jetson Nano can be exploited for GPU computation and can be easily connected to an APU2 board through a Gigabit Ethernet point-to-point link.

It should be mentioned how the Nvidia Jetson Nano employs an ARM Cortex-A57 MPCore CPU, a GPU with a performance up to 512 GFLOPS, and 4 GB of RAM. Due to the available resources, with a model such as MobileNet v3 [45], a Jetson Nano is typically able to perform inference on one frame at a time.

B. Software Components

The European ETSI ITS-G5 set of standards has been taken as reference for the implementation of the enhanced wireless stack. Therefore, CAM messages are used to periodically broadcast, via IEEE 802.11p, vehicle information such as speed, acceleration, position and heading. However, there are currently no standardized messages for the exchange of channel-related and load-related metrics, as required by `Edge-V`. This data is of utmost importance to enable the next generation edge intelligence use cases, which require, at the same time, offloading and local computation to reduce latency. Hence, we designed an additional, ETSI-compliant, optional container (the *Channel and Node Status Container*) that can be inserted inside standard CAM messages, enhancing them with the information needed to enable VEI and task offloading. Our proposed container has been defined by upgrading the standard CAM specifications, written in the ASN.1 description language [31]. Starting from the ASN.1 definition, it was then possible to generate the code for the encoding and decoding functions thanks to the *asn1c* tool [46].

The supplementary container[1] comprises various information, including (i) the load on CPU, GPU, and RAM of the OBU, (ii) available disk space, (iii) RSSI and data rate of both V2X-dedicated (e.g., DSRC) and additional (e.g., mmWave)

[1]The ASN.1 file is available here: https://github.com/francescoraves483/EnhancedCAMs-asn1

links, (iv) IP addresses of on-board services, and (v) MAC addresses of on-board devices.

In addition to the definition of enhanced CAMs, we developed a new software module called *Open Cooperative Awareness (CA) Basic Service* (OCABS)[2] to implement the ETSI CA Basic Service. It includes an implementation of GeoNetworking [32], Basic Transport Protocol [47] and the Facilities Layer for the transmission of both standard and our enhanced CAMs. OCABS needs GNSS data as input, in order to properly encode and broadcast the CAMs. This data can come either from pre-recorded traces (which are then replayed in a container thanks to tools like `gpsfake`), or from a USB GNSS receiver, thanks to the default Linux GNSS daemon `gpsd`. Furthermore, we developed a novel *Automotive Integrated Map* (AIM)[3] to realize the `Edge-V` LDM. AIM can receive and decode enhanced CAMs, and store the information contained in the *Channel and Node Status Container* into the local database, and V2X services can request the needed information (including the VEI-specific data) through a JSON-over-TCP interface. AIM is designed to be updated following the ETSI standards, that foresee a CAM transmission frequency between 1 Hz and 10 Hz, depending on the vehicle dynamics and kinematics. This helps reducing the load on the channel when kinematics are slowly evolving, increasing the frequency when more frequent updates are needed due to rapidly changing dynamics, and guaranteeing proper update rates for use cases such as task offloading. Finally, a novel lightweight protocol, encapsulated inside UDP, has been defined to transfer with low latency CPU, GPU and RAM usage of the Nvidia Jetson Nano to OCABS. This custom protocol, called Extra Device Communication Protocol (EDCP) is based on a client-server paradigm. Each Nvidia Jetson Nano runs a server as a service, waiting for requests from OCABS (acting as EDCP client). Every time a request is received, it is parsed and a reply is immediately generated, containing the resource usage information. This reply is then sent to the EDCP client (i.e., OCABS) running on the APU2 board, and the information within is used to populate enhanced CAMs.

VI. `Edge-V` : Performance Evaluation

Thanks to our prototype, it was possible to evaluate `Edge-V` both in a laboratory environment and on the road with two real vehicles and one RSU. Two significant use cases have been investigated: (i) direct data exchange between vehicles with high throughput and low latency; (ii) DL-based object detection in a VEI scenario. Before presenting the results of the tests with our prototype, we have also evaluated `Edge-V` through trace-based simulations, considering an OM implementing DG-VEIP.

A. Simulation with Vehicular Traces

An ad-hoc MATLAB simulator, integrating a function with DG-VEIP, has been developed to simulate a vehicular communication system starting from the SAMARCANDA dataset [13]. This dataset, in its CSV version, comprises the traces of 19 real vehicles travelling in an area near Turin, Italy. All vehicles are equipped with `Edge-V` and are supposed to be able to reach the cloud through a proper deployment of RSUs in the simulated scenario. As baselines for comparison, we consider both the case in which only a remote cloud node is leveraged (i.e., the same cloud node to which vehicles can send their tasks when no neighboring vehicles are available), and the case of a nearby MEC server satisfying the requests from vehicles.

The following simulation parameters have been used, to analyze a high-load scenario in which vehicles are busy performing other local tasks in addition to the tasks to offload:

- total simulation time: $1124\,s$, corresponding to the length in time of the shortest trace in SAMARCANDA;
- task generation and DG-VEIP execution periodicity: $1\,s$;
- mmWave V2V latency $L(i,j) = 0.7\,ms$, consistently with the results of [18];
- overhead time $o_j \sim 0$, i.e., considered to be negligible;
- mmWave cloud latency $L(i,k)$: Generalized Extreme Value latency distribution with $\sigma = 6.89932\ ms$, $\mu = 64.5928\,ms$ and $\xi = 0.11209$, obtained from several measurements from our laboratory towards a real cloud Amazon AWS virtual machine;
- MEC latency $L(i,k)$, when a MEC server is considered instead of a cloud node for the baselines: Logistic latency distribution with $\mu = 10.8047\,ms$ and $\sigma = 1.1361\,ms$, obtained from real measurements towards a MEC server located less than 1 km away;
- $d_{lim} = 140$, consistently with the `Edge-V` road tests described later;
- $\mathcal{S} = \mathcal{N} \backslash \{k\}$, i.e., each vehicle has been considered as both a source and possible destination node;
- each source always offloads its own tasks, without fulfilling them by itself;
- cloud capacity $c_k = 45\,\frac{computations}{s}$
- MEC server capacity, when a MEC server is considered instead of a cloud node for the baselines, $c_k = 45\,\frac{computations}{s}$
- varying task size $f_i = [0.1, 0.5, 1, 10, 20, ..., 60]$;
- maximum task deadline: $1\,s$.

