DRIVE-B5G: A Flexible and Scalable Platform Testbed for B5G-V2X Networks

Taki Eddine Toufik Djaidja*, Bouziane Brik*, Abdelwahab Boualouache†, Sidi Mohammed Senouci* and Yacine Ghamri-Doudane‡
* *DRIVE Laboratory* EA1859, Univ. Bourgogne Franche Comté, France;
{taki-eddine.djaidja,bouziane.brik,sidi-mohammed.senouci}@u-bourgogne.fr
†*FSTM*, University of Luxembourg, Luxembourg; abdelwahab.boualouache@uni.lu
‡*L3I Laboratory*, Univ. La Rochelle, France; yacine.ghamri@univ-lr.fr

***Abstract*—Unlike previous mobile networks, 5G and beyond (B5G) networks are expected to be the key enabler of various vertical industries such as eHealth, intelligent transportation, and Industrial IoT verticals. To support that, B5G networks enable to sharing of common physical resources (radio, computation, network) among different tenants, thanks to network slicing concept and network softwarization technologies, including Software Defined Networking (SDN) and Network Function Virtualization (NFV). Therefore, new research challenges related to B5G networks have emerged, such as resources management and orchestration, service chaining, security, and QoS management. However, there is a lack of a realistic platform enabling researchers to design and validate their solutions effectively, since B5G networks are still in their early stages. In this paper, we first discuss the different methods for deploying realistic B5G platforms for the V2X vertical, including the key B5G technologies. Then, we describe DRIVE-B5G, a novel platform that serves as an end-to-end test-bed to emulate a vehicular network environment, allowing researchers to provide proof of concept, validate, and evaluate their research approaches.**

***Index Terms*—B5G, B5G-V2X, network slicing, SDN, NFV, Test-bed.**

I. Introduction

5G and beyond networks (B5G) are growing to support various applications related to multiple vertical industries such as eHealth, intelligent transportation, and Industrial IoT verticals [1]. One of the most promising 5G applications is vehicle-to-everything (V2X) communication, which is considered a key enabler for autonomous and connected driving. V2X communication-related applications are also expected to revolutionize and shape future intelligent transportation systems. Indeed, both academia and industry have increased their research on V2X-based networks, notably to include different types of communications between vehicles and vehicles (V2V), Infrastructure (V2I), Pedestrians (V2P), and cloud/network applications (V2N), under the framework of cellular vehicle to everything (C-V2X) [2].

V2X-based networks have enabled the emergence of new use-cases related to road safety, driving experience, autonomous driving, traffic efficiency, and environmental friendliness [2]. However, these new use cases differ from traditional ones because of their heterogeneous requirements, including high bandwidth, low access latency, communication reliability, the support of massive numbers of vehicles, etc. Cos of their revolutionary technology, B5G systems are considered one of the key systems to support such requirement heterogeneity. B5G systems are intended to provide highly adaptable and programmable end-to-end communication, networking, and computing infrastructure. Thanks to the network slicing, B5G systems can run several logical networks on the top of the same physical infrastructure, where each logical network meets the requirement of a particular application. In other words, B5G, through the network slicing concept, consists of logically isolating network functions and resources on the top of a single customised network that is tailored to the needs of the service [1]. In this context, three main services may be provided via network slicing: enhanced mobile broadband (eMBB) for services requiring high data rates, massive machine-type communications (mMTC) for services supporting a large number of connected devices, ultra-reliable and low latency communications (uRLLC) for services having stringent latency and reliability requirements [3]. Moreover, the main enablers for network slicing are Network Function Virtualisation (NFV), Software-Defined Networking (SDN), and orchestration. NFV allows virtualizing/containerizing network functions onto light-weighted Virtual Network Functions (VNFs), that can be instantiated at the cloud platform level. At the same time, the key role of the SDN is to ensure, throughout its controller, VNF chaining paths and responding to network outages. Finally, the orchestrator is in charge of the cycle management of the slice's VNFs; this includes instantiating, scaling, and service placement in the cloud or the Multi-access Edge Computing (MEC) nodes [4].

Besides, the 3rd generation partnership project (3GPP) standard has recently introduced 5G-V2X in its release 17 [5]. It describes the main enabled applications along with their requirements as well as enabled communication technologies, such as the PC5 interface for direct communications (V2V) and Uu interface for V2N-based applications. We note that most V2X applications are uRLLC such as safety-related applications and remote driving. However, there are also eMBB applications like 3D map exchanging and infotainment (music streaming), and mMTC applications such as road traffic monitoring are also supported.

