IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 11, 2019, accepted December 29, 2019. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.2965258

Offloading Edge Vehicular Services in Realistic Urban Environments

KATJA GILLY[1], (Member, IEEE), ANASTAS MISHEV[2], (Member, IEEE), SONJA FILIPOSKA[2], AND SALVADOR ALCARAZ[1]

[1]Department of Computer Engineering, Miguel Hernandez University in Elche, 03202 Alicante, Spain
[2]Department of Computer Networks and Information Systems, Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, Skopje 1000, North Macedonia

Corresponding author: Katja Gilly (katya@umh.es)

This work was supported by a grant from "Development cooperation funds 2018-Miguel Hernandez University" under Agreement UMH-GV. This research was performed in cooperation with the Institution.

ABSTRACT The imminent deployment of 5G and the rapid development of multi-access edge computing standards are demanding advances in terms of vehicular low latency offloading design and modelling proposals. In this paper we describe the functionalities of a high-level multi-access edge computing orchestrator that arranges location based vehicular edge services by the means of hierarchical dynamic resource management. In this way low latency responses can be guaranteed due to the geo-aware and energy efficient service allocation and dynamic migration. The first steps towards the definition of a vehicle to infrastructure communication specification are also provided. We study the efficiency of our proposal applied to an infotainment case study deployed in the city centre of Alicante, Spain. The simulation results obtained show that latencies perceived by vehicles generally range from optimal to first order sub-optimal scales all over the coverage area and that the presented offloading solution energetically scales with the number of hosts at the edge.

INDEX TERMS Energy efficiency, location based services, migration, multi-access edge computing, orchestration, vehicular low latency offloading.

I. INTRODUCTION

The automotive industry is experiencing significant changes in the last years with the advances in communication networks that have opened the door to a whole new range of revolutionary services including autonomous driving and vehicle platooning [1]. With the rise of 5G networks and their ability to support different service requirements by introducing network slicing within the programmable infrastructure, the proliferation of vehicle-to-everything services (where vehicles exchange data with each other, the infrastructure or any other communication entity) has literary exploded enabling not just an enhanced safety, but also a vast support for different types of infotainment on the road. In essence, the new connected vehicles have been transformed into platforms that offer innovative services aiming to enrich the users' experience [2].

To be able to fully use the potential of the next-generation in-vehicle experiences, service requirements heavily rely on the promises of the 5G ecosystem: ultra low latency, high reliability, very high data rates [3]. In addition, since the amount of data that needs to be processed in real time is massive, there is also a need to be able to offload (at least part of) the computing to high-performance computing entities that will supplement the vehicular computing power whenever that is needed. Thus, in addition to the traditional low/high requirements imposed on 5G communications, a support for Multi-access Edge Computing (MEC) [4] is a necessity for many of the newly developed vehicle use cases such as: augmented reality, see-through displays, in-car entertainment, remote vehicle analytics, etc.

The MEC platform is an overlay of a distributed set of servers that are co-located with the 5G base stations, that is reachable via the vehicle to infrastructure (V2I) communication [5]. The processing power of these servers can be accessed through a virtualisation layer and used to augment vehicles' local computing capabilities. By connecting to the edge computing facilities, vehicles are able to support demanding services while not sacrificing the low latency requirement as it is the case of cloud computing. Recently, the automotive industry and research communities have both recognised the benefits of extending vehicular systems with MEC capabilities, which have led to a fast growing body of

The associate editor coordinating the review of this manuscript and approving it for publication was Yan Huo.

knowledge that aims to solve the problem of efficient and reliable use of available MEC resources for different vehicular use-case services [6]. Due to the high rate of changes in the dynamic vehicular environment, developed approaches must take into consideration the vehicles mobility and density when deciding on resource provisioning. The system must be able to adapt to the mobility by ensuring that the hosted services are always as close to the vehicle that requested them as possible, while finding a way to cope with the growing density of vehicles in an urban scenario.