Figure 5 plots the comparison of our DG-VEIP algorithm with respect to (i) a cloud-only baseline, (ii) a MEC server-only baseline, and (iii) a solution gathered through the Gurobi solver, with different values of task size f_i and by assigning a random remaining capacity to each vehicle at each time frame between 0 and a maximum capacity value.

The Gurobi solution has been obtained as a baseline to show what can be achieved through a nearly optimal solution, that, however, requires significantly more time to be computed. Additionally, running a solver like Gurobi on an embedded OBU of a vehicle would be hardly feasible as they are not designed for execution on small embedded operating systems.

It should be noted how the Gurobi baseline does not represent the optimal solution, as the solver was stopped after 1 second of computations for each time frame. On the one hand, the choice of a 1 second time limit was necessary to gather our results in a reasonable time, as computing the

[2]https://github.com/francescoraves483/OCABS-project
[3]https://github.com/francescoraves483/AIM-AutomotiveIntegratedMap

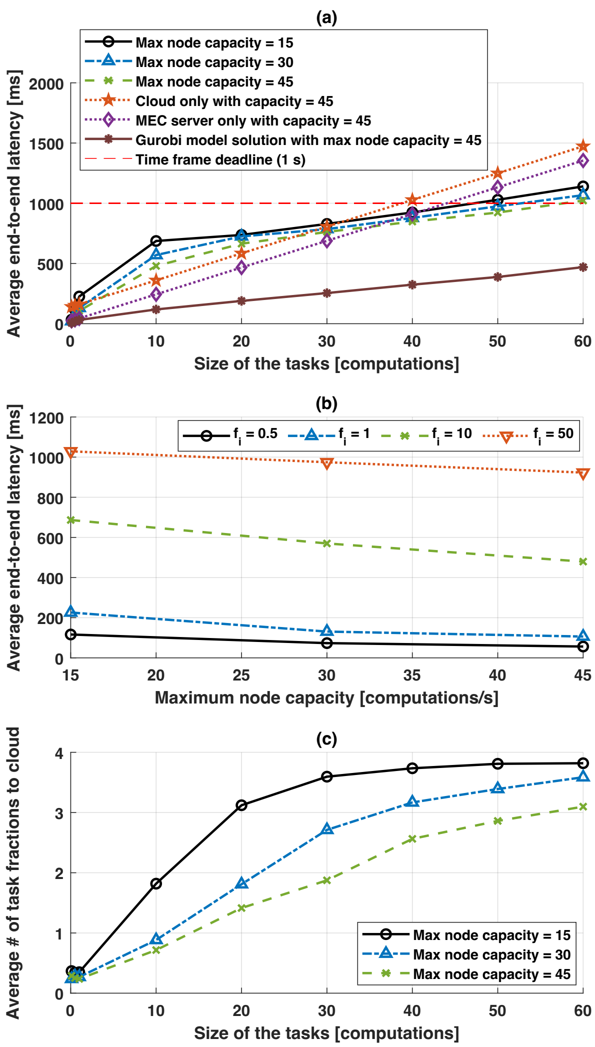

Fig. 5. Average end-to-end latency of the DG-VEIP as a function of (a) task size, (b) maximum On-Board Unit computational capacity. Plot (c) shows the total number of task fractions offloaded to cloud due to unavailability of neighboring vehicles.

optimal solution of a complex MIQCP problem could require a very long amount of time for each time step. On the other hand, this time already exceeds the requirements of real-time vehicular networks, in which latency may be required to be as low as 100 ms, and in which the minimum standardized message periodicity for CAMs is 1 second [31]. Additionally, experiments show it already yields results not too far from the optimal and a fair baseline for DG-VEIP: on an instance with $f_i = 20$, raising the limit to 5, 10, 20 s improves average latency by only 24.4%, 29.8%, 30.8%, with gains plateauing after around 20 seconds.

When compared to a cloud-only baseline, our VEI solution is much more effective when large tasks are being offloaded, such as images for object detection. This is due to the larger tasks being splitted with a higher probability, depending on the available neighboring vehicle capacity. These tasks can thus be more efficiently splitted and sent to nearby vehicles, that will be able to execute the task fractions in parallel and then send back the results with the low latency guaranteed by V2V.

For instance, a set of tasks of size $f_i = 50$ can be executed within the deadline by `Edge-V`, as opposed to a cloud-only approach when cloud capacity is 45 computations/s. Although the results could be improved with more advanced optimization techniques, it should be noted that the obtained trend should retain generality, with large tasks benefiting more from local offloading than smaller tasks. Furthermore, the results show how an increase in the maximum local vehicle capacity can lead to an improved overall latency in a VEI approach.

When considering instead a MEC server-only baseline, the advantages of leveraging computing resources located nearby the end user are clearly expressed by a 9% decrease of the end-to-end latency with respect to the cloud-only baseline. However, DG-VEIP appears to outperform MEC-only cases for large tasks.

