A testbed for CCAM services supported by edge computing, and use case of computation offloading

Ignacio Royuela
Universidad de Valladolid
Valladolid, Spain

Juan Carlos Aguado
Universidad de Valladolid
Valladolid, Spain
0000-0002-2495-0313

Ignacio de Miguel
Universidad de Valladolid
Valladolid, Spain
0000-0002-1084-1159

Noemí Merayo
Universidad de Valladolid
Valladolid, Spain
0000-0002-6920-0778

Ramón J. Durán Barroso
Universidad de Valladolid
Valladolid, Spain
0000-0003-1423-1646

Diego Hortelano
Universidad de Valladolid
Valladolid, Spain
0000-0001-7776-6862

Lidia Ruiz
Universidad de Valladolid
Valladolid, Spain
0000-0001-6241-5998

Patricia Fernández
Universidad de Valladolid
Valladolid, Spain
0000-0001-5520-0948

Rubén M. Lorenzo
Universidad de Valladolid
Valladolid, Spain
0000-0001-8729-3085

Evaristo J. Abril
Universidad de Valladolid
Valladolid, Spain
0000-0003-4164-2467

***Abstract*—Mobile technologies have undergone a great leap forward in a few years, and while 5G networks are already being deployed, there are not yet many proven applications that can fully utilize the advantages of this new technology. Connected and autonomous vehicles are a specific and demanding case, particularly in terms of delay and bandwidth requirements, which can leverage not only 5G but also edge computing technologies. Therefore, the development of testbeds to demonstrate future applications is crucial to enable the full deployment of 5G and edge computing possibilities. In this paper, we present a flexible and modular testbed, targeted towards the evaluation of Cooperative, Connected, and Automated Mobility (CCAM) applications, and we demonstrate a use case (using a 4G system) where an autonomous vehicle offloads processing tasks to an edge server which analyzes images, makes routing decisions, and sends guidance commands back to the vehicle, thus proving the possibilities of edge computing and wireless technologies.**

Keywords— testbed, 4G, 5G, computation offloading, edge computing, connected vehicle, autonomous vehicle

I. Introduction

Nowadays, 4G networks are completely deployed and the deployment of 5G has just started. At the same time, the requirements of very low latency, high bandwidth, reliability and security of new mobility services force a change in the network architecture, thus leading to a decentralized service platform based on Multi-access Edge Computing (MEC) and/or fog computing, which can bring the necessary resources closer to the user [1]. Although researchers have put enormous effort into defining new architectures and evaluating services through simulation, there is a pressing need to test services in the field to see how different elements of the system contribute to overall performance [2], or at least to test those services in testbeds.

Services related to Cooperative, Connected, and Automated Mobility (CCAM) are a particular case which requires this type of tests. The 5G standard includes CCAM as a specific set of use cases to be tested [3], and a full set of services have been envisioned to be deployed during the next ten years [4]. Many of those services are crucial to get the advantages that the new mobility promises. For example, it has been widely demonstrated that a higher market penetration of automated vehicles will not improve the traffic flow because of their conservatism when negotiating complex driving situations (see for example [5]). One solution is to create services in the infrastructure to help autonomous cars to make these complex, context-aware and coordinated decisions.

Although it is possible to find research results regarding connected vehicles, a complete experimental set-up is only available to a few companies and research groups. Unlike drone-related use cases, for which many experimental setups can be found, there are only a few with cars, as for example [6]. This way, many advantages can arise from the development of experimental testbeds including vehicles and 5G or other wireless technologies.

The contribution of this paper is twofold. We first present the design and building of a testbed to test different services offered to connected and autonomous vehicles. This mock-up includes (or will include) a road circuit, remote-controlled cars, and a network infrastructure including various wireless station technologies, access and core networks. Then, we use the testbed to demonstrate a use case of a connected and autonomous vehicle which offloads image processing and decision-making tasks to an edge server. The edge server, after performing those processing tasks, sends guidance commands back to the vehicle. A companion video to this paper, showing that demonstration, is also provided.

II. Design of a Testbed For CCAM Services

A. Requirements of the testbed

The testbed should be able to test a wide variety of CCAM-related scenarios and services. Therefore, it should meet several features:

- The road circuit of the testbed should be complex and large enough to test realistic scenarios, including metropolitan environments, which means that the road circuit should have at least junctions and crossroads to test crash avoidance and other similar services. Scenarios that require coordination of cars, such as platooning applications, should also be considered.

