electronics

MDPI

Article

SURROGATES: Virtual OBUs to Foster 5G Vehicular Services

José Santa [1,2,*], Pedro J. Fernández [1], Jordi Ortiz [1], Ramon Sanchez-Iborra [1] and Antonio F. Skarmeta [1]

[1] Department of Information and Communications Engineering, University of Murcia, 30100 Murcia, Spain; pedroj@um.es (P.J.F.); jordi.ortiz@um.es (J.O.); ramonsanchez@um.es (R.S.-I.); skarmeta@um.es (A.F.S.)

[2] Department of Electronics and Computer Technology, Polytechnic University of Cartagena, 30202 Cartagena, Spain

* Correspondence: josesanta@um.es; Tel.: +34-868-884-616

Received: 17 December 2018; Accepted: 20 January 2019; Published: 22 January 2019

check for updates

Abstract: Virtualization technologies are key enablers of softwarized 5G networks, and their usage in the vehicular domain can provide flexibility and reliability in real deployments, where mobility and processing needs may be an issue. Next-generation vehicular services, such as the ones in the area of urban mobility and, in general, those interconnecting on-board sensors, require continuous data gathering and processing, but current architectures are stratified in two-tier solutions in which data is collected by on-board units (OBU) and sent to cloud servers. In this line, intermediate cache and processing layers are needed in order to cover quasi-ubiquitous data-gathering needs of vehicles in scenarios of smart cities/roads considering vehicles as moving sensors. The SURROGATES solution presented in this paper proposes to virtualize vehicle OBUs and create a novel Multi-Access Edge Computing (MEC) layer with the aim of offloading processing from the vehicle and serving data-access requests. This deals with potential disconnection periods of vehicles, saves radio resources when accessing the physical OBU and improves data processing performance. A proof of concept has been implemented using OpenStack and Open Source MANO to virtualize resources and gather data from in-vehicle sensors, and a final traffic monitoring service has been implemented to validate the proposal. Performance results reveal a speedup of more than 50% in the data request resolution, with consequently great savings of network resources in the wireless segment. Thus, this work opens a novel path regarding the virtualization of end-devices in the Intelligent Transportation Systems (ITS) ecosystem.

Keywords: NFV; OBU; Data collection; MEC; 5G; ITS

1. Introduction

In vehicular scenarios, on-board units (OBU) have evolved from specific purpose units designed for telematic services, such as fleet management or road tolling, to generic networked nodes capable of interconnecting other in-vehicle devices, acting as mobile routers. These units can maintain connectivity with the infrastructure (Vehicle-to-Infrastructure, V2I) or other vehicles (Vehicle-to-Vehicle, V2V) using wireless communications, and the access to in-vehicle sensor data is becoming of paramount importance in Smart City scenarios, where the vehicle will be considered another (mobile) sensor [1]

The arrival of 5G technologies will complement already available communication technologies, such as 4G or IEEE 802.11 running Outside the Context of a Basic Service Set (OCB), formerly known as 802.11p. Virtualization, network orchestration, and multi-access edge computing (MEC) are novel strategies that can provide great benefits for Cooperative Intelligent Transportation Systems (C-ITS) [2]. Virtualization of network and computing nodes is a reality in cloud deployments, and it is expanding to the edge of the access network in designs involving MEC-based configurations, where processing is moved from the final devices and, partially, from cloud nodes to an intermediate layer near the access network.

Next-generation vehicular services, such as the ones in the area of urban mobility and, in general, those interconnecting on-board sensors, require continuous data gathering and processing, but current architectures in the field are stratified in two-tier solutions in which data is collected by OBUs and sent to cloud servers. As initially stated in [3], intermediate cache and processing layers following fog/edge computing principles would be highly beneficial for covering data sensing needs from vehicles in scenarios of smart cities/roads. The MEC paradigm can further extend this capability in V2I data flows, by taking advantage of the communication path through the access network in order to move such computing near the end-devices. This scenario presents a good frame to offload data analytics tasks from OBUs, which should focus on actions of higher priority such as maintaining vehicle connectivity, managing communication flows, or applying security measures to data traffic.

The migration of processing tasks to the network as virtual network functions (VNFs) is not direct, and a proper management plane should be integrated, above all in dynamic scenarios as the vehicular one. In this line, management and orchestration facilities (MANO) are becoming popular to dictate the operation of virtualized infrastructures. In the C-ITS domain, virtualized functions should be instantiated on-demand and an efficient real-time management of resources should be provided. These strategies closely follow the precepts defined for the upcoming 5G architectures. Therefore, the flexible management of network and processing resources in 5G will be greatly exploited in the vehicular ecosystem [4].

One of the most important markets within the commonly known as "connected car" field is monitoring of vehicle vital signs, which will be the seed of many services. This is basically the extraction of on-board sensor data but requires a proper support of a V2I network dealing with vehicle communication flows. Although initial approximations in the area of fleet management were exclusively based on 2.5 G cellular networks, soon new approaches appeared exploiting the advances of new 3G/4G technologies together with vehicular ad hoc networks (VANETs) [5]. These hybrid solutions could group vehicles in clusters and select a representative proxy to communicate with the Internet. This reduced data exchange with potentially congested 3G base stations, by aggregating and compressing monitoring data [6]. Although recent approaches follow a similar approach in new application areas such as bicycles [7], the research community has recently focused on the expansion of Internet of Things (IoT) in the vehicular domain, in which it is known as the Internet of Vehicles (IoV) [8,9]. 5G advances such as network function virtualization (NFV), MANO or MEC, together with new 3GPP radio advances and the hybridization with other vehicular and IoT communication technologies, are now found as useful enablers to cope with the IoV vision and cover data gathering and processing without the need for such clustering solutions. SURROGATES follows this approach to go a step further in IoV and outperforms currently available cloud-based schemes.

In this paper, we propose the SURROGATES solution, which aims at virtualizing physical OBUs into VNFs in which computationally-intensive tasks and data caching are performed. The virtual OBU (vOBU) becomes the source of pre-processed information to be used by cloud services or other data analytics modules, which finally feed user applications in the areas of traffic efficiency or even safety. vOBUs are instantiated in the access network domain, implementing an MEC approach, and MANO features are used to scale the platform automatically. Apart from computing performance improvements, the MEC layer with vOBUs offers a proxy point where data requests from services are solved without the need to access the actual OBUs and, hence, consume radio resources of the wireless vehicular network. This is done transparently to the services or users, which only perceive better performance in the access. Additionally, this approach extends the life of battery-powered physical OBUs such as the ones installed in light vehicles [10], since energy-consuming tasks regarding processing and communications are limited. Thereby, the main contributions of this work are the following: (i) the integration of virtualization techniques in vehicular environments in order to improve the system efficiency and adaptability is explored; (ii) an end-to-end architecture based on NFV and MEC paradigms is proposed and analyzed; and (iii) a real implementation and deployment of the aforementioned architecture is presented and the results extracted from a series of validation tests are discussed.

The remaining of the paper is organized as follows: Section 2 places the work in the literature and presents the advances beyond the state of the art; Section 3 presents the general architecture of the NFV-powered edge computing solution; Section 4 details the implementation details of the system; Section 5 includes the evaluation of the platform in terms of validation and performance indicators; and, finally, Section 6 concludes the paper and introduces the future research directions.