Figure 5(c) illustrates the total number of task fractions offloaded to the cloud during the whole simulation time. These fractions, which may represent entire tasks, have been offloaded to the cloud due to the absence of nearby connected vehicles with enough free resources. This trend proves how, for larger tasks, splitting is more efficient and leads to a stabilization in the number of task fractions offloaded to the cloud for task sizes greater than 30 computations per second, further justifying the behaviour observed in Figure 5(a).

Finally, it is worth noticing how DG-VEIP takes significantly less time to provide a solution that, even if it leads to a worse average latency than the Gurobi baseline, lets vehicles complete the tasks within a given deadline up to a task size of 50. Indeed, Gurobi takes 1 full second to compute the solution depicted in the plot, that would already be enough to miss the deadline if deployed on vehicles instead of DG-VEIP.

B. Laboratory tests

1) Vehicular Data Exchange use case: The first use case generalizes several automotive applications requiring a low-latency direct exchange of data, such as video streaming, *See-Through*, and online gaming. We employed our prototype, in which two laptops (i.e., on-board devices that need to exchange data) are associated with the IEEE 802.11ac access point generated by the respective APU2 boards. We performed several latency and throughput tests to evaluate the performance of our framework, through the prototype hardware.

The tests have been performed with the LTNT measurement framework software [48]. Thanks to LTNT, it was possible to reliably measure RTT and throughput between two OBU boards (i.e., between two vehicles in a static scenario, with the aim of performing a baseline assessment). The results are depicted in Figure 6.

The plots show the Cumulative Distribution Function (CDF) for both throughput and RTT, directly measured between two APU boards. The top plots depict the results of direct mmWave Vehicle-to-Vehicle communication, while the bottom plots illustrate the results of a longer-range V2I2V scenario, where the RSU acts as a message relay between the boards. As can be seen, the overall RTT always remains below 5 ms, with an average of around 1.4 ms for the direct communication case (V2V) and 3.7 ms for the relayed communication (V2I2V).

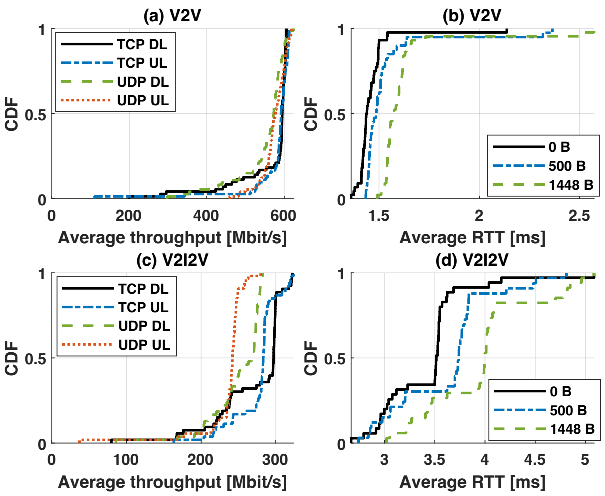

Fig. 6. CDFs of the mmWave latency and throughput during a one-day-long test. (a) V2V throughput (b) V2V latency (b) V2I2V throughput (through RSU) (d) V2I2V latency.

This proves that mmWave is suitable for a very low-latency high-throughput scenario, as it reaches more than 500 Mbit/s when using both TCP and UDP. This value, despite it could actually increase without the extra step through the mmWave AP, further highlights the benefits that mmWave technology could bring to the next generation of automotive applications.

2) Object detection and task offloading use case: The second use case showcases a DL task offloading application enabled by `Edge-V`, considering three OBUs and one RSU connected to the cloud, i.e., to the Amazon AWS Virtual Machine through the laboratory network. It should be recalled how task offloading can be beneficial both to avoid expensive hardware OP, and to make vehicles able to manage a significant number of tasks in parallel, which could exhaust the resources available on-board. The less safety critical tasks (but still with strict latency and throughput requirements) can be thus offloaded to other vehicles (or, if needed, to the infrastructure) that provide free resources.

With the aim of testing DL task offloading, an object detection service based on the Microsoft Common Objects in Context (COCO) dataset [49] has been developed. This system has been integrated as an on-board service in `Edge-V` and it enables a vehicle, each time, to offload either to other vehicles or the cloud. The OBUs implement the vehicular edge, while a Virtual Machine (a *t2.2xlarge* Amazon AWS instance with 8 virtual CPUs and 32 GB of RAM) has been adopted to implement the cloud. The latter can be reached thanks to the RSU, either connected to our laboratory network, concerning the indoor tests, or to the T-Mobile network thanks to an LTE link, concerning the outdoor tests. The OBUs rely on a less accurate, but less computationally expensive, object detection model, namely *Faster R-CNN Large*, with MobileNet V3 backbone [45] and executed on the Jetson Nano boards. On the other hand, the cloud employs YOLOX-s [50], more computationally demanding but also more accurate.

Due to the available resources on the RSU itself, the latter has has not been considered as an offloading edge node, i.e., offloading occurs on the cloud through the RSU itself. However, it is important to note that a more capable RSU could indeed be considered as a node for offloading tasks and hosting a MEC server.