This work has been supported by Consejería de Educación de la Junta de Castilla y Leon, and the European Regional Development Fund, ERFD (VA231P20 and VA085G19), the Spanish Ministry of Science of Innovation and the State Research Agency (Grant PID2020-112675RB-C42 funded by MCIN/AEI/10.13039/501100011033 and RED2018-102585-T) and the Interreg V-A España-Portugal (POCTEP) program 2014-2020 and ERDF (0667_DISRUPTIVE_2_E). Nevertheless, opinions are the sole responsibility of the authors.

978-1-6654-0601-7/22/$31.00 ©2022 IEEE

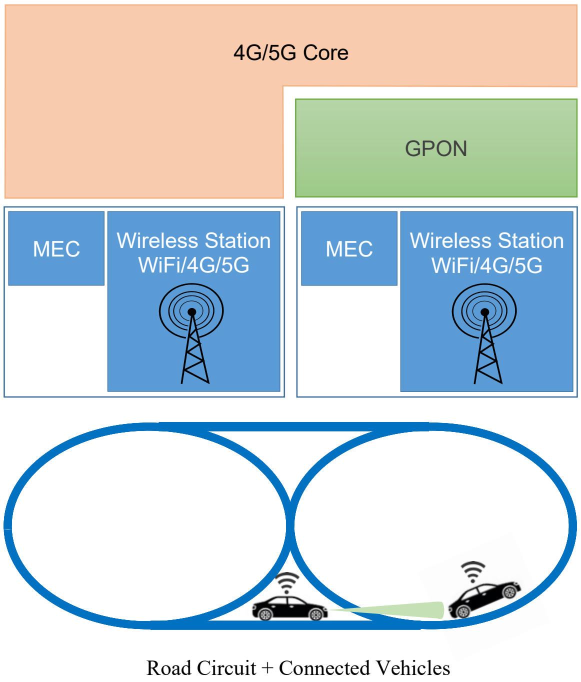

Fig. 1. General view of the testbed design.

- Vehicles (remote-controlled cars) should be able to connect through several communication technologies (or have physical ports to support the installation of suitable modems). They should be able to perceive the environment and perceive themselves (proprioception).
- Since current communication networks are becoming increasingly heterogeneous, the interaction of different technologies should be analyzed. Therefore, the testbed should be prepared to include wireless networks, as well as access networks (including Gigabit Passive Optical Networks, GPON) and core network. Moreover, the Software Defined Networking (SDN) paradigm should also be included for the management of these networks.
- Edge computing should be a central technology in the testbed.

The general view of the testbed, currently under development, is shown in Fig. 1. In the following subsections, we present the design of the circuit road, the selection of the remote-controlled vehicles and the networking infrastructure.

B. Road Circuit

For the construction of the testbed, we have designed a 1:16 scale road circuit. It is based on a series of pieces of different sizes (from 1×1.60 meters to 80×60 centimeters). Thanks to this modular design, once the pieces are built, multiple circuits with different paths can be implemented. Therefore, the circuit allows us to carry out studies on multiple use cases due to its flexibility. The pieces have been designed according to the following conditions:

- The design of the pieces should facilitate the implementation of a simple guidance algorithm to convert a remote-controlled car into an autonomous vehicle. The reason for seeking such simplicity is that the goal of the testbed is not to do research on autonomous vehicle guidance, but on the associated communication and computing services. Therefore, the design of the circuit pieces should facilitate the implementation of that guidance capability, which, although necessary, is not our central objective.
- The pieces should support junctions, crossroads and multiple parallel lanes. Junctions are necessary to test use cases related to decision making capabilities, crossroads allow the evaluation of crash-avoidance and intelligent traffic lights use cases, and multiple parallel lanes allow the analysis of platooning and overtaking use cases, to name just a few.
- The design should be large enough to test handover use cases in wireless networks, but the circuit should also be reconfigurable and easily removable.

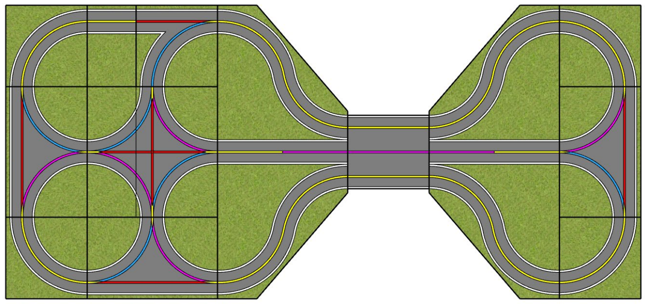

Fig. 2. Example of circuit design. Different colors are used in the central line of the pieces to facilitate the definition of routes to be followed by vehicles (if required).