Furthermore, new research challenges related to B5G-V2X networks have emerged, such as resources management

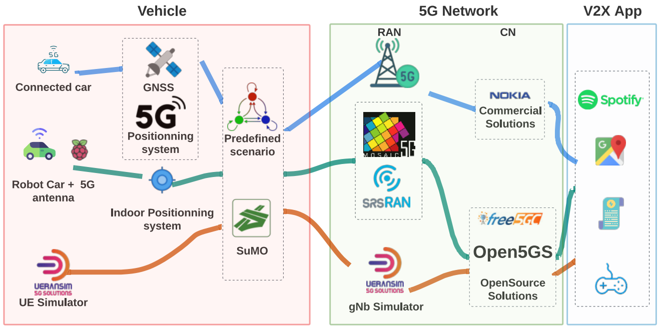

Fig. 1. Possible Solutions of B5G-V2X implementations.

and orchestration, service chaining, security, and QoS management, and researchers are investigating new concepts and approaches, particularly those using Artificial Intelligence/Machine Learning techniques to optimize, enhance and secure the management of such networks. However, there is a great lack of realistic platforms enabling researchers to design and validate their solutions effectively, since B5G-V2X networks are still in their early stages. In this paper, we first discuss the different methods for deploying realistic B5G platforms for the V2X vertical, including the key B5G technologies. Then, we describe a novel and unique platform named DRIVE-B5G as an E2E test-bed to emulate a vehicular network environment that involves different parts, ranging from vehicle nodes, radio access network (RAN), Core Network (CN), services to the Management and Orchestration (MO) as well as edge computing.

The rest of the paper is organized as follows. Section II describes the main building blocks of the B5G-V2X stack. Section III lists the potential options for deploying a 5BG-V2X test-bed. Section IV describes the architecture of the proposed platform DRIVE-5G, as well as the different building blocks: vehicles simulation, RAN, CN, and Vehicular applications. Section V presents the performance results on latency and CPU use in various deployment situations. Finally, section VI discusses the work and brings the study to a close.

II. Background on B5G-V2X stack

In this section, we describe the different entities involved in the B5G-V2X ecosystem, including the vehicles, the B5G network, and the V2X applications. Moreover, we discuss the various solutions available to implement each entity on the testbed.

A. Mobile Nodes: Vehicles

Connected and Automated Vehicles (CAVs) are designed to support two network interfaces, PC5 for direct communications between vehicles (V2V) and Uu for V2N communications through the Internet [6]. We refer to CAV as simply vehicles.

Usually, to design an effective B5G-V2X testbed, CAV can be considered in three different ways: (i) The first and most realistic choice/method is to use real vehicles. In fact, new vehicles are equipped with network interfaces, embedded software like Advanced driver-assistance systems (ADAS) or autopilot, in addition to different sensors including radars (long and short-range), cameras, lidar, ultrasound, etc. Also, both the Global Navigation Satellite System (GNSS) and 5G may be used as positioning systems as defined in [7]. However, the 5G positioning system tends to be more precise [8]. (ii) The second option is to use robot cars equipped with a RaspberryPi [1], and 5G antennas to enable V2N communications. A certain number of sensors can also be integrated into such robots, depending on the targeted scenario. It is also required to have an indoor positioning system, as it was designed/discussed in [9], and (iii) The last option consists to simulate vehicles by using 5G User Equipment (UE) simulators; where the radio interface between UE and RAN is simulated through IP communications. This last solution is less realistic due to the lack of radio transmissions in this environment. Nonetheless, the last solution is more efficient in testing the other components of the B5G-V2X stack, with a high number of UEs (scalability).