In addition to the object detection system, we realized an implementation of the Offloading Manager. Notably, we developed two components. The first is an *Object Detection Offloading Manager*, running on one OBU. It includes a full COCO dataset to emulate frames coming from an on-board camera, on which object detection needs to be performed. As the aim is to implement an Offloading Manager based on the VEIP model, the decisions need to be taken to minimize the overall average latency. Therefore, this component will select the best destinations for each offloaded task through the information available within AIM, including available resources on the target, distance from the target and RSSI of the mmWave channel to the target. Offloading is performed to the cloud through the RSU only if no vehicles near enough or with enough free resources are available. To this aim, the Object Detection Offloading Manager implements a simplified version of DG-VEIP, in which offloading to another vehicle occurs only if it is within a certain mmWave RSSI and distance (i.e., if the mmWave link can guarantee a stable connectivity and a low $L(i, j)$) and if it has enough resources available (i.e., if it satisfies the VEIP Constraint (1h) in the case of one destination node only). The second component is instead an *Object Detection Offloading Worker*, running on the other OBUs and on the cloud. This component represents the implementation of a V2X on-board service waiting for frames from other nodes, on which object detection should be performed. When frames are received from the *Object Detection Offloading Manager*, it performs the inference, and then returns the detection results to the sending node, in a JSON format.

The results of a test session on the full COCO dataset (i.e., 5000 images, in the selected version – *2017 Val images* [49]) are shown in Figure 7 (with 95% confidence intervals). The plots depict the average end-to-end task offloading and object detection latency, and the mean average precision, as a function of the frame offloading frequency, which has been varied from one image every 1.6 s (i.e., 0.625 Hz) to 10 images per second (i.e, 10 Hz). A cloud-only approach has been considered as a baseline. With the aim of providing comparable results, we measured the time needed by the cloud to perform inference on the Nvidia Jetson Nano model and assigned that computing time to each actual inference on the Nvidia boards, for each image, instead of considering the embedded board computing times. This is technically sound, as actual vehicles are expected to provide much better computation power.

As can be seen in Figure 7(a), offloading to nearby vehicles leads to a noticeably reduced overall latency. This comes with a minor loss in terms of precision, as depicted in Figure 7(b). Indeed, offloading to nearby vehicles can help to reduce the end-to-end latency of more than 150 ms, up to 5 Hz. The best result is achieved at 1.67 Hz, with up to 286 ms latency saving, corresponding to a 59% improvement with respect to a cloud-only baseline. Despite a slight decrease in precision from 0.403 to approximately 0.328, the reduction in latency is considerably more significant, showing the advantages of `Edge-V`.

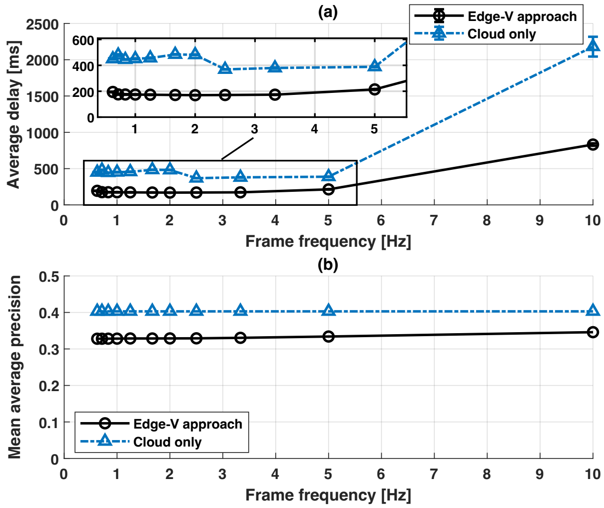

Fig. 7. (a) End-to-end processing and network latency and (b) Overall mean average precision accuracy of the object detection, as a function of the image generation frequency.

In addition, it is possible to notice how the latency of `Edge-V` stays around 200 ms under 5 Hz, and then raises noticeably when the frame frequency changes to 10 Hz, with a slight increase already observable over 3 Hz. This also corresponds to a small accuracy increase. This behavior arises as each OBU is limited by the resources available in each Nvidia Jetson Nano. Indeed, as stated earlier, each Nvidia board can only perform inference on one frame at a time. The Offloading Manager may find that no vehicle is available for offloading when the frequency is high enough, thus automatically sending that frame to the cloud, which causes a higher latency but provides slightly better accuracy. This behaviour becomes more noticeable as the frequency is increased to 10 Hz as frames are more likely to be offloaded to the cloud.

Finally, a slight latency decrease can be observed in the cloud-only case, when the frame frequency is increased from 2 to 3 Hz. This is solely due to the network architecture between the test location and the Amazon AWS Virtual Machine, which seems to perform better when data is exchanged more often.

C. Road tests

The road tests, with two real vehicles and one RSU, have been performed on a straight stretch of road near Scarborough, Maine, USA, allowing us to test the effect of distance on the mmWave links up to 240 m, under real environmental conditions.

We equipped the two vehicles with the needed OBU hardware and a GNSS receiver (with 10 Hz update rate) and set up a fixed RSU, as depicted in Figure 8. The chosen GNSS receiver, equipped with a U-blox chipset, offers a high update rate and precise localization. This ensures that `Edge-V` can depend on accurate positioning data, circumventing the problems of low positioning accuracy that could adversely affect the performance of algorithms like DG-VEIP.

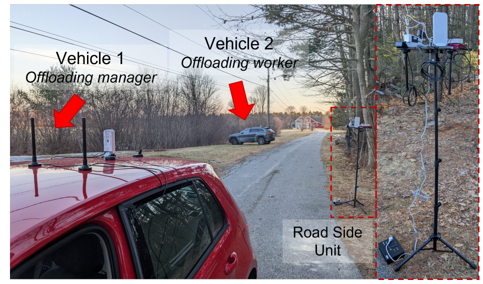

Fig. 8. Road tests experimental setup.

Indeed, testing the effect of different positioning accuracies on the performance of the framework would require many vehicles to get relevant results in the field, and it is thus outside the scope of this paper.

Additionally, we employed one laptop inside each vehicle, connected to the internal IEEE 802.11ac Access Point, to act as on-board device connected to `Edge-V`, similarly to the laboratory test setup.