2. Related Work

Virtualization of vehicular resources has been considered in the last years to improve performance, perception and context-awareness by using cloud architectures [11,12]. The research has passed from vehicles using cloud resources to real vehicular cloud computing, involving infrastructures and OBUs in the cloud, or even hybrid approaches.

On the OBU side, the usage of virtualization has been also explored. In [13], the authors presented the idea of improving security of vehicular units by using virtualization and controlling the access to real hardware. A more up-to-date conception of this idea is the usage of NFV in the same OBU to create a flexible architecture that could be updated and managed remotely [14]. In this case, a proper architecture is provided to grant access to resources depending on service needs. Even migration schemes of virtual machines between OBUs have been considered to offer services adapted to available resources [15].

The SURROGATES proposal is just between the cloud and OBU planes, by exploiting the computing capabilities of network nodes near the edge of the access networks. This novel network architecture, known as MEC, is a very recent area of study in vehicular scenarios, as the related works included in this section reflect. An MEC layer implemented in road-side units was presented in [16]. The novelty here comes from the distribution and kind of road-side units used, developed as drones, but the MEC layer is directly mapped on physical devices without presenting a real advance. The improvement implied by MEC for vehicle to vulnerable road user communications was evaluated in [17], using 4G simulation and observing up to 80% of latency reduction as compared with a cloud-based system. In [18], the MEC paradigm was used to improve the performance of a delay-tolerant network, by providing caching of messages at the access network for temporally disconnected vehicles. In [19], a way to orchestrate computing and caching in vehicular scenarios between MEC and remote servers was proposed. In [20], an MEC layer was proposed to process information coming from vehicles implementing a vehicular social network, although the potential of NFV and MANO was not exploited to cope with the dynamism of the system. The same lack is found in [21], in which a content delivery network used a MEC-based architecture to speed-up data sharing among vehicles.

An interesting feature included in the architecture presented in [22] is the combination of MEC and a hybrid vehicular communication system, which is also considered in the current work. The authors describe the usage of MEC for local storage and processing, and 4G together with G5 (European profile of 802.11 OCB) for vehicle connectivity. Unfortunately, NFV flexibility was not exploited, and the architecture was only partially developed and evaluated. The architecture presented in [23], although it was not validated, is a generic 5G platform supporting NFV and orchestration following ETSI guidelines, and hybrid communications are supported at the infrastructure side by the MEC layer, in order to be open for different vehicular communication media. In the same way, the work in [24] implemented a MEC layer by using VNFs, which is also the implementation strategy followed in our solution. Authors described a procedure to process status information collected from vehicles and detect potential collisions. However, this comprises the development over a 5G platform of a reference vehicular application that is not open to further ITS services and is not evaluated in terms of validation and performance.

Being also a solution addressing MEC computation, SURROGATES bets on a novel idea that has not been considered in the literature up to now: virtualizing real OBUs at the edge in order to create always available OBUs where computing intensive operations are executed while saving current and past state information. Real OBUs will maintain a proactive communication with its virtual substitute, in order to provide monitoring parameters that will feed a set of services. This presents a generic approach for the vehicular domain. This information will be pre-processed at the virtual unit, which will serve any request of information about the OBU from the same wired network, avoiding the access to the itinerant real OBU, which is subject to potential communication losses. vOBUs are dynamically managed as VNFs orchestrated by a MANO entity, which is essential for scaling the system and supporting the dynamism of vehicular scenarios.

3. Architecture of the NFV-Powered Edge Computing Approach

The main architecture of the SURROGATES solution is shown in Figure 1. As can be seen, the approach is to create virtual images of OBUs (i.e., vOBUs) as VNFs at the edge of the network, which can be based on road-side units, cellular base stations, WiFi access points, etc. These VNFs pre-process raw data from in-vehicle sensors that is sent through a monitoring middleware deployed in both the real OBU and the NFV-based one at the edge virtualization domain. Data to be collected from the OBUs consider navigation information, diagnosis and mechanical monitoring details of the car, and readings from other sensors connected to the OBU. Data analytics tasks envisaged at vOBUs include pre-processing, aggregation or pattern recognition involving individual vehicles, while global processing of data involving high-volume Big Data is performed at the cloud virtualization domain. Processed or cached information are accessed by vehicular services using vOBUs or the data analytics module. Hence, vOBUs act as computing offloading units for real OBUs and as a proxy in the downstream channel from services and users. This solves the issue of accessing the physical OBU from multiple services and requiring monitoring data when the OBU is temporally offline due to mobility issues.

An OBU Manager module has been defined to be the initial contact point for physical OBUs wanting to connect to a virtual counterpart. This manager is also a VNF, and communicates with the MANO entity to indicate the need for a virtual OBU instance. The MANO entity offers the capabilities to manage all NFVs, and it provides the OBU manager the key function of scaling the system in the edge virtualization domain when new vehicles connect to the platform.

The vehicle connectivity is managed by a hybrid approach with 802.11 OCB and 4G technologies, and using an IPv6-based network middleware based on our previous research [25]. In the multi-homed scenario considered, the OBU, which is in fact a mobile router, selects the most convenient interface, while, at the same time, seamless handovers are performed. This middleware allows the physical OBUs to connect to vOBUs, but also to the general Internet. Additionally, an in-vehicle wireless network is available for user devices potentially accessing services at the cloud virtualization domain.

Message exchanges performed in the whole platform use Representational State Transfer (REST) protocols. Communications with the MANO entity use the Application Programming Interface (API) of the orchestrator. The virtualized infrastructure manager (VIM) used, which is detailed in the next section, allows the creation of different virtualization domains.

It is important to note that the platform is open to any transport means. Although regular road vehicles allow the collection of many on-board sensors, our previous works note the potential of interconnecting two-wheelers in smart city and safety scenarios [10]. SURROGATES can both offload computing from OBUs and still present the vehicles as accessible continuously, distributing computation across access networks.

Among the potential services to be implemented on the top of this architecture, the ones within the urban mobility field can exploit the MEC features specially. Pollution monitoring using clustering techniques, tracking and predictive diagnosis, congestion detection, or healthy mobility through urban areas can benefit from data pre-processing at vOBUs, alleviating growing computing loads at cloud data centers.