Regardless of the chosen method, to perform an accurate simulation, a mobility pattern should be devised to determine how vehicles move. We usually distinguish two options to generate vehicles mobility models: (i) SUMO[2]: stands for Simulator for Urban Mobility. It enables generating realistic traffic traces; the output of SUMO could be used as a traffic pattern, and (ii) Another option would be to design a Markov model to define the mobility pattern, and the way cars move between regions. This approach is demonstrated in equation 1 for a situation with three road regions, where π_0 represents the initial distribution and P the transition matrix. Moreover, further investigation into the stationary distribution of the model provides insights into vehicle density and the type of the road region (highway, rural, etc.).

$$\pi_0 = (\frac{2}{3}, \frac{1}{6}, \frac{1}{6}); P = \begin{pmatrix} 1/3 & 2/3 & 0 \\ 0 & 1/2 & 1/2 \\ 1/2 & 1/2 & 0 \end{pmatrix} \quad (1)$$

B. B5G Network

New paradigms have been introduced in B5G networks, such as network slicing, SDN, and NFV, all contributed to making this network generation revolutionary and connecting different verticals, such as the automotive vertical. A B5G network is mainly composed of two components: RAN and CN (as shown in Figure 1). RAN is the part of the system that links devices to the network via radio transmissions. RAN can be realized in outdoor or indoor environments. The former, by use of commercial base stations (gNBs) solutions, differ depending on the scale, coverage, and technologies implemented, such as Multiple Input Multiple Output (MIMO) and RAN slicing. The latter can be implemented by the use of Software Defined Radio (SDR) solutions like srsRAN or mosaic5G[3], in addition to Universal Software Radio Peripheral

[1] raspberrypi.org
[2] eclipse.org/sumo
[3] srslte.com; mosaic5g.io

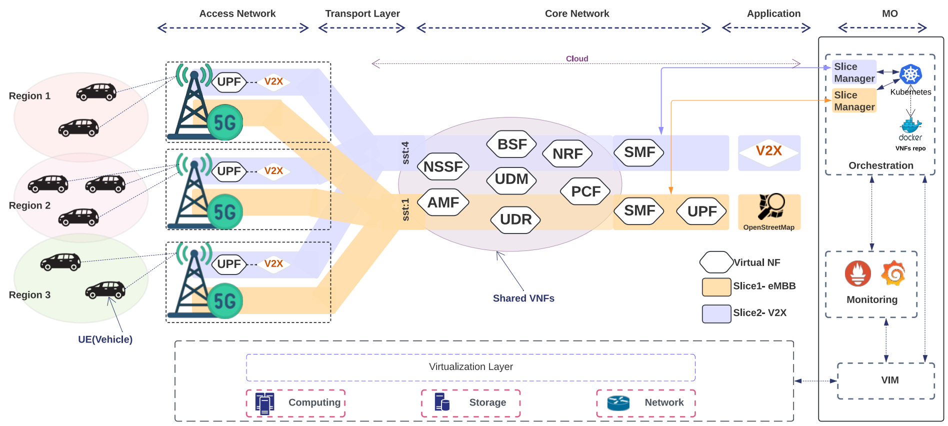

Fig. 2. Overview of our DRIVE-B5G Platform

(USRP) equipment. The SDR solutions provide a realistic and up-to-date environment but in a small coverage area, which makes it more suitable for in-lab emulations. Besides, there is also the option of simulating the radio interface, using a simulator in the same way as UEs. It is important to note that the control and user interfaces should operate normally. This type of solutions are developing and expanding as they are currently able to support network slicing.

Another critical component of the B5G network is the core network (CN). It is made up of a set of VNFs, that are generally deployed in containers. We can list two main types of VNFs: control and user planes. The VNFs communicate through HTTP protocol, and their exposed APIs (VNFs). The core network in B5G is sliced to tailor to the needs of the provided services, while the SDN controller performs communication and chaining between the different services. Furthermore, the core network also comprises an orchestration and management module for running network slices. This module also has many other functions, like managing the multi-tenancy, quality of service (QoS), service placement (in the cloud or edge), service scaling, and the security of the network slices. For the implementation of B5G CN, both commercial and open-source projects are available. As examples of the open-source project, there are the recent Open5Gs and free5GC[4].

C. V2X Apps

Vehicular applications were defined along with the 3GPP standards; Release14 and Release15 introduced basic safety and non-safety applications, while Release16 and Release17 specified advanced and non-safety applications with strict performance requirements. Indeed, vehicle applications are divided into two categories: advanced driving-related apps and infotainment apps [11]. The former aims to ensure safety, as exampled by cooperative driving, maps exchanging, platooning, and remote driving. The latter aims to improve the driving experience by allowing users to listen to music, play video games, or watch videos while moving. Each vehicle application has its proper performance requirements, known as Key Performance Indicators (KPI). Therefore, each service is characterized by its category (eMBB, uRLLC, and mMTC), and appears as a slice in the B5G network.