The primary objective of the road tests was to assess our POC in real-world conditions, where environmental factors are expected to affect communication.

1) Vehicular Data Exchange use case: With the aim of testing the vehicular data exchange use case, as described in Section VI-B1, we developed specialized Python scripts that can output synchronized network performance (i.e., latency and throughput) and relative distance information, thanks to the reception of data from the GNSS receiver. These scripts make use of both the LaTe latency measurement tool [51] and iPerf 3.

One vehicle (*Vehicle 1*) was then moving along the road, and one vehicle (*Vehicle 2*) was parked off the road, while the RSU was placed 32.6 meters in front of *Vehicle 2*. We measured the effect of distance on the capability of `Edge-V` to provide a high-throughput, ultra low-latency communication between two on-board devices. As mentioned earlier, the laptops located in two different vehicles can directly communicate thanks to the IEEE 802.11ac access points located inside each vehicle, and to the mmWave links established between the different nodes (i.e., vehicles and RSU). We focused our analysis on the UDP throughput, as UDP appears to be more suitable than TCP to vehicular scenarios, and on the RTT with a UDP payload of 524 B. Both a direct V2V communication, and communication through the RSU (realizing a V2I2V communication) have been tested. The first scenario (V2V) has been realized by mounting the mmWave AP on *Vehicle 1*, while the second scenario (V2I2V) was configured by moving the mmWave AP from *Vehicle 1* (which was consequently equipped with a client) to the RSU installation. The V2I2V communication reflects now a real relayed communication thanks to the AP being now directly connected to the RSU APU board.

The most significant results are depicted in Figure 9. As can be seen, the obtained throughput and RTT results are in line with the laboratory measurements, considering the addition of two IEEE 802.11ac links between the devices (one inside each vehicle), besides the mmWave communication. The results

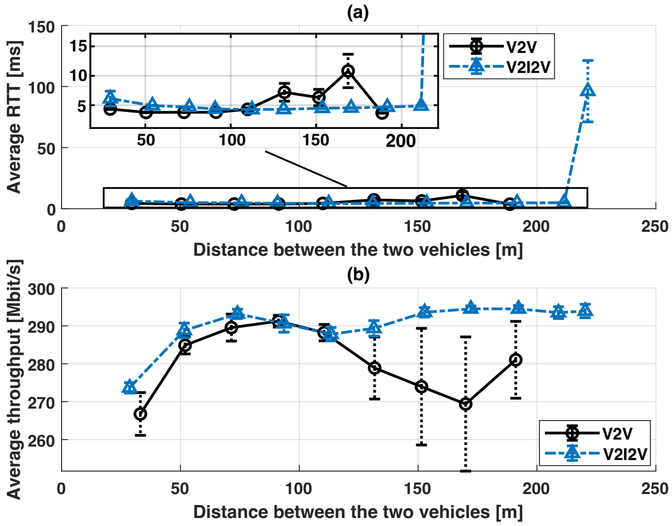

Fig. 9. RTT and throughput as a function of the distance between two vehicles; each point corresponds to a one-minute-long test, with 95% confidence intervals. The plots show: (a) Average RTT between the on-board devices, (b) Average UDP throughput.

show how `Edge-V` can yield a very stable RTT, on average lower than 5 ms, up to a distance of 110 m. Then, the V2V communication starts to experience an increased instability when the distance becomes greater than 130 m, even though the RTT mostly remains lower than 10 ms. These values are very good, as they are compatible with the requirements of even the most demanding automotive applications. Indeed, ETSI defines a maximum end-to-end latency of 50 ms for safety applications [13], [52], [53], which was reduced to 10 ms by the 5G-CARMEN project [54] for highly automated centralized maneuver management.

The relayed V2I2V communication, through the RSU, can instead provide stable values up to around 220 m. Then, 230 m represents a limit distance, since the RSU is placed 32 m ahead of *Vehicle 2*. The RTT below 100 m is overall slightly higher, but the results show how a V2I2V communication can effectively provide an extended range mmWave communication, thanks to the deployment of one or more RSUs.

Similar considerations hold for the measured throughput, which is always above, on average, 270 Mbit/s, thanks to the combination of IEEE 802.11ac and mmWave. Combining these values with the ones measured indoor without the IEEE 802.11ac links (i.e, more than 500 Mbit/s), it is possible to prove how the overlying mmWave network can provide enough bandwidth to accommodate multiple on-board devices on each vehicle.

The results also indicate that a distance between 130 and 150 m can be considered a *safe threshold* for stable and high quality communication. Indeed, the measured values become less stable when the relative distance between the two vehicles, in a direct V2V communication, becomes greater than 130 m. Furthermore, the slight reduction in throughput for relatively short distances is likely due to the angle between the mmWave devices. Indeed, our devices have a limited angular range (i.e., 60 degrees) and, for shorter distances, the effect of slightly different angles in the device placement becomes more

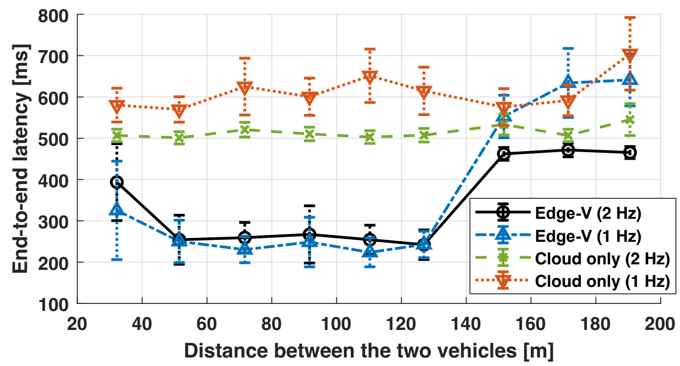

Fig. 10. Average end-to-end object detection latency as a function of vehicle distance and task generation frequency.

evident.