We have designed nine different types of pieces, including curves, straight roads with one or more lanes, double and triple junctions, and crossings. In this way, complex circuits can be built to support decision-making scenarios for route selection, overtaking and lane changes, among others. Fig. 2 shows an example of a circuit composed by thirteen pieces, which includes the nine designed pieces. To facilitate the definition of routes to be followed by a vehicle (if required), the central line of the pieces may have different colors. Section III will show an example of that functionality.

Fig. 3 shows the different materials composing a piece of the circuit. The road is made of a rubber-based material and has a central line painted on it indicating the trajectory that the vehicle must follow. The line is necessary for the car guidance algorithm (explained in Section III). Outside the road, artificial grass has been laid to avoid the vehicle having a direct vision of the ground of the place where the mock-up is mounted. In this way, the surrounding environment of the roads is always the same (artificial grass) independently of the place where the testbed is located, which simplifies the guidance algorithm. To make the pieces sturdy, portable and removable, the artificial grass and the roads have been

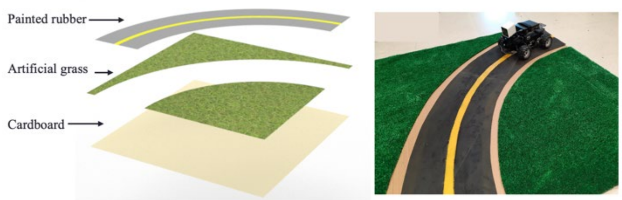

Fig. 3. Assembly of one piece, and picture of a real piece of the road circuit

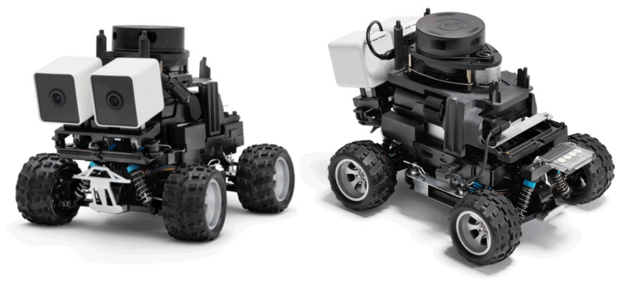

Fig. 4. Front and rear view of Amazon DeepRacer Evo vehicle [7].

mounted on cardboard sheets. Between the grass and the rubber we have left a space of 1.5 cm that emulates the side lines of the road, enabling the implementation of guidance algorithms more similar to those of real autonomous cars if necessary.

C. Connected vehicles

The general conditions to be met by the remote-controlled vehicle have already been described in Section II.A. The chosen scaled vehicle should emulate the sensory system of current autonomous cars and communicate using different technologies. In addition, the vehicle should have a programmable computer capable of controlling the vehicle (acceleration and steering angle) and receiving the information from all the sensors.

We have chosen the Amazon DeepRacer Evo (Fig. 4) [7]. This vehicle is sold with the aim of performing autonomous car races powered by artificial intelligence. It includes a 2D LIDAR with a 12 meter 360-degree scanning radius, 4 MP stereo cameras, 802.11ac Wi-Fi connection, three USB ports, an IMU (Inertial Measurement Unit) with gyroscope and accelerometer, and a computer running Ubuntu 16.04.3 LTS with 4 GB of RAM and a 1.6 GHz Intel Atom processor. The vehicle reaches a maximum speed of approximately 4 m/s and its turning radius is 70 cm.

The advantages of this vehicle are multiple as we obtain in the same package a scale motorized vehicle with LIDAR sensor, cameras and an embedded computer running Linux. However, there are several drawbacks to using it. It does not include any system for speed measuring (although it is possible to hold a constant unknown speed), nor is it equipped with a geopositioning system. In addition, the turning radius of 70 cm at 1:16 scale is equivalent to a radius of 11.2 meters, which is much larger than the turning radius of some utility vehicles (about 5 meters). In fact, the size of the circuit pieces depends mainly on the turning radius of the vehicle.