A variety of applications must be supported in order to establish a realistic B5G-V2X testbed. Fortunately, many of them are open source and may be used, such as OpenStreetMap[5] for map exchanging and Ampache[6] for music and video streaming. Alternatively, an application might be developed locally to simulate a test situation, such as overtaking, lane merging, and splitting.

III. B5G-V2X Testbed Deployment Options

When the above-mentioned B5G-V2X stack solutions are chained together to form a B5G-V2X test-bed, two deployments are possible: outdoor and in-lab deployment. The former uses commercial solutions, while the latter may be performed either through emulating or simulating the radio interface. Figure 1 summarises these deployment possibilities.

a) Outdoor deployment: depicted by the blue line in Figure 1, experiments are carried out at trial sites (such as Transpolis[7]). In general, trail sites are equipped with all B5G-V2X stack components. CAVs have all network interfaces and various sensors, while the B5G network is managed by a teleco operator. Hence, commercial RAN and CN are used. In addition, CAVs move in the trail site circuit, following a predefined mobility model, and may use GNSS or 5G as a positioning system.

However, this approach is costly and its scalability is limited. In addition, the B5G network part is generally used as a black box, since it is provided by a teleco operator, resulting in a limit on allowed performed actions.

b) In-lab deployment: It is about emulating or simulating all or part of the C-V2X components, while preserving as

[4] open5gs.org; free5gc.org

[5] openstreetmap.org

[6] ampache.org

[7] transpolis.fr

TABLE I
COMPARISON BETWEEN B5G TESTBED DEPLOYMENT OPTIONS

	Outdoor env	In-Lab	
		Option A	Option B
Cost	High	Medium	Low
Realistic	+++++	+++	++
Sensors supporting	+++++	++	/
Scalability	+	++	+++++

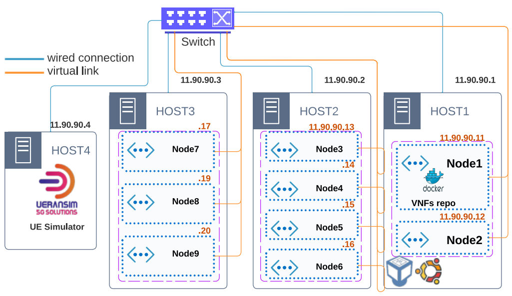

Fig. 3. Hardware and Virtualized Infrastructure.

many realistic aspects as possible and keeping the costs under control. Below is a detailed explanation of these two options:

- Option A (emulation): the green line in Figure 1. Vehicles are imitated using miniature four-wheeled robots, equipped with 5G antennas. They may move in a built-in circuit following a specified moving scenario and must have an indoor positioning system. For the RAN, software defined radios, a USRP and srsRAN can used. This approach might accommodate tens of vehicles as well as many USRP-based gNBs.
- Option B (simulation): the orange line in Figure 1. UERANSIM[8] is used to simulate vehicles and RAN. For mobility, both SUMO and predefined scenarios are supported. One downside is that wireless Radio Interface is mimicked via wired IP connections, which reduces the realism. Meanwhile, this choice appears to be the ideal for conducting research on the CN and V2X app parts, as we can add tens of Vehicles and gNBs.

In regards of vehicular applications, there are a plethora of open-source projects available, as well as prototype applications that may be developed locally. In addition, standards and specifications can be found in the different 3GPP releases. Table I compares the various test-bed deployment options.

IV. DRIVE-B5G ARCHITECTURE OVERVIEW

In this section, we describe the main building blocks of the proposed DRIVE-B5G platform. Figure 2 depicts the DRIVE-B5G testbed. Noting that we chose an In-lab solution, where the radio interface is simulated.

A. Hardware and Virtualized Infrastructure

This building block stands for Virtualisation and Infrastructure management (VIM) and is in charge of managing the hardware resources, and the virtualized infrastructure. Our testbed is made up of 4 compute nodes ($HOST_i, i \in [1, 4]$), as depicted in Figure 3. On each host, Ubuntu-server is installed, as well as VirtualBox as a virtualisation solution. A local network is formed by connecting hosts to a switch. Nine virtual machines are formed in the first three hosts (node_j $j \in [1, 9]$), these nodes use a virtual bridged network interface, allowing them to be part of same local network. The different testbed parts are intended to be deployed on these VMs. While, UEs are intended to run on Host_4.
Given that RAN and MEC will be installed in nodes running on Host_3, which are expected to be close to users in Host_4; We established an intentional network delay of 10ms between nodes in Host_3 and those in Host_1 and Host_2. The reason is to imitate the delay caused by distance, this delay corresponds to a distance of 650 kilometres (Paris-Marseilles).