Finally, it should be highlighted that all the RTT and throughput values have been collected while the vehicles are exchanging enhanced CAMs through the 5.9 GHz DSRC link to populate their LDMs. However, both the internal Wi-Fi and mmWave were unaffected thanks to the usage of different unlicensed spectrum bands.

2) Object detection and task offloading use case: Concerning the DL task offloading use case, we considered the same setup as depicted in Figure 8, with two vehicles (i.e., two OBUs) and one RSU. We employed the same object detection service developed for the laboratory tests. As shown, the Object Detection Offloading Manager was deployed on *Vehicle 1*, moving along the road, while the Object Detection Offloading Worker has been launched on *Vehicle 2*, parked off the road.

The most significant results are reported in Figure 10, showing the average end-to-end latency experienced by the Offloading Manager in `Edge-V` as a function of the relative distance between the two vehicles. All the measurements are gathered with *Vehicle 1* offloading frames either to *Vehicle 2* or to the cloud, and compared to a cloud-only offloading baseline, in which frames are offloaded to the cloud via the RSU. As mentioned earlier, the Object Detection Offloading Manager implements a modified and simplified version of DG-VEIP, with a distance limit of 140 m (to guarantee a stable mmWave connectivity, in accordance with the previous road tests) and an RSSI limit of -65 dBm. The last limit comes from the IEEE 802.11ad field test results presented in [18].

As can be seen, the latency reduction with respect to the exclusive usage of cloud is significant, as long as *Vehicle 2* is available and offloading occurs locally. In particular, the maximum latency improvement was observed to be around 65% for the 1 Hz case (at around 130 m) and 52% for the 2 Hz case (at around 110 m). Moreover, it can be noticed how, when the distance is increased over 140 m, the performance decreases since no vehicular edge offloading is performed and the cloud is used. Despite the latency increase, after some tuning of the distance limit, `Edge-V` still continues to provide a reliable service, even when no vehicles are available or under no stable V2V mmWave coverage.

The plot in Figure 10 also shows a slightly lower (up to few tens of milliseconds) average latency when offloading frames at 1 Hz. This can be explained by the fact that the resources on *Vehicle 2* are more often busy when new frames need to be offloaded at 2 Hz, thus causing slightly more frames to be offloaded to the cloud. On the other hand, the difference in the cloud case is due to the network architecture with the Amazon AWS Virtual Machine mentioned earlier, that seems to perform better for higher frequency data exchange. Finally, it is worth mentioning how the same considerations reported for Figure 6 also apply here to explain the slightly higher latency, on average, when the distance between the vehicles is relatively short.

D. Considerations on scalabilty

It should be highlighted how the laboratory and field tests involved a limited number of vehicles and devices due to, on the one hand, the complexity of setting up a laboratory tests with a very large number of devices, and, on the other hand, the unavailability at present time of comprehensive simulation platforms supporting both DSRC and 60 GHz mmWave scenario. However, the results retain significance as they show the performance of `Edge-V` in an easily reproducible scenario, that represents a baseline for deployment also at a larger scale. In addition, we plan to extend the ms-van3t simulator [42] to include Edge-V and mmWave as future work.

When deployed at a much larger scale, for instance considering a city-scale deployment, `Edge-V` can still provide a fairly good performance, thanks to (i) mmWave employing beamforming to reduce interference that would more significantly affect an omnidirectional V2X communication, and that can potentially handle a few hundreds of devices with proper device selection and beamforming optimization algorithms [55], combined with narrow mmWave beams [4], (ii) DSRC reacting to an increased channel load with techniques such as the Decentralized Congestion Control (DCC) [56] and (iii) the larger number of connected vehicles enabling offloading to more nearby nodes, making it less likely to rely on the infrastructure, that would incur an increased communication latency.

VII. Concluding remarks

We have proposed `Edge-V`, a framework exploiting the combination of unlicensed spectrum technologies, together with an embedded smart offloading service, to provide full-fledged Vehicular Edge Intelligence (VEI). We have presented a detailed description of `Edge-V`, modeled the Vehicular Edge Intelligence Problem (VEIP) and demonstrated that it is NP-Hard. We have developed a distributed greedy algorithm (DG-VEIP) to efficiently solve the VEIP, and have evaluated its performance on an open vehicular dataset [13]. We have developed a proof-of-concept testbed for `Edge-V`, based on open-source software and low-cost customizable hardware. Our road and laboratory tests proved how `Edge-V` can provide very low latency and high throughput. We have also shown how local distributed offloading with `Edge-V` can outclass cloud-based approaches on a real-life object detection system. Finally, we hope our paper will inform ongoing standardization efforts, both for the next generation of Wi-Fi-based V2X technologies and in the definition of the new versions of CAMs, thanks to the openness of both `Edge-V` and the POC we developed.

Acknowledgments

This study was carried out within the MOST – Sustainable Mobility National Research Center and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 – D.D. 1033 17/06/2022, CN00000023).

References

[1] T. Shimizu, V. Va, G. Bansal *et al.*, "Millimeter Wave V2X Communications: Use Cases and Design Considerations of Beam Management," in *2018 Asia-Pacific Microwave Conference (APMC)*. IEEE, 2018, pp. 183–185.

[2] M. Kutila, P. Pyykonen, Q. Huang *et al.*, "C-V2X Supported Automated Driving," in *Proceedings of IEEE ICC Workshops*. IEEE, 2019, pp. 1–5.