The vehicle comes with software capable of receiving information from cameras and LIDAR, controlling the vehicle's engines, and also running machine learning algorithms developed through the Amazon AWS platform. All its software is managed through ROS (Robot Operating System). In our case, we have disabled the processes related to machine learning to have greater control of the vehicle for the evaluation of different CCAM-related use cases, leaving only the processes involved in receiving information from sensors and controlling the vehicle.

In terms of communication technologies, the vehicle only comes with Wi-Fi technology. However, it incorporates several USB ports, thereby allowing the connection of 4G and 5G modems. The only communication technology that we have not yet been able to successfully attach is Dedicated Short-Range Communications (DSRC), since, to our knowledge, there is no USB version of devices for 802.11p implementing OCB (Outside the Context of a Basic service set) mode. DSRC technology is used for vehicle-to-vehicle communications, and therefore it is crucial for some Intelligent Transport Systems (ITS) applications. The OCB mode allows to transmit messages in wireless networks avoiding the authentication and association processes, which are the main delay sources in Wi-Fi networks. Thus, although there are several USB Wi-Fi devices which allow working on 802.11p frequencies, they still need to perform the authentication and association processes.

Finally, we have improved the vehicle with a few modifications. We have included a multiport USB hub, attached a bigger battery to improve the autonomy, and set stronger springs in the wheels to support the weight of the new battery.

D. Mobile network

In order to enable the interaction of connected vehicles with cell phone networks on our testbed, we have implemented a 4G base station. For that aim, we have used the Universal Software Radio Peripheral USRP 2901 from National Instruments [8] and two computers equipped with the srsRAN software [9] and the Open5GS software [10], one becoming an LTE eNodeB and the other the 4G Evolved Packet Core (EPC).

There are other open software options, but we have chosen srsRAN software because of its compatibility with 5G-EmPOWER software, which allows using several wireless technologies simultaneously and provides a RAN Intelligent Controller (RIC) and a manager [11]. On the other hand, we have chosen Open5GS software as core due to its compatibility with S1 handover between two base stations, and its ability to implement a 5G core facilitating the switch to 5G SA in future use cases.

The srsRAN software allows running the eNodeB services involved in the 4G standard for data transmission. It implements a complete LTE eNodeB station via software. However, it is not directly compatible with USRP 2901. In order to ensure compatibility, a firmware has been installed in the USRP 2901, which enables the USRP to be used as an Ettus B210, compatible with srsRAN. Thus, the srsRAN software can directly communicate with the USRP by using the USRP Hardware Driver (UHD). This software communicates directly with the EPC (like a common eNodeB station does), and controls the USRP, which transmits and receives the data. On the other hand, the Open5GS software implements a version of the LTE EPC via software, running several key services necessary for basic operation.

Using both software we can configure multiple parameters of the network such as the MCC (Mobile Country Code) and MNC (Mobile Network Code), output signal power, bandwidth used, transmit and receive band of the base station, among others.

Fig. 5 shows the main elements of the 4G architecture that we have implemented through the srsRAN software and the Open5GS software. In order to be able to register clients in this network, it is necessary to acquire a special SIM card that comes with all the necessary keys in order to configure it in the EPC database of the 4G network, i.e., in the Home Subscriber Service (HSS).

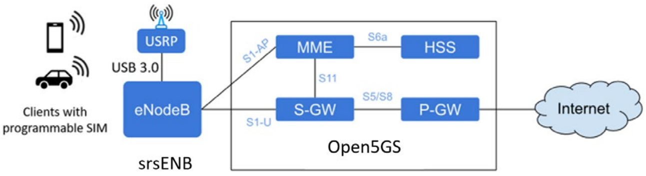

Fig. 5. 4G network architecture assembled through srsRAN software.

After starting up the base station, we have tested the coverage of the radio station by adjusting the antenna power parameter of the USRP between -60 dBm and -20 dBm. Using standard Wi-Fi antennas at full power, the coverage is approximately 40 m (including some obstacles). Such small ranges are interesting for the testbed as they allow the analysis of use cases related to lack of coverage and handover scenarios. Fig. 6 shows the range of two base stations installed at ETSIT (Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad de Valladolid), enabling the analysis of handover procedures when the vehicle travels along the right corridor of the building.