In this paper, we investigate a potential solution for the problem of maintaining low latency between the hosted service in the MEC fabric and the corresponding vehicle that roams throughout the provider network. We build upon the proposed MEC software architecture by ETSI [7] and define the communication patterns and algorithms implemented in its modules. The main goal is to define an approach for resource allocation and migration for offloading services in the MEC layer that will have twofold benefits: efficient resource usage by employing server consolidation and balancing techniques and continuous low latency between the service and the vehicle based on a follow-me triggered migration of services. The effectiveness of the proposed solution is then analysed for various scenarios of vehicle mobility and density in an urban environment. The case study used for the analysis is the centre of the city of Alicante in Spain, with tourist infotainment service embedded in the vehicles. In order to realistically represent the vehicles mobility, all traffic is simulated using the SUMO urban traffic simulator, while all modules from the proposed MEC system level management (the multi-access edge orchestrator) are implemented in the CloudSim simulator.

The rest of this paper is organised as follows: in the next section we discuss related work on the topic of offloading vehicular services to the MEC, then in section III, we provide a description of our low latency offloading proposal with a description of the MEC orchestrator proposal together with a V2I communication specification and definition of the resource management procedures. Section IV describes the case study scenarios implemented in CloudSim and SUMO, and the following section discusses the results obtained from the case study analysis. The final section concludes the paper.

II. RELATED WORK

The rapid development of several breakthrough technologies, such as 5G, edge computing, network virtualisation and software defined networking have strong influence on many industries, with the vehicular industry being one of the most prominent. New applications and use cases arise constantly as these technologies provide solutions for many of the challenges in automated or assisted driving, smart cars and roads, vehicle to vehicle and vehicle to infrastructure communication, etc. More and more of these use cases and applications require agile computing availability, low latency, massive bandwidth etc. According to [6], it is estimated that there will be 152 million actively connected cars on the road by 2020, and an average car will produce up to 30 TB of data each day.

To address the issue of offloading services toward the supporting infrastructure, authors in [1] propose a hierarchical deployment of the edge computing environment, enabling close placement of the resources and low latency of the communication, providing simultaneous access to specialised hardware-accelerated services, as close as possible to the moving vehicles. Task offloading and resource allocation are also studied in [8], where authors study, on one hand, the resource allocation problem with fixed task offloading decision and, on the other hand, the task offloading problem by optimising the uplink power allocation and the computing resource allocation, to finally propose an heuristic algorithm that computes task offloading solutions close to the optimal.

In [9], authors discuss the mobility problem from the point of view of the several different use-cases that require significant processing power and low latency of the communication at the same time. Such use cases include dynamic video processing and real-time interaction, including the application of image-aided navigation, natural language processing, and interactive gaming that represent significant challenges on the placement of the resources and their proximity to the moving vehicles, presented in a simulation environment. However, the authors consider only task-file mobility, which is not a general case and can not be applied to all scenarios. The architecture proposed by [10] includes roadside fog nodes that are essential for providing richer set of capabilities with lower latency. The authors do not tackle the problem of keeping the processing close to vehicles through migrations while they are moving. The notion of vehicular neighbour groups is introduced in [11], as a group of neighbouring vehicles. They are supported by MEC elements in the control plane, enabling higher processing power than the one available at the vehicles themselves. Moving vehicles will require migrations within the control plane, keeping the delay at minimum, but similarly to the previous research, the authors consider task scheduling and resource allocation to provide computing with low latency. The idea is explored in more depth in [12] through the Smart Collaborative Vehicular Network (SCVN) architecture, where frequent handovers are identified as one of the most important problems of the fast moving vehicles. The correlation between the handovers and migrations puts the same importance toward the solving of the problem the migrations themselves. Mobility of the process states of the computing in the MEC is discussed as one of the most important requirements in [13], enabling offloading compute-intensive applications and receiving timely results while constantly keeping the processing close to the user equipment, which can only be achieved through migration. In [14], a survey of IoV system architectures is presented, along with a comparison of Cloudlet, MEC and Fog architectures as necessary building blocks. The proximity and mobility of the computing are crucial parameters in their analysis, combined with the need for agility to clearly identify the migrations as the solution. Authors in [15] use statistical modelling for

computation capabilities as random variables. Their resource allocation policy is designed by considering the vehicle's mobility and the hard service deadline constraint. In [16], authors propose that pre-processing or data-reduction can be delegated to the less powerful edge nodes, distributed between the vehicles themselves and the fog computing nodes that are close to the vehicles. Such a distribution of processing again stresses the importance of an agile and flexible migration mechanism. The authors in [17] stress the importance of the support for the migration of the edge server to be available for use by nearby mobile devices. The proposed platform describes the migration procedure in details, but lacks of the discussion on the resources allocation and management.