B. 5G Network

a) RAN Part: We used UERANSIM gNB for the RAN building block, and we assumed the presence of three regions, each covered by one gNB. Therefore, we built conternized instances of the gNB simulator, which are deployed in nodes 7, 8, and 9. Aside from the simulated radio interface, gNB has two more interfaces: a control interface (with AMF), via the N2 reference interface, and a user interface (with UPF), via the N3 reference interface. Furthermore, in order to realize MEC in our testbed, VNFs may be placed in the aforementioned nodes offering shorten routing path.

b) CN Part: We opted for Open5Gs, a C-language open-source implementation of 5G Core Network, which is complied with the 3GPP release 16. Compared to other projects, Open5Gs has a meaningful advantage of supporting network slicing which motivate our choice. Open5Gs is considered as a complete CN, implementing the set of VNFs introduced in 3GPP Rel16. Each VNF is virtualized in a Docker container. The implemented functions include control plane functions, such as AMF (Access & Mobility Management Function), SMF (Session Management Function), NSSF (Network Slice Selection Function), and data/user plane functions (UPF).
AMF is responsible of the management of end device registration, connection, and mobility. SMF is primarily in charge of creating, updating, and deleting user plane connectivity sessions aka Protocol Data Unit sessions. NSSF is the key network function implemented in Open5Gs. NSSF assists the selection of network slices and the associated network function that will serve a device. The user plane function (UPF) is the gateway that connects a 5G end device (such as a vehicle) to the data network (application part). In addition, traffic redirecting rules are indeed being updated, in order to serve the user with the shortest routing path.

C. MO Part

The Management and Orchestration (MO) building block is composed of a set of automation tools, whose primary function is to manage the life-cycle of network slices.

[8]github.com/aligungr/UERANSIM

For our testbed, we formed a cluster of eight nodes (Node_2-9); and we chose Kubernetes as an orchestration solution; the Kubernetes controller is located on Node_2. Moreover, we used Calico[9] as an SDN controller, where its role is to create a virtual local network between the different containers located in the cluster's nodes, and hence ensure network function chaining.

As previously mentioned, network functions are conternized. Thus, we deployed a private Docker repository to allow the orchestrator to obtain these containers. The repository is hosted on Node_1 and contains the images of RAN and CN VNFs as well as V2X application images. The orchestrator exposes an API that is used by the different slices managers. The slice manager offers another API that the slice owner consumes. In addition, the slice owner communicates the blueprint of its slice to the slice manager via HTTP requests. The blueprint comprises the slice metadata as well as the network requirements. The slice manager will then initiate the various VNFs and decide on their placement and replication policy. Furthermore, Prometheus and Grafana[10], are deployed in the cluster to monitor the slices resources. The main role of Prometheus is to collect different metrics about CPU, memory, disk, and network usage from the different resources: cluster, node, and VNFs. This data is gathered through SQL-based requests. Besides, Grafana is a graphic utility that allows the creation of convivial dashboards in an ergonomic fashion, we used it to display the data collected by Prometheus to the slice owner. We are always introducing new features and automation tools to the MO part of our testbed.

D. V2X Slices

As stated before, 5G networks are service-oriented, with services deployed as end-to-end (E2E) network slices. The service application is included in an E2E network slice, that spans both the RAN and CN parts. Network slices are tailored to the needs of the application; a 5G slice is characterised by its Slice/Service Type (SST), which might be 1, 2, 3, 4 or 5 for eMBB, uRLLC, MIoT (massive IoT), V2X and HMTC (High-Performance Machine-Type Communications) respectively **??**. Unfortunately, because the RAN component is simulated in our test-bed, RAN slicing is not supported. However, CN and Application are sliced. In the vehicular networking domain, we implement two V2X apps with diverse requirements:

a) Slice1- Maps Exchanging: One of the main function of connected vehicles is to assist the driver. Hence, the need to display maps continuously is primordial. For this, we set up a dedicated slice for vehicles to access maps' data; Vehicles are intended to periodically gather maps while moving.
Except for SMF and UPF, which are exclusive to the slice, CN shares control plane VNFs with Slice 2. The slice has $SST = 1$, the service consumes the OpenStreetMaps API.

b) Slice2- CAM V2X: The exchange of cooperative awareness messages (CAM) between surrounding vehicles is an important application in vehicular networks. CAMs are communicated directly (V2V, Vehicle to Vehicle) or via 5G networks (V2N2V, Vehicle to Network to Vehicle). We created an UDP-based V2N2V CAM exchanging application, that receives CAM messages and broadcasts them to the neighbouring vehicle. Vehicles are designed to communicate CAMs at a high frequency and as quick as possible. Consequently, the SST value of this slice is 4.