We have also tested the downstream and upstream throughput. The bitrate is directly proportional to the used bandwidth and the Physical Resource Blocks (PRB). The maximum bandwidth that our computer supports is 20 MHz (100 PRB), for which a maximum bitrate of 75 Mbps downstream and 50 Mbps upstream has been achieved. We have checked the values for several bandwidths and compared them with the results showed in [12]. All the results are in full agreement with the expected ones.

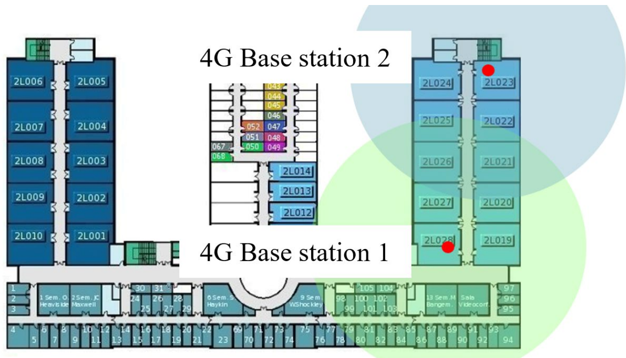

Fig. 6. Approximate coverage area of base stations at ETSIT building.

E. Optical access network

Regarding optical access networks, we have developed a GPON testbed which supports SDN management [13]. This testbed has been developed independently of the one we are presenting here but will be part of the complete testbed (as shown in Fig. 1).

The GPON testbed is shown in Fig. 7. The equipment and devices used in the testbed are from the Telnet-RI vendor. We have installed several level two and level three Optical Network Terminals (ONTs), and the SmartOLT 350 Optical Line Terminal (OLT), which complies with the ITU-T G.984 and G.988 specifications. The SmartOLT 350 supports up to four Passive Optical Networks (PON) with GPON interfaces, enabling data rates of 2.488 Gbps downstream and 1.244 Gbps upstream. The connection between the OLT and the ONTs is done by means of three spools of optical fiber (two of them 5 km long, and one 10 km long), two 1:8 splitters, and a distribution network whose links can be configured with lengths ranging from 100 m to 5 km. Consequently, the GPON

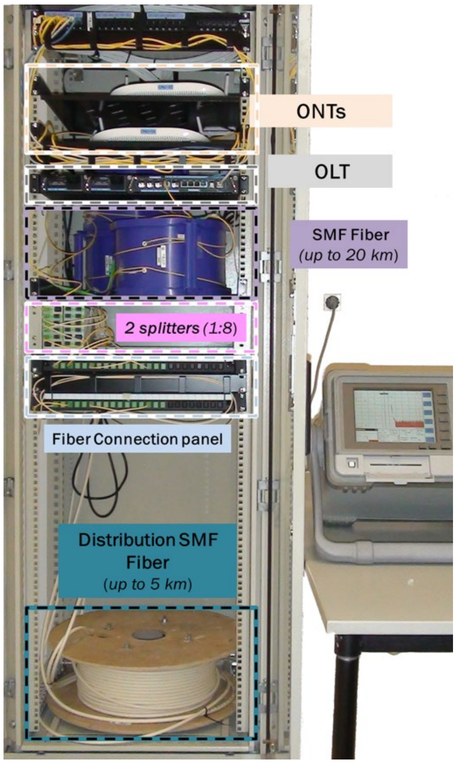

Fig. 7. GPON testbed

testbed allows us to have different propagation delays for different ONTs, which can be useful to test delay-sensitive use cases.

Finally, we have developed a SDN solution that permits to configure the testbed by means of OpenFlow using and external SDN controller [13]. The management automation of the access networks will allow to analyze the interaction between the different networks and services, which, in this case, considering the stringent delays and high bandwidths demand, is crucial.

III. Evaluation of a Use Case in the Testbed: Computation Offloading to the Edge for Autonomous Driving

Once the testbed has been presented, we now describe its utilization to demonstrate the use case of a connected and autonomous vehicle, with limited on-board computing resources, which offloads processing tasks in real time to an edge server. This use case is just an example of the possibilities offered by the testbed, whose final goal is to demonstrate the potential of 4G/5G networks and edge computing to provide high-capacity and high-value services to connected and autonomous vehicles.

A connected vehicle (the modified Amazon DeepRacer Evo previously described), moving through a configuration of the road circuit testbed, transmits the video from its camera, in real time, to an edge server located in the 4G testbed network. The edge server implements a guidance and decision-making algorithm which consists of three different modules: (1) image processing for line detection and tracking, (2) route decision making, and (3) determination of the required steering angle to be sent back to the vehicle.