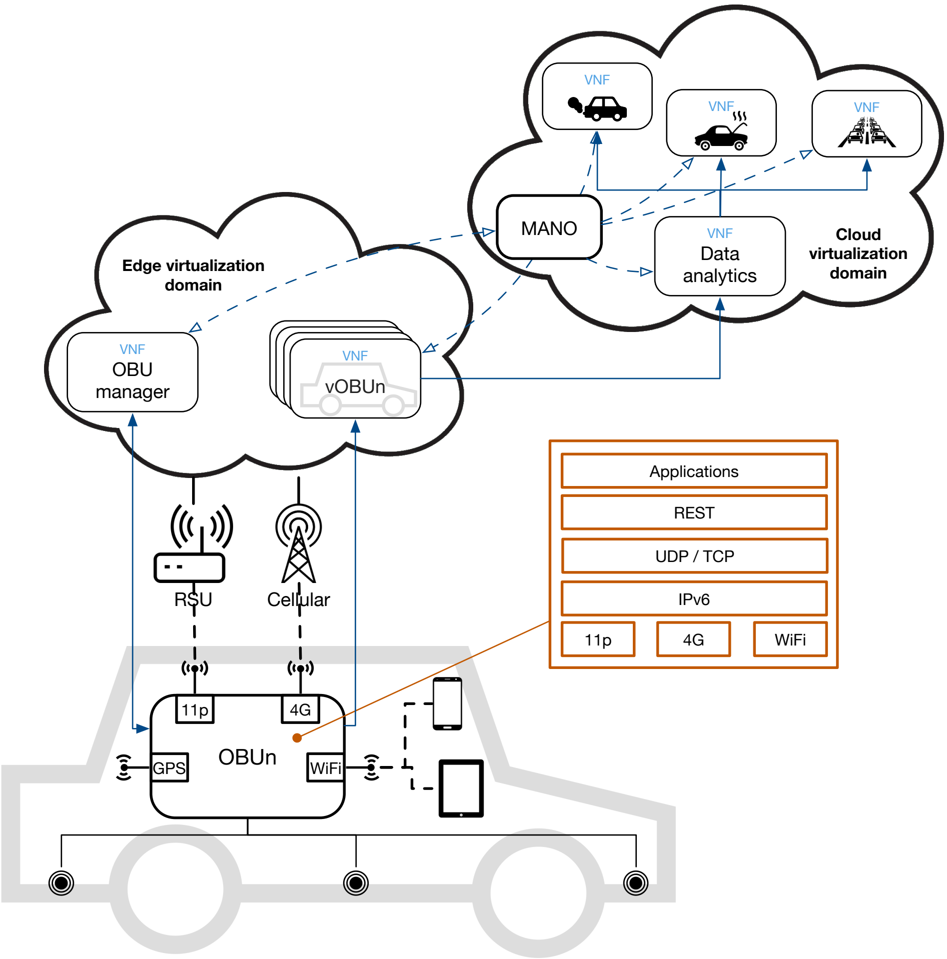


Figure 1. Overall architecture of the virtualization infrastructure.

4. Deployment and Operation of the Solution

The SURROGATES solution has been implemented in a real testbed using Open Source MANO (OSM) (https://osm.etsi.org) and OpenStack (https://www.openstack.org), as MANO and VIM, respectively. As it is shown in Figure 2, a single instance of OSM is used for the management of platform VNFs, while two OpenStack domains have been deployed. The one including MEC capabilities hosts vOBU VNFs, while the cloud OpenStack domain contains the rest of modules. Three network services (NS) have been deployed for the vOBUs, management of the platform and final services.

The bootstrapping of the vOBUs is depicted in Figure 2 on the left (orange color). Upon starting up the physical OBU, if it is the first time that the unit connects with the platform, a *vOBUrequest* is sent to the well-known address or domain name of the OBU manager (1). Considering that there is not a vOBU allocated for the OBU, a new one is required to OSM using its northbound interface (2). Then, OSM instantiates a new vOBU (3) and sends the details of the NFV to the OBU manager (4), which, in turn, provides the vOBU access details to the physical OBU (5). From now on, all data extracted from vehicle sensors are collected by the vOBU (6). Given the localized nature of MEC processing, the physical OBU will launch a new vOBU request when the vehicle drives outside the area covered by the edge domain.

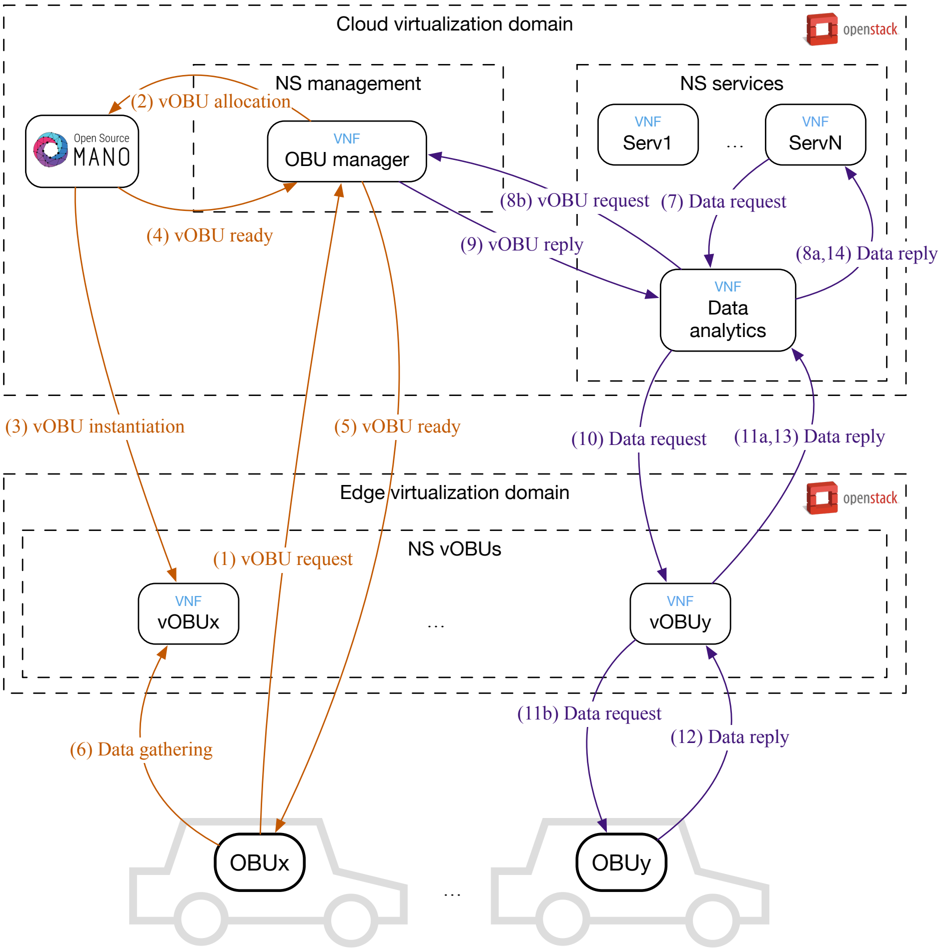


Figure 2. Operation of the system for virtual OBU (vOBU) bootstrapping (**left**) and data gathering (**right**).

Given that vOBUs are deployed in more powerful servers in relation to OBUs, they do not only offer more processing throughput, but also the capability to run a wider variety of software by using x86_64 (CISC) equipment. This inherent point in NFV is highly relevant in the transport ecosystem and, in general, the embedded hardware field, since most of the OBUs are usually low powered ARM (RISC) architectures. In addition, these elements are usually vendor blocked and limited in terms of library upgrades and integration by closed firmware, while VNFs can evolve their software independently, easily adopting new technological trends or applying new software development techniques.

Data requests from final services can be resolved at three layers in the platform, as shown on the right of Figure 2 (purple color). When a service VNF requires data from the vehicular platform, it launches a *Datarequest* message that is processed by the data analytics module (7). If the data had been fetched previously or comply processed values, the request is locally resolved (8a). However, if data from a particular vehicle is necessary and it is not cached, the involved vOBU is searched through the OBU manager (8b, 9), and the request is forwarded to it (10). The vOBU can locally resolve the request most of the times (11a), since the monitoring middleware collects data continuously from the vehicles (6). However, if an instantaneous data about a particular information source is needed, the request could be forwarded to the physical OBU (11b).