While dynamic resource management targets toward lowering the latency, offloading the processing requirements from the vehicles to the infrastructure can be identified as a common point of interest in all mentioned related works. One approach in addressing this could be through orchestrating migrations of computing capacities in order to keep them as close to vehicles as possible to maintain a low latency. To incorporate orchestrators that will serve these goals into the established concepts like MEC complements its functionalities with software enhancements, based on already proved concepts and algorithms in cloud computing data centres. This paper draws from these concepts and discusses the functionalities of such a MEC orchestrator, which, along with the resource management procedures needed to achieve the low latency offloading, mitigates the previously identified open issues.

III. LOW LATENCY OFFLOADING PROPOSAL

Vehicular networks need the support of computation offloading techniques that cope with the extra computational resources, apart from storage space and networking facilities, required and demanded from the applications that are running on connected devices. The infrastructure that supports vehicular offloaded services must be rather local in order to maintain the latency obtained to the minimum possible response time. It is therefore recommended, when considering MEC as the solution to reduce latencies and increase bandwidth, to allocate services in the closest MEC micro-datacentre (co-located group of MEC hosts). This must be implemented in accordance with individual application demands or service group demands, thus encompassing services that actively interact among themselves, effectively leading to a location that simultaneously optimises the communication distance among related services and the vehicular user equipment.

The network infrastructure supporting a vehicular network will cover a wide physical area and it will be, therefore, designed based on complex network topologies in order to provide the shortest possible connection distance between two nodes. However, the choice of the location of the MEC host where the MEC service is provided is important when aiming to reduce latency to the vehicle's connected device to a minimum. We have adapted Filiposka *et al.* two level hierarchical framework [18], that was originally designed for cloud computing infrastructures to edge computing solutions, implementing a mobile-aware dynamic resource management [19] that also includes a minimum latency "follow me" proposal based on applications/services fast migration [20] as a mobile device moves around the coverage area. Considering a vehicular network, we continue this work while going some steps further in the definition of the MEC architecture and detailing the structure, operation and interaction of the necessary MEC modules/functionalities in order to provide dynamic and low latency MEC vehicular services.

We cover in this section, firstly, the building blocks definition of our MEC solution proposal to, secondly, propose a general V2I communication specification and apply it to an example of vehicular service migration pattern interaction between two different MEC platforms.

A. MEC SOLUTION PROPOSAL

The MEC architecture is accessed through the network provider infrastructure by a Road Side Unit (RSU), a cell tower or a base station and it is composed of several components whose general functions and coarse-grained module connections are already defined by the ETSI in its *Framework and Reference Architecture* document [21]. Among these components, there is a high level orchestrator named MEO (Multi-access Edge Orchestrator) defined as the core component of the MEC system level, that is responsible for providing an overall view of the MEC system by taking into account and organising the available resources, services and topology at the mobile edge host level. This includes a close relation of the MEO with the different MEC micro-datacentres that are composed of MEC hosts where the demanded services and applications run in a virtualised fashion. The MEO is also responsible for the integrity and authenticity of application packages and the validation of the necessary rules and requirements in compliance with the operator policies. In essence, the MEO is the component in charge of all operations related to the management of the global pool of virtualised resources of the complete MEC system that run vehicular services, beside other MEC services. The management of applications/services, virtual resources and hosts is done via the communication between the MEO and the MEC platform manager of each MEC host that deals with the local issues on the host level.

The ETSI MEC architecture [21] is a high-level architecture that defines the general components necessary to enable MEC services: the MEO and the MEC platform manager that sit on top of the virtualised infrastructure. The document also describes the requirements and global functionalities that need to be provided by these components, but does not provide a detailed design or implementation. We have illustrated these general components in Fig. 1, and in addition we define the minimum number of required modules that we consider essential for the MEO's functionalities in terms of the MEC's services management. These modules are listed and shortly described in Table 1.