To meet the above requirements, we created a slice in which the data plane VNF (UPF) and an instance of the app are placed in the MEC as closes as possible to end devices (in nodes 7, 8, and 9) and replicated across all regions. Additionally, Slice 2 has its own SMF, and shares the other control plane VNFs with Slice 1.

E. Vehicles

Vehicles are modeled as multi-agent system; managed by python-based simulator that we also develop in this work. A few specifications are required to start a simulation, such as the simulation period, the number of vehicle agents, and their distribution in the regions are among them. The agents are launched in Host_4, vehicles use UERANSIM to connect to the 5G network, and they are linked to the gNbs operating on nodes 7, 8, and 9.
A termination signal is produced when the simulation timer expires. When this message is received, all of the agent's task threads are ended, and the agent is killed.
The set of tasks that an agent vehicle may perform during the simulation time are:

- **Connect**: The task is performed immediately when the agent is instantiated. As 5G devices are envisioned to have access to several network slices, the connection procedure must ensure a successful attachment to all network slices. Furthermore, if there are failures when attempting to connect, the connection task may be performed many times. If the issue persists(up to max_retries), the agent is terminated. When the vehicle agent succeeds to connect to the two slices (Slice1-MapsExchanging and Slice2-CAM V2X), it starts to perform the other tasks: send CAM, receive CAM, and get maps.
- **Send CAM**: Vehicles are expected to produce CAM messages at a frequency of 10Hz; CAM messages contain kinematic information, including position, velocity, heading, etc. This task is performed every 100ms in a cyclic fashion. In our case, we generated random values for our CAM messages and sent them to the V2X-CAM application via the V2X slice.
- **Receive CAM**: The vehicle agent should keep listening for CAM messages sent by the other vehicles. We store the received messages to measure the various performances (latency, packet delivery ratio, etc.).
- **Get MAPS**: This task is (re)executed by the vehicle, while visiting new locations, in order to collect a map of them. To retrieve the data, the vehicle agent sends an HTTP request to the OpenStreetMaps server over Slice1.

[9]calicolabs.com
[10]prometheus.io; grafana.com

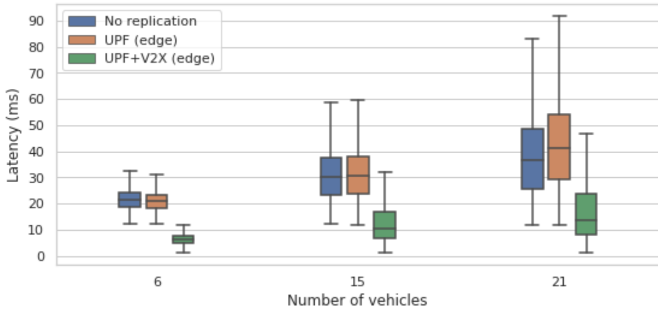

Fig. 4. Generated Latency in DRIVE-B5G Platform.

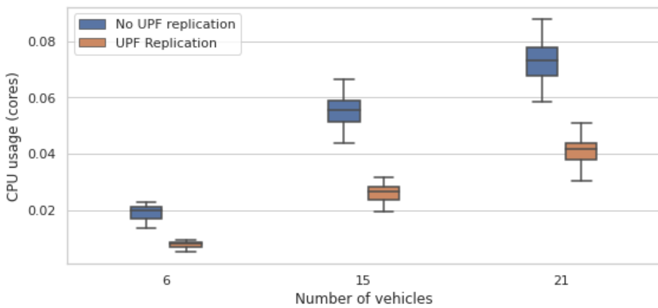

Fig. 5. CPU usage in DRIVE-B5G Platform.