[3] D. Maggiorini, C. Quadri, and L. A. Ripamonti, "Opportunistic Mobile Games Using Public Transportation Systems: a Deployability Study," *Multimedia systems*, vol. 20, no. 5, pp. 545–562, 2014.

[4] J. Choi, V. Va, N. Gonzalez-Prelcic *et al.*, "Millimeter-wave Vehicular Communication to Support Massive Automotive Sensing," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, 2016.

[5] M. Drago, T. Zugno, M. Polese *et al.*, "MilliCar: An ns-3 module for mmWave NR V2X networks," in *Proceedings of the 2020 Workshop on ns-3*, 2020, pp. 9–16.

[6] S. A. A. Shah, E. Ahmed, M. Imran *et al.*, "5G for Vehicular Communications," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 111–117, 2018.

[7] 5G Americas, "5G: The Future of IoT," 5G Americas, Bellevue, Washington, Tech. Rep., July 2019.

[8] P. Dai, K. Hu, X. Wu *et al.*, "Asynchronous Deep Reinforcement Learning for Data-Driven Task Offloading in MEC-Empowered Vehicular Networks," in *IEEE INFOCOM 2021*, 2021.

[9] Z. Li, T. Yu, R. Fukatsu *et al.*, "Towards Safe Automated Driving: Design of Software-Defined Dynamic MmWave V2X Networks and PoC Implementation," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 78–93, 2021.

[10] A. Molina-Galan, B. Coll-Perales, and J. Gozalvez, "C-V2X Assisted mmWave V2V Scheduling," in *2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS)*, 2019, pp. 1–5.

[11] M. Polese, M. Giordani, T. Zugno *et al.*, "Integrated Access and Backhaul in 5G mmWave Networks: Potential and Challenges," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 62–68, 2020.

[12] F. Raviglione, C. Casetti, and F. Restuccia, "Edge-V: Enabling Vehicular Edge Intelligence in Unlicensed Spectrum Bands," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–5.

[13] F. Raviglione, S. Zocca, A. Minetto *et al.*, "From collaborative awareness to collaborative information enhancement in vehicular networks," *Vehicular Communications*, vol. 36, p. 100497, 2022.

[14] R. Molina-Masegosa, J. Gozalvez, and M. Sepulcre, "Configuration of the C-V2X Mode 4 Sidelink PC5 Interface for Vehicular Communication," in *2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, 2018, pp. 43–48.

[15] L. Liang, H. Ye, and G. Y. Li, "Toward Intelligent Vehicular Networks: A Machine Learning Framework," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 124–135, 2018.

[16] W. Miao, G. Min, X. Zhang *et al.*, "Performance Modelling and Quantitative Analysis of Vehicular Edge Computing with Bursty Task Arrivals," *IEEE Transactions on Mobile Computing*, 2021.

[17] T. Nitsche, C. Cordeiro, A. B. Flores *et al.*, "IEEE 802.11 ad: Directional 60 GHz Communication for Multi-Gigabit-per-Second Wi-Fi," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 132–141, 2014.

[18] F. Raviglione, M. Malinverno, S. Feraco *et al.*, "Experimental Assessment of IEEE 802.11-Based V2I Communications," in *18th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*. ACM, 2021, p. 33–40.