The way data is accessed and processed reduces the request resolution delay, saves wireless resources in the last mile network segment, and deals with the issue of several services wanting data from the same vehicles. Additionally, it is important to bear in mind that vehicles are connected through a mobile network that can suffer from eventual disconnections and can be an expensive channel in terms of cost and energy, above all when considering two-wheel vehicles.

5. Evaluation

The base platform presented above has been deployed and tested within the 5GINFIRE project ecosystem and using a testbed at the University of Murcia, Spain (UMU). For the sake of clarity, the deployment done at UMU is detailed and used in the tests, given the flexibility offered by operating our own OpenStack and OSM deployments. The architecture has been evaluated both in terms of validation and performance, analyzing its operation and using real user-level software exploiting the data collected from vehicles.

5.1. Testbed Set-Up

The deployment considered in the testbed is depicted in Figure 3. The vehicular network used to collect data from vehicles is the one presented in [25]. It supports two communication channels using 3G/4G and 802.11 OCB, although only the 3G/4G one is used in these tests, as can be seen in Figure 3. All traffic generated during the trials within the platform was IPv6 and, given that the telecom operator does not support it, an OpenVPN tunnel was established with a router in our laboratory at UMU premises. This is the router in charge of managing IPv6 network mobility, although this function is not used in this evaluation for the sake of clarity.

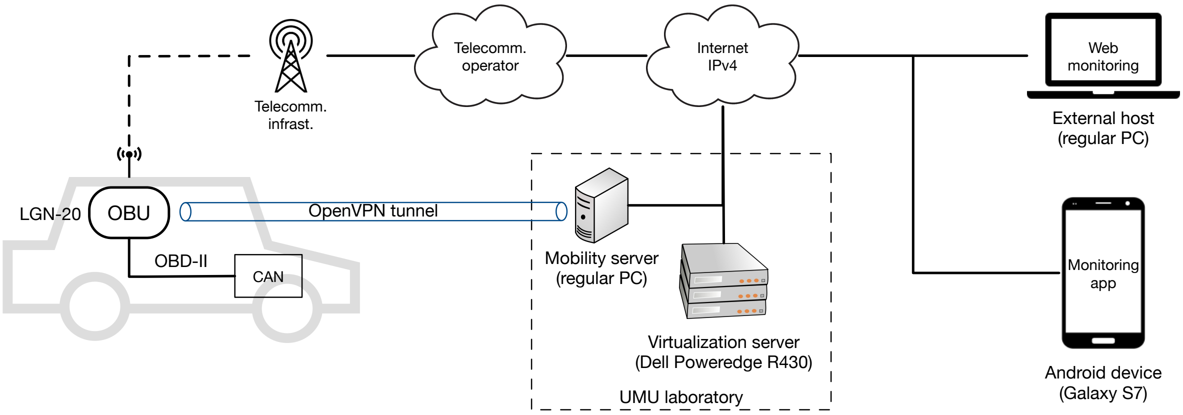


Figure 3. Deployment carried out in the testbed.

The employed OBU was a Laguna LGN-20 unit from Commsignia (Budapest, Hungary), running the communication stack described in [25] and provided with a TP-LINK MA-180 modem for 3G/4G (fully high-speed packet access (HSPA) compliant) connectivity. It was connected to the in-vehicle Control Area Network (CAN) bus using an On-Board Diagnosis (OBD)-II interface to gather vehicle state data. The used car was a Renault Clio 4th generation (Boulogne-Billancourt, France), while the OBD-II interface was an OBDLink SX with a USB connector.

The virtualization infrastructure ran on a Dell Poweredge R430 server (Round Rock, TX, USA) with Dual Socket Intel(R) Xeon(R) CPU E5-2630 v4 (Santa Clara, CA, USA) at 2.20 GHz. It ran CentOS 7 (GPL), using OSM version 4 (ETSI, Sophia Antípolis, France) and OpenStack Pike (OpenStack Foundation). OSM was configured with four cores and 24 GB RAM, while OpenStack was provided with eight cores and 24 GB RAM. This virtualization server was physically installed next to the mobility server, in the UMU laboratory.

VNFs have been described and provided with a proper software image, all of them using a base Debian 9 operating system (GNU). In order to make easier the software configuration and upgrade, the base images instantiated onto the OpenStack by OSM were provided and configured with their software depending on their kind by means of Ansible scripts. Four VNF types have been on-boarded in the platform according to the roles described previously:

- vOBU, provided with 1 CPUs, 1 GB RAM and 5 GB HD.
- OBU manager, provided with 2 CPUs, 1 GB RAM and 5 GB HD.
- Data analytics, provided with 2 CPUs, 1 GB RAM and 10 GB HD.
- Vehicle monitoring service, provided with 2 CPUs, 2 GB RAM and 40 GB HD.

Several instances of vOBU can be launched as described above, however, during the tests a single vehicle was used. Moreover, as aforementioned, vOBUs are deployed in an individual OpenStack virtualization domain, while the rest of VNFs operate in a different one to be deployed usually in a data center. In our deployment, both virtualization domains have been set-up in the virtualization server. Hence, the data path between the cloud and edge layers is reduced to a virtual connectivity between VNFs running on the same server. This means that our performance results have much more room for improvement when edge and cloud planes are deployed at different and distant locations.

Vehicle data is maintained by vOBUs, while the data analytics module aggregates data from all vehicles and feed the monitoring software service. The data analytics module saves all collected data, calculates parameters, and stores them in a MySQL database. In turn, the monitoring service is powered by Grafana version 5.3.1, which is a web service configured to show several vehicle parameters using plots. This VNF generates periodic data requests gathering vehicle data as explained above.

Additionally, the VNF for data analytics is also capable of processing vehicle data requests coming from outside. In our case, we have implemented an Android application to also access vehicle data from a mobile device. In the testbed, this application ran on a Galaxy S7 (Samsung, Seoul, Korea) with Android Nougat. Hence, as can be noted (see Figure 3), the testbed supports two equivalent methods to gather vehicle data: web-based and using a mobile device app.

5.2. Evaluation Methodology

A number of trials have been carried out with the aim of obtaining significant results, above all for the performance evaluation. The test location has been the north area of the Espinardo Campus at the University of Murcia, depicted in Figure 4. For each individual trial, the vehicle starts moving at the upper green mark in the map and stops at the lower red one. The length of the circular path is 1 km.

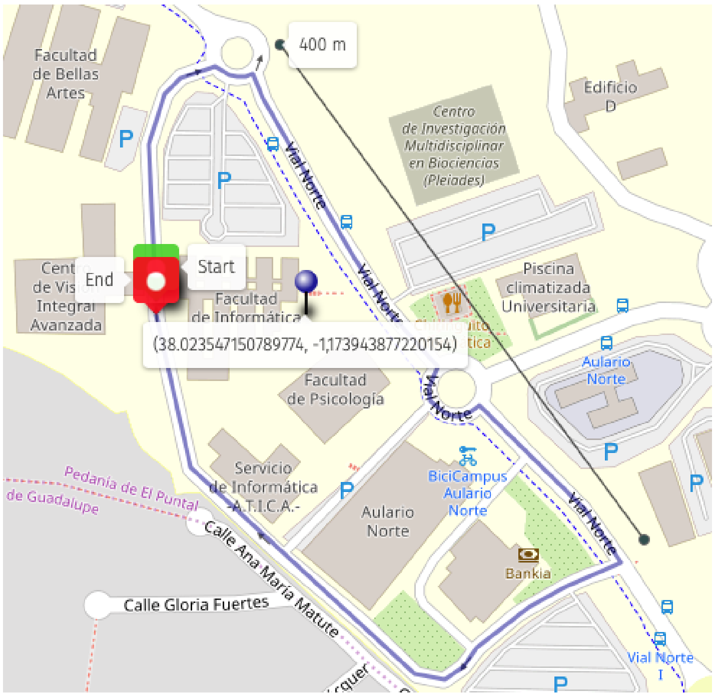


Figure 4. Test-site and path followed by the car.