V. Proof of Concept

In order to demonstrate the feasibility of the proposed DRIVE-B5G platform, we conducted simulations with a varied number of vehicles in three different deployment scenarios. In the first one, one instance of UPF and V2X app are deployed as VNFs, at the centralised cloud. In the second scenario, the UPF VNF is replicated and placed at the edge nodes, meanwhile the V2X app is kept at the cloud level. In third scenario, both UPF and V2X application are replicated in the edge nodes. This scenario reproduces the same architecture depicted in Figure 2.

Figure 4 shows the latency in three different testbed implementations: blue, orange, and green; which are respectively the three deployments scenarios presented above. We observe that the latency increases as we increase the number of vehicles (density). Nonetheless, low latency (25ms) is introduced in the third scenario, despite varying the number of vehicles. We note that the first two scenarios (blue and orange), have a slightly higher latency since both UPF and V2X app are deployed at the edge level in scenario 3 which minimises the latency as compared to the cloud.

Figure 5 depicts the CPU consumption in the first two cases (blue and orange). We can see that distributing traffic across UPF divides the CPU usage among each UPF instance, lowering the average CPU load.
These results indicate that our platform is capable of providing a realistic testbed for validating various research ideas such as service placement and replication, resource management.

VI. Conclusion and Perspectives

In this paper, we design a novel In-lab platform named, DRIVE-B5G, as an E2E test-bed to emulate vehicular network environment, that implements the latest introduced technologies and concepts in B5G networks, including network slicing, SDN, and NFV. DRIVE-B5G involves different parts, ranging from vehicle nodes, radio access network, core network, services to the Management and Orchestration as well as edge computing.
DRIVE-B5G platform enables researchers not only to validate their research solutions, but also generate realistic dataset about running network slices and their managements. Indeed, recent architectures are being designed in B5G networks context, such as zero touch management architecture, which requires a heavy usage of machine learning, to auto build the suitable decisions. However, to build efficient learning models, machine learning algorithms need realistic datasets, which may be gathered using our DRIVE-B5G platform.
As a future work, we plan to extend our platform to consider other vertical industries, such as Internet of things and drone networks.

Acknowledgment

This work was supported by the 5G-INSIGHT bilateral project, (ANR-20-CE25-0015-16), funded by the Luxembourg National Research Fund (FNR) and by the French National Research Agency (ANR).

References

[1] Karim Boutiba, Adlen Ksentini, Bouziane Brik, Yacine Challal, and Amar Balla. 2022. NRflex: Enforcing network slicing in 5G New Radio. Comput. Commun. 181, C (Jan 2022), 284–292. https://doi.org/10.1016/j.comcom.2021.09.034
[2] Abdel Hakeem, S.A., Hady, A.A. & Kim, H. 5G-V2X: standardization, architecture, use cases, network-slicing, and edge-computing. Wireless Netw 26, 6015–6041 (2020).
[3] C. Campolo, A. Molinaro, A. Iera and F. Menichella, ”5G Network Slicing for Vehicle-to-Everything Services,” in IEEE Wireless Communications, Dec. 2017, doi: 10.1109/MWC.2017.1600408.
[4] W. Hammedi, B. Brik and S. M. Senouci, ”Toward Optimal MEC-Based Collision Avoidance System for Cooperative Inland Vessels: A Federated Deep Learning Approach,” in IEEE Transactions on Intelligent Transportation Systems, doi:10.1109/TITS.2022.3154158.
[5] “Release 17 Description” 3rd Generation Partnership Project (3GPP), Technical report (TR) 21.917, 2021.
[6] M. H. C. Garcia et al., ”A Tutorial on 5G NR V2X Communications,” in IEEE Communications Surveys & Tutorials, doi: 10.1109/COMST.2021.3057017.
[7] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson and H. Wymeersch, ”Location-Aware Communications for 5G Networks: How location information can improve scalability, latency, and robustness of 5G,” in IEEE Signal Processing Magazine, Nov. 2014, doi: 10.1109/MSP.2014.2332611.
[8] 3rd Generation Partnership Project (3GPP), Technical report (TR) 22.872 V16.1.0, 2018.
[9] Mendoza-Silva, G.M.; Torres-Sospedra, J.; Huerta, J. A Meta-Review of Indoor Positioning Systems. Sensors 2019, 19, 4507. https://doi.org/10.3390/s19204507
[10] ”System architecture for the 5G System (5GS); Stage 2” (3GPP), Technical specification (TR) 23.501, 2022.
[11] Alnasser, Aljawharah & Sun, Hongjian & Jiang, Jing. (2019). Cyber Security Challenges and Solutions for V2X Communications: A Survey.