[19] S. Wang, J. Huang, and X. Zhang, "Demystifying Millimeter-Wave V2X: Towards Robust and Efficient Directional Connectivity under High Mobility," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. ACM, 2020.
[20] J. Senic, A. Bhardwaj, C. Gentile *et al.*, "Challenges for 5g and beyond," in *2022 16th European Conference on Antennas and Propagation (EuCAP)*, 2022, pp. 1–5.
[21] S. Mumtaz, J. M. Jornet, J. Aulin *et al.*, "Terahertz Communication for Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, 2017.
[22] M. Giordani, A. Zanella, T. Higuchi *et al.*, "On the Feasibility of Integrating mmWave and IEEE 802.11p for V2V Communications," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–7.
[23] B.-J. Qiu, C.-Y. Hsieh, J.-C. Chen *et al.*, "DCOA: Double-Check Offloading Algorithm to Road-Side Unit and Vehicular Micro-Cloud in 5G Networks," in *GLOBECOM 2020*, 2020, pp. 1–6.
[24] E. Krijestorac, A. Memedi, T. Higuchi *et al.*, "Hybrid Vehicular and Cloud Distributed Computing: A Case for Cooperative Perception," in *GLOBECOM 2020*, 2020, pp. 1–6.
[25] C. Tang, C. Zhu, X. Wei *et al.*, "Task Caching in Vehicular Edge Computing," in *IEEE INFOCOM Workshops 2021*, 2021, pp. 1–6.
[26] T. Higuchi, S. Ucar, and O. Altintas, "Offloading tasks to vehicular virtual edge servers," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, 2019, pp. 162–163.
[27] I. Royuela, J. C. Aguado, I. de Miguel *et al.*, "A testbed for ccam services supported by edge computing, and use case of computation offloading," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 2022, pp. 1–6.
[28] B.-J. Qiu, C.-Y. Hsieh, J.-C. Chen *et al.*, "Tcoa: Triple-check offloading algorithm for roadside units and vehicular microclouds in 5g networks and beyond," *IEEE Access*, vol. 11, pp. 84 985–85 001, 2023.
[29] Q. Wu, H. Ge, Q. Fan *et al.*, "Efficient task offloading for 802.11p-based cloud-aware mobile fog computing system in vehicular networks," *Wireless Communications and Mobile Computing*, vol. 2020, no. 1, p. 8816090, 2020. doi: https://onlinelibrary.wiley.com/doi/abs/10.1155/2020/8816090
[30] M. S. Bute, P. Fan, G. Liu *et al.*, "A cluster-based cooperative computation offloading scheme for c-v2x networks," *Ad Hoc Networks*, vol. 132, p. 102862, 2022. doi: https://www.sciencedirect.com/science/article/pii/S1570870522000622
[31] ETSI, "ETSI EN 302 637-2 V1.4.1 (2019-04)," European Telecommunications Standards Institute, Standard, 2019.
[32] ——, "ETSI EN 302 636-4-1 V1.4.1 (2020-01)," European Telecommunications Standards Institute, Standard ETSI EN 302 636-4-1, 2020.
[33] Z. Li, Y. Guo, and X. Ge, "Performance analysis of urban mmwave multi-hop v2v communications with shifted-exponential distribution headway," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2019, pp. 1–6.
[34] ETSI, "ETSI EN 302 895 V1.1.1 (2014-09)," European Telecommunications Standards Institute, Standard, 2014.
[35] ——, "ETSI TS 103 301 V2.1.1 (2021-03)," European Telecommunications Standards Institute, Standard ETSI TS 103 301 V2.1.1, 2021.
[36] Y. He, D. Zhai, R. Zhang *et al.*, "A Mobile Edge Computing Framework for Task Offloading and Resource Allocation in UAV-assisted VANETs," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–6.
[37] PC Engines. (2025) PC Engines apu2 system boards [online]. https://www.pcengines.ch/apu2.htm.
[38] NVIDIA. (2025) NVIDIA Jetson Nano [online]. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/#:~:text=Other%20I%2FO%203x%20UART%2C%202x,DIMM%20connector.
[39] MiktroTik. (2025) MiktroTik Routers and Wireless - Products: wAP 60G [online]. https://mikrotik.com/product/wap_60g.
[40] S. Burer and A. Saxena, "The MILP Road to MIQCP," in *Mixed Integer Nonlinear Programming*, J. Lee and S. Leyffer, Eds. New York, NY: Springer New York, 2012, pp. 373–405.
[41] V. V. Vazirani, *Knapsack*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 68–73. doi: https://doi.org/10.1007/978-3-662-04565-7_8
[42] F. Raviglione, C. R. Carletti, M. Malinverno *et al.*, "ms-van3t: An integrated multi-stack framework for virtual validation of v2x communication and services," *Computer Communications*, vol. 217, pp. 70–86, 2024. doi: https://www.sciencedirect.com/science/article/pii/S0140366424000227
[43] F. Raviglione. (2021) OpenWrt-V2X [online]. https://github.com/francescoraves483/OpenWrt-V2X.
[44] F. Raviglione, M. Malinverno, and C. Casetti, "Characterization and Performance Evaluation of IEEE 802.11p NICs," in *1st ACM MobiHoc Workshop on Technologies, MOdels, and Protocols for Cooperative Connected Cars*. ACM, 2019, p. 13–18.
[45] A. Howard, M. Sandler, G. Chu *et al.*, "Searching for MobileNetV3," *CoRR*, vol. abs/1905.02244, 2019. doi: http://arxiv.org/abs/1905.02244
[46] L. Walkin. (2021) The ASN.1 Compiler [online]. https://github.com/vlm/asn1c.
[47] ETSI, "ETSI EN 302 636-5-1 V2.2.1 (2019-05)," European Telecommunications Standards Institute, Standard, 2019.
[48] F. Raviglione. (2022) LTNT (Long Term Network Tester) [online]. https://github.com/francescoraves483/LTNT.
[49] T.-Y. Lin et al., "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014, pp. 740–755.
[50] Z. Ge, S. Liu, F. Wang *et al.*, "YOLOX: Exceeding YOLO Series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.
[51] F. Raviglione, M. Malinverno, and C. Casetti, "A Flexible, Protocol-Agnostic Latency Measurement Platform," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–5.
[52] ETSI, "ETSI TS 101 539-2 V1.1.1 (2018-06)," European Telecommunications Standards Institute, Standard, 2018.
[53] ——, "ETSI TS 101 539-2 V1.1.1 (2013-11)," European Telecommunications Standards Institute, Standard, 2013.
[54] 5G-CARMEN. (2022) Home page - 5G-CARMEN [online]. https://5gcarmen.eu/.
[55] T. H. L. Dinh, M. Kaneko, and K. Fujii, "Device selection and beamforming optimization in large-scale mmwave iot networks," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 395–25 408, 2022.
[56] ETSI, "ETSI TS 102 687 V1.2.1 (2018-04)," European Telecommunications Standards Institute, Standard, 2019.

Francesco Raviglione is an assistant professor with time contract at the Department of Electronics and Telecommunications in Politecnico di Torino. He then got a Ph.D. cum laude in Electronics and Telecommunication Engineering in Politecnico di Torino in 2022, with a thesis titled "Open Platforms for Connected Vehicles". His research mainly focuses on services, applications and technologies for connected vehicles, with an emphasis on open hardware and software solutions.

Claudio Casetti is a full professor at the Department of Control and Computer Engineering, Politecnico di Torino. He has published more than 200 papers in peer-refereed international journals and conferences on the following topics: vehicular networks, 5G networks, transport and network protocols in wired networks, and IEEE 802.11 WLAN.

Francesco Restuccia [M'16, SM'21] is an Assistant Professor in the Department of Electrical and Computer Engineering at Northeastern University. He received his Ph.D. in Computer Science from Missouri University of Science and Technology in 2016, and his B.S. and M.S. in Computer Engineering with highest honors from the University of Pisa, Italy in 2009 and 2011, respectively. His research interests lie in the design and experimental evaluation of next-generation edge-assisted data-driven mobile systems.