Although OBD data is periodically reported by the vehicle at a rate of 1 Hz, we have used the Android application to regularly ask the platform for car parameters. The way these requests are organized depends on the three experiments carried out. A first one is focused on evaluating the operation of the platform when different cache success rates are obtained in the vOBU. A cache hit represents a data request that is solved by the vOBU instead of asking the physical OBU. This behavior has been emulated by a success rate setting included in the app, varying from 0% of cache hit rate to 100% of success. Upon receiving a request from the app with a particular hit rate, the vOBU asks its physical counterpart or locally resolve the request with the given probability. Five percentage values have been used and, for each one, five trials have been carried out. Each data request involves a TCP transaction, and the reply is marked with a round-trip delay time (RTT) as it traverses the different nodes of the data path. This way, we can analyze the request RTT in different network segments.

A second experiment is focused on studying the ratio of request losses. Since the network rarely loses packets due the the usage of a cellular connection and the TCP protocol, we have forced the system to be more rigorous regarding the time the vOBU waits for a response from the OBU. This behavior is more realistic because it is expected that packet losses appear when involving crowded urban areas, rural locations with poor coverage, or when communication technologies with a limited deployment are used. Hence, we have developed the vOBU software to provide the cache data when a certain delay threshold is reached. This way, when the OBU is accessed, the communication is not blocked waiting for TCP timeouts. This delay threshold is considered with values of 300 ms, 400 ms and 500 ms, given that, in our testbed, the time measured at the vOBU to receive a response from the OBU is around these values after a set of initial tests. Again, for each configuration, we have carried out five trials.

The third experiment has been carried out to analyze the system scalability and the time a new vehicle needs to register and obtain a vOBU instance. In this case, it is important to note that our solution hides the instantiation time by using a pool of vOBUs that is dynamically scaled by OSM through monitoring the free VNFs. Although the number of free vOBUs to be maintained in the pool is static at the moment, it is actually dependent on various external factors. This opens up future work to make an estimation of the number of cars that might be traversing a certain VNF infrastructure based on location or time (day, month) by means of machine learning and open data sources. For the tests, 30 registrations have been carried out, measuring the whole time needed by the registration process.

As can be noted, a total of 45 real trials for the first two experiments, plus the registration trials, are considered in the results presented in the next section. For each trial configuration, the delay times are averaged and, specifically for the second experiment, the ratio of "lost" requests is obtained. Moreover, for each configuration, the 95% confidence interval has been calculated for these two metrics and included in the plots.

5.3. Validation of the Platform

The correct operation of the SURROGATES solution has been successfully checked by accessing the Grafana monitoring tool. The screenshot provided in Figure 5 shows the periodic data collected from our car during seven of our trials. From up to down and left to right, the data showed are the speed, revolutions per minute (RPM), engine temperature and air flow. There exists a direct relation between speed and RPM, as normal, and the air intake represents the moments in which the car accelerates. The engine temperature increases at the beginning and maintains about 80°.

Data processed by the monitoring service are obtained by periodical data messages sent by vehicles to the vOBU and then reported to the data analytics module, which finally reach the service. Figure 6 shows the Android application when data requests are generated from the mobile device. Here, the percentage of engine load is showed, including the current value and the progression in the trial. Below the plot, a set of log data is included about the last request, printing the parameter asked for, a timestamp, the delay threshold (if any) and the request RTT measured at the vOBU, data analytics module and mobile device. These are the base data gathered for the performance results presented in the next section.

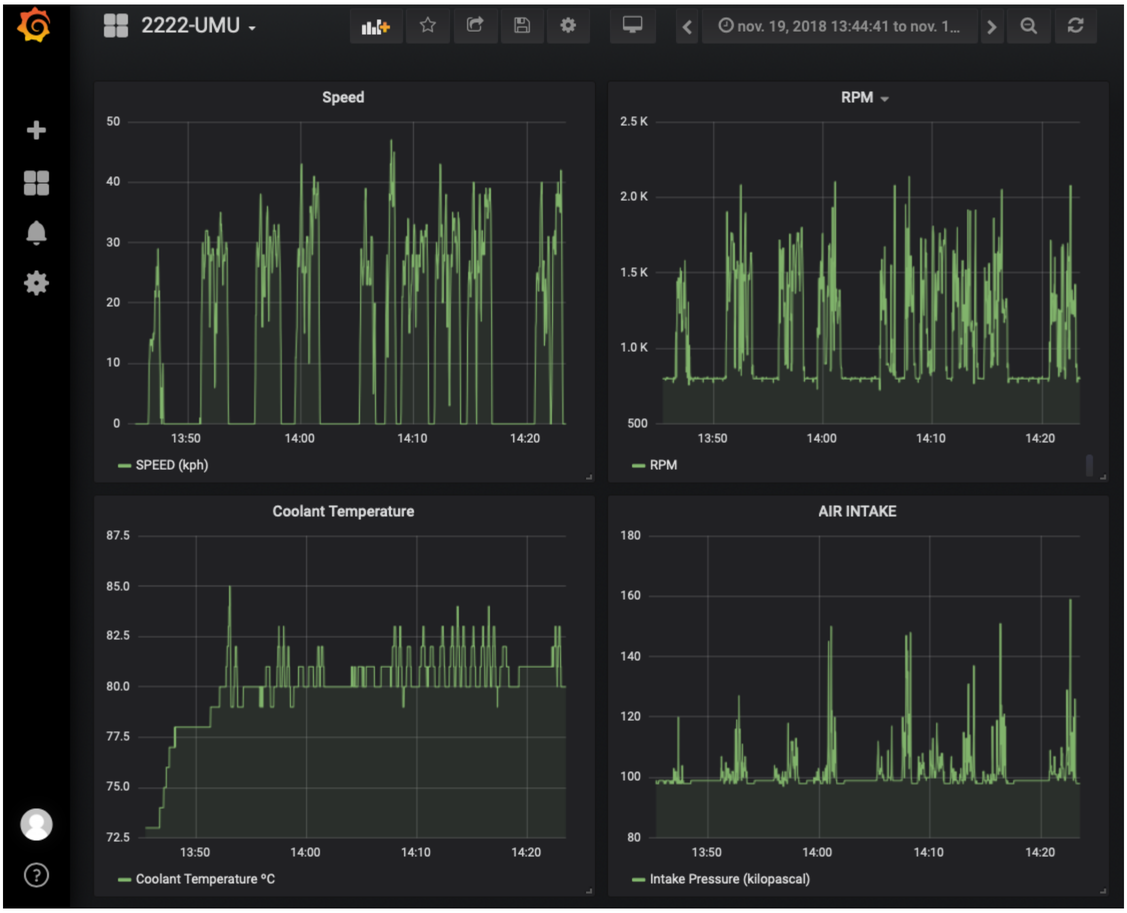
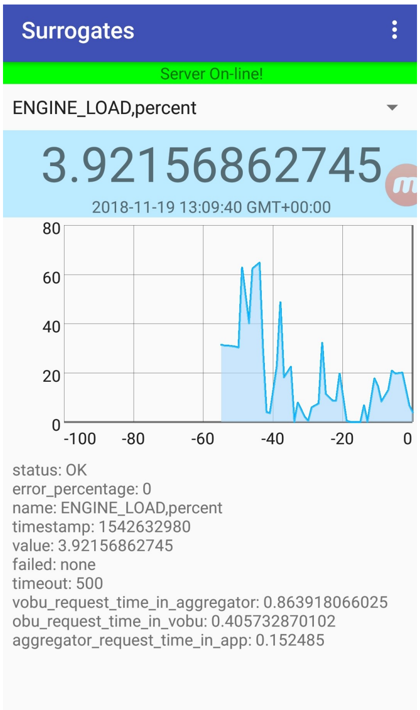