TABLE 1. Modules description of the mobile edge orchestrator.

Module	Functionality
Location and mobility tracking module	tracks the location of the VUE requesting the service.
VM management module	manages MEC virtualised resources on a global level.
Security and identity module	provides the needed features for the authentication of the VUE and user identity confidentiality.
QoS monitoring module	monitors level of service provided and the resource usage of each service and VUE.
Cost analysis module	calculates the cost associated to the resource usage of each service and VUE.
Communication service module	in charge of the interaction of the MEO with the VUEs.

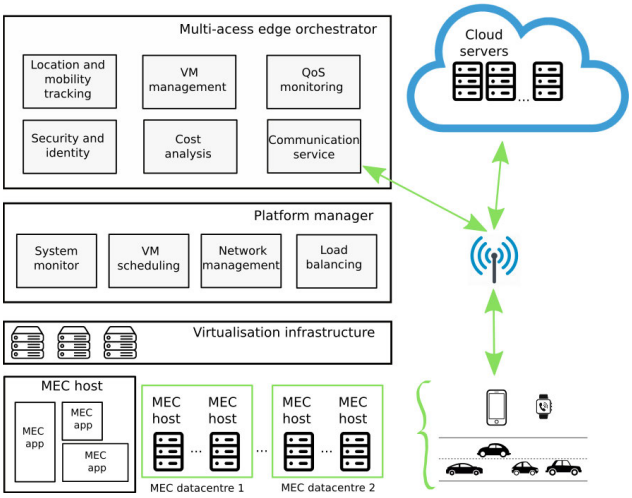


FIGURE 1. MEC architecture components and modules of interest.

On one hand, we define a *Communication service* module as the responsible for the communications with the vehicular user device application that requests an application by asking for its instantiation with the necessary credentials that authenticates the user as valid and authorised to use a specific service whose image has to be already in a record of on-boarded packages or can be directly accessed from the MEO through a link to the application package.

We have incorporated a *Location and mobility tracking* module that detects the location of the newly connected devices entering in the coverage area, tracks the mobility of the VUE (Vehicle User Equipment) while the service is provided and communicates the departure of a VUE to the *Virtual Machine (VM) management* module when the device leaves the coverage area. The *Security and identity* module covers the needed requirements to authenticate the user of the VUE application and provides the necessary features for a trustworthy service provision.

The allocation of the service requested from the MEC system is performed optimally by the *VM management* module either if it is a new VUE service or in case it is a migration of a service that was already allocated somewhere else in the MEC system. This module runs the algorithms for global allocation and migration of services, where the main goal is to select the MEC micro-datacentre and host where the computing virtualised resources are going to be allocated for the requested vehicular application in a form of a VM or a container. Thus, this module is very much dependent on fresh and accurate information from MEC platform managers.

The decision on where to allocate a new service request, or where to migrate an already running service in order to obtain the minimum possible latency within the virtualised platforms created on the MEC hosts can be done using specialised optimisation algorithms. Within the proposed solution, the implemented algorithms are based on a hierarchical view of the overall network provider infrastructure, where each MEC micro-datacentre represents a local community or neighbourhood of virtual resources, that are joined together into a multi layered hierarchical tree based on their proximity. One of the tasks of the VM management module is to build and maintain this tree that is used as a reference during the resource management process. When deciding where to place or migrate a service, the main goal is to choose the local community that is as close to the requesting user equipment as possible so that the minimum latency requirement is achieved. Thus, the hierarchical communities tree is searched for available resources from the bottom up. Once the available resources are found within a group of MEC hosts, the final decision on the resource usage is done considering either load balancing or server consolidation. In this way, in addition to the minimum possible latency, performance and energy efficiency are also considered during the decision process. The pseudo code for both algorithms, for placement and migration, can be found in [19].