In order to connect the vehicle to the 4G testbed network we have used a 4G modem (Huawei E3372) with a programmable SIM (sysmocom sysmoISIM-SJA2 [14]). This

SIM comes with the necessary keys to add the user to the 4G network user database. We have also configured the MCC and MNC codes of the base station to match the IMSI (International Mobile Subscriber Identity) of the SIM, so that the 4G modem will automatically search for and connect to the base station of the testbed. We have used two Intel NUC BXNUC9I9QNX to run the different network services. The first computer runs the eNB software. On the other hand, the second computer runs the core services (EPC) and the edge server running the guidance algorithm.

Then, regarding video transmission, each frame is compressed to JPEG and sent via UDP. There are more efficient options for real-time video transmission using codecs, but real-time video codecs introduce computation delays, and the parameter optimization to reduce those delays can be extremely challenging when having limited on-board computing resources (as in our case). The technique that we have used has low delay while providing good image quality.

Based on those images, the edge server executes the guidance algorithm. Although there are many complex but efficient guidance algorithms, we have implemented a simple method since our goal, as previously mentioned, is to focus on communication and computing services rather than on guiding techniques. The implemented guidance algorithm is based on the use of image processing techniques for line detection and tracking, and the Stanley algorithm to determine the steering angle parameter required for vehicle control [15].

Fig. 8 shows a visual example of the image processing performed at the edge server when receiving the camera images from the connected vehicle. First, a perspective transformation of the received image is performed. In this way, a bird's eye view image is obtained, which enables the correct computation of the angle associated to the trajectory to be followed by the vehicle. Then, the bird's eye image is filtered to get the central line of the road, and two points on this line, separated by a specific distance, are determined. Finally, a line between those two points is drawn and its angle and lateral displacement is computed.

The Stanley algorithm [15] determines the steering angle that the vehicle should set to follow the desired trajectory. The steering angle is a function of the angle of the desired route with respect to the longitudinal axis of the vehicle, the distance of the reference point of vehicle to the center of the route (lateral displacement), the speed of the vehicle and a constant parameter (*K*).

The guidance algorithm that we have implemented has several parameters that must be calibrated either experimentally or by simulations. Particularly, the distance between the two points to be selected in the central line and the *K* parameter of the Stanley algorithm must be calibrated. Moreover, since the Amazon DeepRacer Evo vehicle does not provide information about its speed, we have entered an approximate expected speed into the Stanley algorithm (which has also been experimentally calculated as 1 m/s).

When there are bifurcations in the roads, the edge server determines which route should be taken by the car. In this experiment, the complete route to be followed by the car has been predefined. To specify the route, we take advantage of the different colors of the central line of the roads. Therefore, a route can be defined by the sequence of colors to be followed by the vehicle. The circuit has red, blue and pink lines, which are connected with intermediate yellow lines (Fig. 2). Let us suppose that a red-yellow-blue-yellow-pink route has been defined. The image processing method searches for both the current color and for the next one to compute a filtered image showing the line to follow (Fig. 8). Thus, in the example, the method starts detecting and following the red color (current) but also the yellow one (next color). When it detects that the vehicle has entered the yellow section, it starts detecting and following the yellow color (current) but also the blue color (next one). This procedure is repeated until the last color is detected. When the end of the line with the last color of the sequence is reached, the vehicle is commanded to stop.

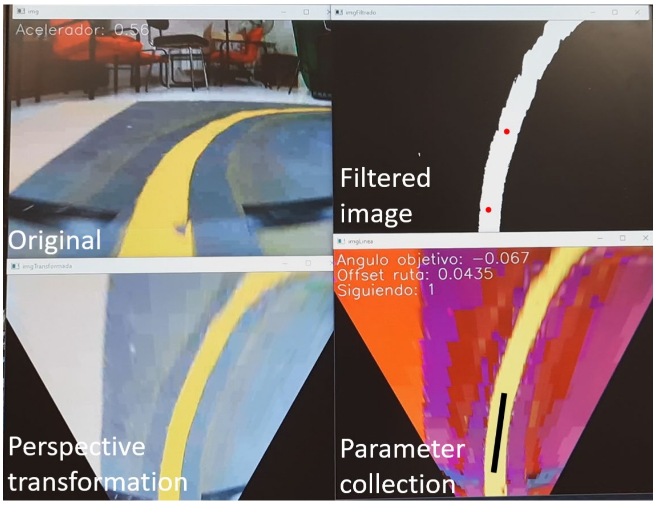

Fig. 8. Image processing performed at the edge server.