Figure 5. Screenshot of the Grafana view for monitoring vehicles.



status: OK
error_percentage: 0
name: ENGINE_LOAD,percent
timestamp: 1542632980
value: 3.92156862745
failed: none
timeout: 500
vobu_request_time_in_aggregator: 0.863918066025
obu_request_time_in_vobu: 0.405732870102
aggregator_request_time_in_app: 0.152485

Figure 6. Screenshot of the Android app requesting vehicle parameters.

5.4. Performance Analysis

The results collected in the first experiment, which evaluates the request RTT under different vOBU hit rates, are included in Figure 7. With the aim of removing the impact of the mobile network used by the Android phone, the RTT measurements are taken from the data analytics module. The overall request RTT trend shows that, as the request hit ratio increases, the delay is greatly reduced, given that less data requests are sent to the physical OBU and are solved directly by the vOBU. The variability of results also diminish when the hit rate is higher, attending to the confidence intervals. This is due to the connection with the vOBU at the infrastructure is far more stable than using the wireless channel to reach the OBU. When a hit rate of 0% is employed, no RTT values are included for vOBU, given that in this case all requests are sent to the OBU. On the contrary, when a hit rate of 100% is considered, no RTT values from the OBU are reported, since all requests are solved by the vOBU. The overall RTT results when vOBU answers data requests is stable through all configurations, but the ones obtained when accessing the OBU tend to slightly improve when higher hit rates are used. This is attributed to the reduction of requests queued for OBU resolution.

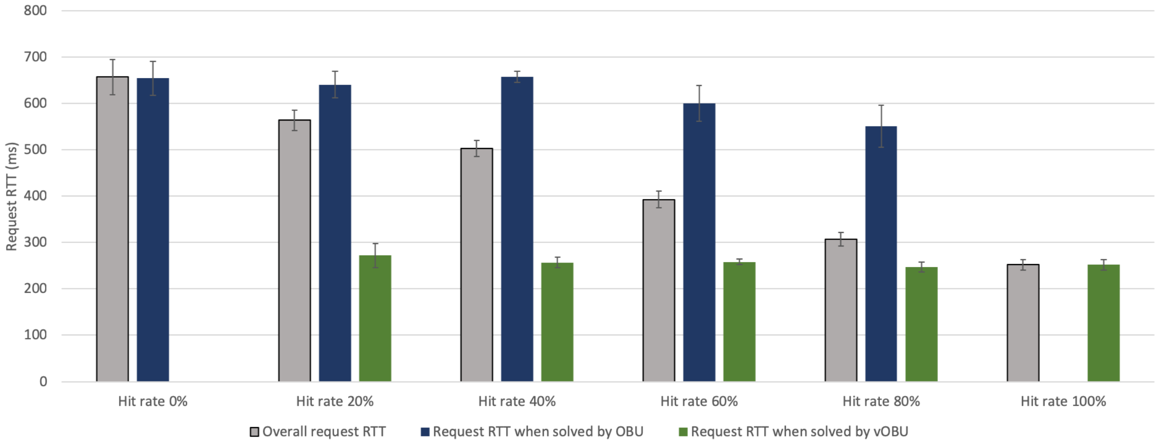


Figure 7. Round-trip delay time (RTT) values for requests when the vOBU hit rate varies from 0% to 100%.

With the aim of highlighting the impact of the wireless medium, we have analyzed the time spent by requests on each network segment in this first experiment. Figure 8 shows the overall request RTT values broken down into the time the transactions spent on each part of the network. It can be seen in the plot that the time the request is present in the infrastructure (i.e., from the network analytics module to vOBU) is maintained at around 230 ms, while the time consumed by the OBU ranges from 400 ms, when a hit rate of 0% is used, to 22 ms, when a hit rate of 100% is used. Of course, these values embrace the whole transactions, including the processing time needed by each node, but it is evident that the vOBU proposal is able to soften the impact of the wireless last mile when accessing the OBU.

The results of the second experiment about the ratio of request losses are plotted in Figure 9. Here, the vOBU hit ratio has been fixed to 0%, hence all requests are addressed to the OBU. In this analysis, the losses are represented by those requests processed by the vOBU after waiting for a response from the OBU for a parametrized time. It can be seen that a timeout of 300 ms implies that nearly 90% of the requests are solved by the vOBU. When increasing the timeout, the increase of requests successfully processed by the OBU is noticeable, at the expense of a greater delay, as explained above. The application of this timeout is needed in order to assure a proper operation of the system, since TCP transactions with default connection timeouts imply large waiting times when the OBU connection fails. In this case, the vOBU gives a good second choice to provide a cached data from the OBU and allow applications to receive sensor data although the OBU is not available. A time stamp is provided together with the replies, in order to make the application aware of the age of the data.

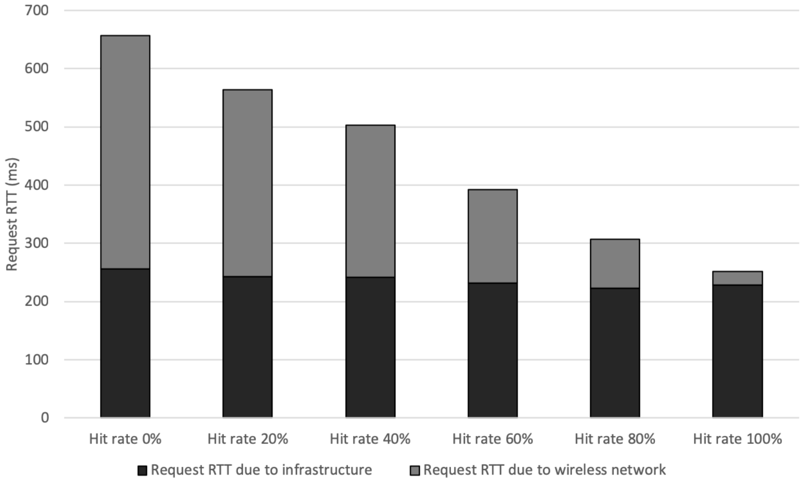


Figure 8. Request RTT per vOBU hit rate attending the network segment involved.

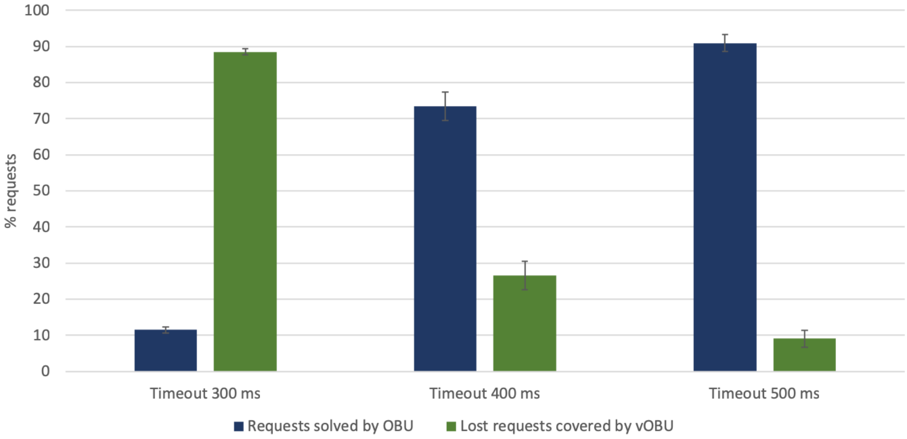


Figure 9. Requests solved by OBU and vOBU as the response timeout varies.

Regarding the time needed for the registration of a new vOBU, Figure 10 includes the mean time involved by the process when using the 4G connection. It can be seen that the registration involves near 400 ms in our testbed, and this value is independent from the number of already assigned vOBUs, since the VNFs are pre-loaded and maintained in a pool beforehand.

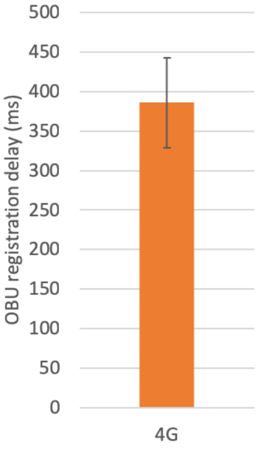


Figure 10. OBU registration delay.

The performance results clearly indicate the delay improvements and the expected request delivery ratio when the SURROGATES proposal is used. The speedup depends on the amount of requests solved by the VNF nodes. In our case, the study has been focused on the resolution of requests from the vOBU, but these could be also solved by the same data analytics NFV instance, even improving the results. A deployment in which the edge and cloud planes were set-up at different locations would also imply better results. In general, when using a HSPA-capable cellular network (3.75 G), the time required to access the OBU is more than the double than the time spent by the vOBU to reply the data request. As more users and services access the same OBU, this improvement will be more evident, since the data hit rate is expected to be higher as the rate of requests increase.

As compared with other solutions for vehicle data gathering, such as those based on VANET clustering and the use of proxies for V2I communications [5–7], SURROGATES focuses on downstream data requests and the reduction of their impact by vOBU caching, and moves data aggregation and further pre-processing capabilities to the MEC layer. For continuous monitoring of vehicles, uplink messages are sent using multiple communication technologies [25], following the recent trend of hybridizing data links, and integrating new-generation 5G and IoT technologies such as Low-Power Wide Area Networks (LPWAN) in the vehicular domain [26].

In addition, it is important to bear in mind that the SURROGATES approach makes services and user-level application agnostic of communication failures with OBUs, which is something quite probably to occur due to mobile wireless networks used by vehicles.

6. Conclusions

The coming of 5G technologies will imply a revolution in transportation services, with lots of new services in the areas of safety, traffic efficiency, and infotaintment. In fact, new communication capabilities of 5G even offer a competitor of the IEEE 802.11 OCB technology. In this paper, we have exploited the virtualization and edge computing capabilities of 5G and, more concretely, the potential of MEC to virtualize OBU operations.

The SURROGATES approach bets on a virtual substitute for OBUs to offload processing serves as a communication proxy to report on-board data, and cache information to resolve requests from services. This is accompanied by a whole infrastructure to gather data from vehicles and the management of the virtualized platform to create vOBU instantiations using OSM capabilities. All components of the architecture have been developed using a proper virtualization environment powered by OSM version 4 and OpenStack Pike, and a reference vehicular communication platform has been used to evaluate the proposal.

A validation has been carried out through the implementation of a monitoring service using Grafana and an Android app, demonstrating the good operation of the proposal. The performance evaluation has been focused on the delay implied by data requests and their losses due to communication problems in the wireless segment. The results reveal a 50% reduction in the overall delay when using the virtual OBU proposal, and the possibility of avoiding the unstable wireless channel thanks to the local data cache maintained by vOBUs. This further improves reliability of vehicular services such as monitoring platforms, which can suffer from connection outages due to mobility or poor coverage.

Future lines of work include the research on vOBU migration when different virtual domains are traversed and this VNF needs to be maintained near the access network. Moreover, the SURROGATES approach is being ported to the monitoring of light vehicles by using IoT communication technologies and Constrained Application Protocol (CoAP) messaging. Here, the offloading of computing tasks will be further analyzed, taking into account common data preprocessing and curation to be migrated to vOBU, which will also imply an impact on OBU energy usage. Finally, the registration process will be transparently managed without the limitation of using a name registry for the manager thanks to SDN, which will also be useful for vOBU migration.

Author Contributions: Conceptualization, J.S. and A.F.S.; Data Curation, J.S., P.J.F. and J.O.; Funding Acquisition, J.S. and A.F.S.; Investigation, J.S., J.O. and R.S.-I.; Methodology, J.S., P.J.F., J.O., R.S.-I. and A.F.S.; Project Administration, J.S. and A.F.S.; Resources, J.S., P.J.F., J.O. and A.F.S.; Software, P.J.F. and J.O.; Supervision, J.S. and A.F.S.; Validation, P.J.F. and J.O.; Visualization, J.S., P.J.F. and R.S.-I.; Writing—Original Draft, J.S., P.J.F., J.O. and R.S.-I.; Writing—Review and Editing, J.S., P.J.F., J.O., R.S.-I. and A.F.S.

Funding: This research was funded by the European Commission, under the projects 5GINFIRE (Grant No. 732497) and 5G-MOBIX (Grant No. 825496), by the Spanish Ministry of Economy and Competitiveness, under the project PERSEIDES (Grant No. TIN2017-86885-R) and the Ramon y Cajal Program (Grant No. RYC-2017-23823), and by the BBVA Foundation, under the 2018 Leonardo Grant for Researchers and Cultural Creators.

Acknowledgments: The authors would like to thank 5GINFIRE partners for their technical support during the execution of experiments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

4G	Fourth Generation (mobile/cellular networks)
5G	Fifth Generation (mobile/cellular networks)
API	Application Programming Interface
C-ITS	Cooperative ITS
CoAP	Constrained Application Protocol
GPS	Global Positioning System
CPU	Central Processing Unit
ETSI	European Telecommunications Standards Institute
HD	Hard Disk
HSPA	High-Speed Packet Access
IEEE 802.11 OCB	IEEE 802.11 running outside the context of a basic service set
IPv6	Internet Protocol version 6
IoT	Internet of Things
IoV	Internet of Vehicles
ITS	Intelligent Transportation Systems
MANO	Management and Orchestration
MEC	Multi-Access Edge Computing
NFV	Network Function Virtualization
OBD	On-Board Diagnosis
OSM	Open Source MANO
OBU	On-Board Unit
RAM	Random Access Memory
REST	Representational State Transfer
RTT	Round-Trip delay Time
SURROGATES	Virtual OBUs and Hybrid Communications to Foster 5G Vehicular Services
SDN	Software Defined Network
TCP	Transport Control Protocol
UDP	User Datagram Protocol
VIM	Virtualized Infrastructure Manager
VNF	Virtualized Network Function
vOBU	Virtual OBU

References

1. Meneguette, R.I.; De Grande, R.E.; Loureiro, A.A.F. *Intelligent Transport System in Smart Cities: Aspects and Challenges of Vehicular Networks and Cloud*, 1st ed.; Urban Computing, Springer International Publishing: Cham, Switzerland, 2018. [CrossRef]
2. Talib, M.S.; Hussin, B.; Hassan, A. Converging VANET with Vehicular Cloud Networks to reduce the Traffic Congestions: A review. *Int. J. Appl. Eng. Res.* **2017**, *12*, 10646–10654.