A video showing the demonstration of this use case can be seen at https://youtu.be/3tZkcO21vjg.

We have also performed an analysis of the delays (Fig. 9). The values shown are the average delays at each point. To carry out the processing delay measurements, we have analyzed the timestamp at different points in the software. To perform the network delay measurements, we have used the Qosium software [16]. This software allows us to obtain multiple network parameters for various data flows, such as delays, throughput, jitter and lost packets. To use the program, the Qosium Probe software is installed on the devices involved in the data flow to analyze (CORE, eNB and vehicle) and, on the other hand, the Qosium *Scope* software is installed on another independent machine. In this way, from Qosium Scope we can configure the network devices in order to send information about the network packets they are handling. Based on this data, the software can provide multiple network quality parameters such as delay in real time.

Fig. 9 shows the different delays in the testbed. It takes 12.6 ms for the vehicle to compress a frame and send it over the network. The uplink delay from the vehicle to the computer with the eNB services is 24.4 ms and the downlink delay 18.9 ms. The packets (encapsulated in GTP) are forwarded to the EPC, which decapsulates, implements packet routing and forwards them to the edge server. Forwarding and encapsulation/decapsulation tasks in the EPC take a time of 0.2 ms. The server takes an average of 33 ms to process a frame (although this delay could easily be reduced by using a more powerful server). In summary, a total delay of 90.9 ms is obtained due essentially to the compression time of the frames in the vehicle, the delay of the 4G network (which is consistent with the measurements reported in [12]), and the processing delay of the server.

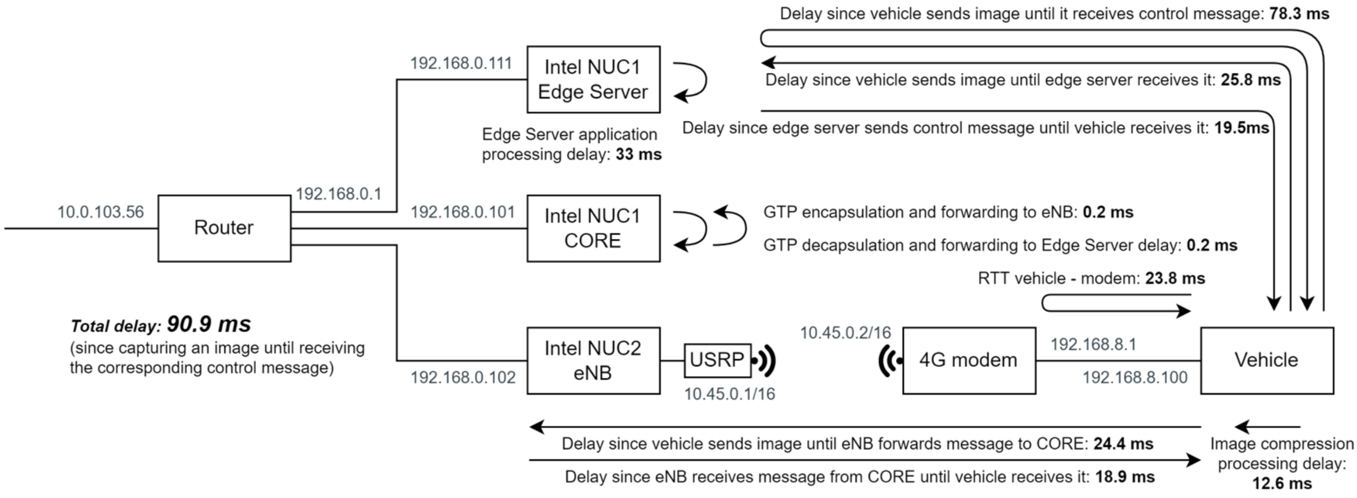

Fig. 9. Network architecture in the use case and measured delays.

One aspect we have not been able to analyze is the delay added by the 4G modem. This modem performs a network address translation (NAT) between the 4G network and the vehicle. Due to the impossibility of installing any software on the modem, we are only able to check the round-trip time (RTT) through an ICMP message. This delay is 23.8 ms, which suggests that the modem adds a considerable delay.