3. Katsaros, K.; Dianati, M. A conceptual 5G vehicular networking architecture. In *5G Mobile Communications*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 595–623.
4. Kljaić, Z.; Škorput, P.; Amin, N. The challenge of cellular cooperative ITS services based on 5G communications technology. In Proceedings of the 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 30 May–3 June 2016; pp. 587–594. [CrossRef]
5. Benslimane, A.; Taleb, T.; Sivaraj, R. Dynamic Clustering-Based Adaptive Mobile Gateway Management in Integrated VANET — 3G Heterogeneous Wireless Networks. *IEEE J. Sel. Areas Commun.* **2011**, *29*, 559–570. [CrossRef]
6. Remy, G.; Senouci, S.; Jan, F.; Gourhant, Y. LTE4V2X—Impact of high mobility in highway scenarios. In Proceedings of the Global Information Infrastructure Symposium—GIIS 2011, Da Nang, Vietnam, 4–6 August 2011; pp. 1–7. [CrossRef]
7. Chen, L.W.; Peng, Y.H.; Tseng, Y.C.; Tsai, M.F. Cooperative Sensing Data Collection and Distribution with Packet Collision Avoidance in Mobile Long-Thin Networks. *Sensors* **2018**, *18*, 3588. [CrossRef] [PubMed]
8. Gerla, M.; Lee, E.; Pau, G.; Lee, U. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea, 6–8 March 2014; pp. 241–246.
9. Kaiwartya, O.; Abdullah, A.H.; Cao, Y.; Altameem, A.; Prasad, M.; Lin, C.; Liu, X. Internet of Vehicles: Motivation, Layered Architecture, Network Model, Challenges, and Future Aspects. *IEEE Access* **2016**, *4*, 5356–5373. [CrossRef]
10. Santa, J.; Fernández, P.J. Seamless IPv6 connectivity for two-wheelers. *Pervasive Mob. Comput.* **2017**, *42*, 526–541. [CrossRef]
11. Aliyu, A.; Abdullah, A.H.; Kaiwartya, O.; Cao, Y.; Usman, M.J.; Kumar, S.; Lobiyal, D.K.; Raw, R.S. Cloud Computing in VANETs: Architecture, Taxonomy, and Challenges. *IETE Tech. Rev.* **2018**, *35*, 523–547. [CrossRef]
12. Jabbarpour, M.R.; Marefat, A.; Jalooli, A.; Zarrabi, H. Could-based vehicular networks: A taxonomy, survey, and conceptual hybrid architecture. *Wirel. Netw.* **2017**. [CrossRef]
13. Pelzl, J.; Wolf, M.; Wollinger, T. Virtualization technologies for cars: Solutions to increase safety and security of vehicular ECUs. *Automot. Saf. Secur.* **2008**, *2008*, 164–173.
14. Zhu, M.; Cao, J.; Cai, Z.; He, Z.; Xu, M. Providing flexible services for heterogeneous vehicles: An NFV-based approach. *IEEE Netw.* **2016**, *30*, 64–71. [CrossRef]
15. Baron, B.; Campista, M.; Spathis, P.; Costa, L.H.M.; de Amorim, M.D.; Duarte, O.C.M.; Pujolle, G.; Viniotis, Y. Virtualizing vehicular node resources: Feasibility study of virtual machine migration. *Veh. Commun.* **2016**, *4*, 39–46. [CrossRef]
16. El-Sayed, H.; Chaqfeh, M. The deployment of mobile edges in vehicular environments. In Proceedings of the 2018 International Conference on Information Networking (ICOIN), Jeju Island, Korea, 31 January–2 February 2005; pp. 322–324.
17. Emara, M.; Filippou, M.; Sabella, D. MEC-Assisted End-to-End Latency Evaluations for C-V2X Communications. In Proceedings of the 2018 European Conference on Networks and Communications (EuCNC), Ljubljana, Slovenia, 18–21 June 2018; pp. 1–9.
18. Kumar, N.; Zeadally, S.; Rodrigues, J.J.P.C. Vehicular delay-tolerant networks for smart grid data management using mobile edge computing. *IEEE Commun. Mag.* **2016**, *54*, 60–66. [CrossRef]
19. He, Y.; Zhao, N.; Yin, H. Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach. *IEEE Trans. Veh. Technol.* **2018**, *67*, 44–55. [CrossRef]
20. Campolo, C.; Molinaro, A.; Iera, A. A reference framework for social-enhanced Vehicle-to-Everything communications in 5G scenarios. *Comput. Netw.* **2018**, *143*, 140–152. [CrossRef]
21. Al-Badarneh, J.; Jararweh, Y.; Al-Ayyoub, M.; Fontes, R.; Al-Smadi, M.; Rothenberg, C. Cooperative mobile edge computing system for VANET-based software-defined content delivery. *Comput. Electr. Eng.* **2018**, *71*, 388–397. [CrossRef]
22. Ojanperä, T.; Mäkelä, J.; Mämmelä, O.; Majanen, M.; Martikainen, O. Use Cases and Communications Architecture for 5G-Enabled Road Safety Services. In Proceedings of the 2018 European Conference on Networks and Communications (EuCNC), Ljubljana, Slovenia, 18–21 June 2018; pp. 335–340.

23. Rapone, D.; Quasso, R.; Chundrigar, S.B.; Talat, S.T.; Cominardi, L.; la Oliva, A.D.; Kuo, P.; Mourad, A.; Colazzo, A.; Parmeggiani, G.; et al. An Integrated, Virtualized Joint Edge and Fog Computing System with Multi-RAT Convergence. In Proceedings of the 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Valencia, Spain, 6–8 June 2018; pp. 1–5.
24. Iovanna, P.; Pepe, T.; Guerrero, C.; Moscatelli, F.; Ksentini, A.; Chiasserini, C.F.; Casetti, C.; Mangues-Bafalluy, J.; Valcarenghi, L.; Martini, B.; Li, X.; Zennaro, G. 5G mobile transport and computing platform for verticals. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain, 15–18 April 2018; pp. 266–271. [CrossRef]
25. Fernandez, P.J.; Santa, J.; Pereñiguez, F.; Skarmeta, A.F. Towards seamless inter-technology handovers in vehicular IPv6 communications. *Comput. Stand. Interfaces* **2017**, *52*, 85–96. [CrossRef]
26. Santa, J.; Sanchez-Iborra, R.; Rodriguez-Rey, P.; Bernal-Escobedo, L.; Skarmeta, A.F. LPWAN-Based Vehicular Monitoring Platform with a Generic IP Network Interface. *Sensors* **2019**, *19*, 264. [CrossRef] [PubMed]