Despite the total delay, as shown in the video, offloading the guidance computation task to the edge is feasible at low speeds. However, it imposes a limit on the maximum speed of the vehicle, since it is no possible to react in time on curves. If, for example, the average speed of the car is 1 m/s, it means that the car travels 9.04 cm before reacting to any event in the road due to that round-trip delay. By doubling the speed, the distance is doubled, which makes it difficult for the vehicle to adapt to changes in the trajectory.

IV. Conclusions

5G networks together with edge computing are called to be key technologies to enable the deployment of new services with high bandwidth and low delay requirement. Connected and autonomous vehicles can benefit the most from these technologies. However, the whole deployment of an experimental set-up with real networks and vehicles cannot be easily affordable. To this end, we have built a flexible testbed to test several use cases regarding connected and autonomous vehicles. In this paper we have demonstrated the use of this model to test a real-time off-loading application, showing the flexibility and the possibilities of the testbed.

In the future, we will include new functionalities to complete the testbed. Currently, a 4G system has been implemented. We are currently deploying the RAN 5G version of OpenAirInterface [17] and we are also testing handover situations. As we have also described, we have developed a GPON testbed with SDN management, but requires to be integrated in the testbed to complete the whole architecture. In addition, the evaluation of different use cases will also be the subject of future work.

References

[1] A. Asensio, X. Masip-Bruin, R.J. Durán, I. de Miguel, G. Ren, S. Daijavad, A. Jukan, "Designing an efficient clustering strategy for combined Fog-to-Cloud scenarios", Future Generation Computer Systems, vol. 8, pp. 302-406, 2020

[2] L. Galluccio, C. Grasso, M. Grasso, R. Raftopoulos and G. Schembra, "Measuring QoS and QoE for a Softwarized Video Surveillance System in a 5G Network" *2019 IEEE International Symposium on Measurements & Networking (M&N)*, pp. 1-6, 2019

[3] 3GPP TR 38.913, "Study on scenarios and requirements for next generation access technologies", V16.0.0, July 2020, Available online: https://www.3gpp.org/ftp/Specs/archive/38_series/38.913/. Accessed on April, 2020

[4] T. Linget, "A visionary roadmap for advance driving use cases, connectivity technologies, and radio spectrum needs", 5G Automotive Association, Nov. 2020, online: https://5gaa.org/wp-content/uploads/2020/09/A-Visionary-Roadmap-for-Advanced-Driving-Use-Cases-Connectivity-Technologies-and-Radio-Spectrum-Needs.pdf . Accessed on July, 2021

[5] T. Cohen, C. Cavoli, "Automated vehicles: exploring possible consequences of government (non)intervention for congestion and accessibility", Transport Reviews, Vol. 39, pp. 129-151, 2019

[6] E. Coronado, G. Cebrián-Márquez, and R. Riggio, "Enabling Autonomous and Connected Vehicles at the 5G Network Edge", 6th IEEE International Conference on Network Softwarization (NetSoft), pp. 350-352, 2020

[7] https://aws.amazon.com/deepracer/, accessed on August 2021.

[8] https://www.ni.com/en-us/support/model.usrp-2901.html, accessed on August 2021.

[9] "Open-source 4G and 5G software", https://www.srslte.com/, accessed on January 2022

[10] https://open5gs.org/, accessed on January 2022

[11] "An O-RAN compliant near-RT RAN intelligent Controler for Het RANs", https://5g-empower.io/, accessed on January 2022

[12] M. Laner, P. Svoboda, P. Romirer-Maierhofer, N. Nikaein, F. Ricciato and M. Rupp, "A comparison between one-way delays in operating HSPA and LTE networks," 2012 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), pp. 286-292, 2012

[13] N. Merayo, D. de Pintos, J.C. Aguado, I. de Miguel, R.J. Durán, P. Fernández, R.M. Lorenzo, E.J. Abril, "Experimental validation of an SDN residential network management proposal over a GPON testbed", Optical Switching and Networking, vol. 42. pp. 1-14, 2021

[14] H. Welte, "sysmo USIM/sysmo ISIM User Manual" Available online: https://sysmocom.de/manuals/sysmousim-manual.pdf, accessed on June 2021.

[15] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo and S. Thrun, "Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing," 2007 American Control Conference, pp. 2296-2301, 2007

[16] https://www.kaitotek.com/qosium, accessed on January 2022.

[17] https://openairinterface.org/, accessed on January 2022.