The Quality of Service (QoS) level provided is constantly monitored by the *QoS monitoring* module that collects its data from the platform monitor that have direct access to the virtualisation infrastructure manager of the MEC host in order to get reliable information about the usage of the virtual resources assigned to each service. The module is able to raise an alarm if the QoS level provided does not correspond to the SLA rules defined for the given MEC service, such as low latency. The *Cost analysis* module works together with the *VM management* module and computes the cost of each service provided by the MEC system.

B. V2I COMMUNICATION SPECIFICATION

In this section we cover the first steps towards the definition of the communication protocol needed to connect online services to MEC offloading facilities that are allocated in a vehicular network.

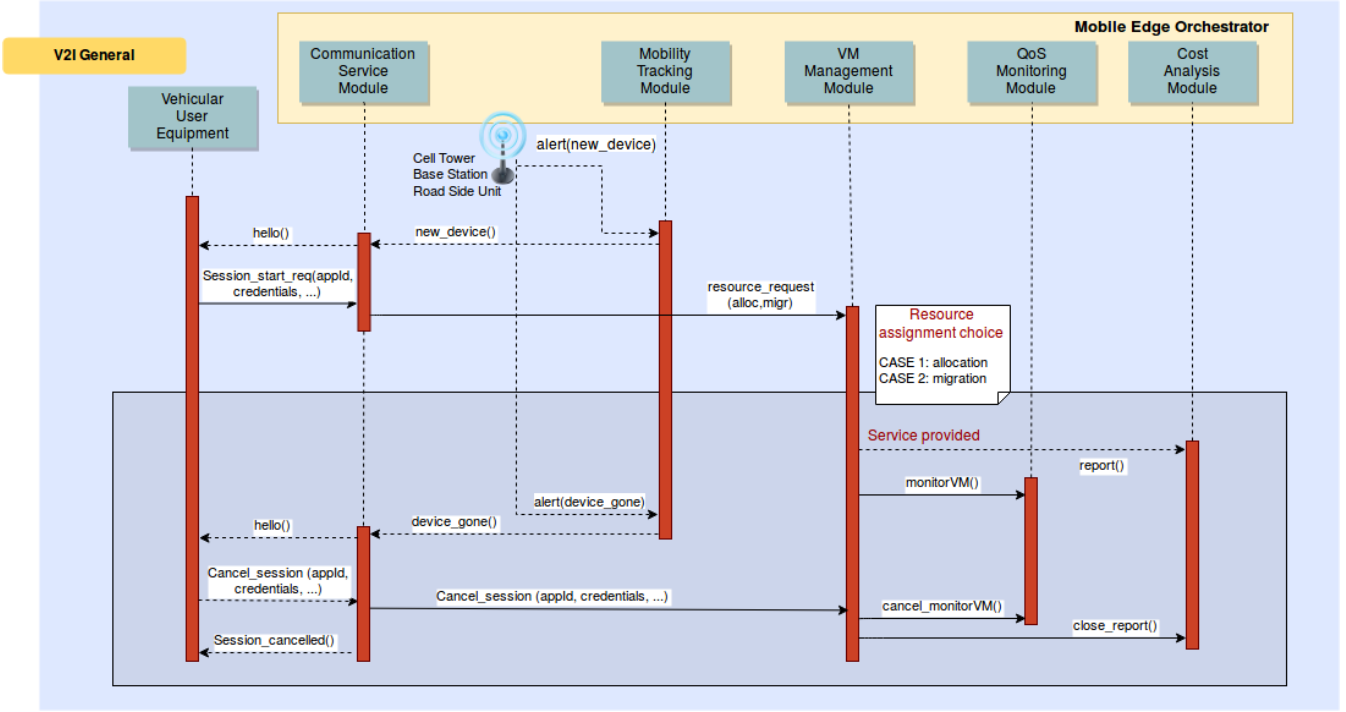


FIGURE 2. V2I general.

Whenever a new VUE accesses the MEC-enabled coverage area served by a MEO, the *Communication Service* module is the one in charge to contact the new vehicular device. The detection of the potential new client is done via the access point to the network (ie. the cell tower, a base station or a RSU) that goes through the *User Application LifeCycle Management Proxy (UALCMP)*, as it is defined in [22], and then arrives to the MEO. The orchestrator then commands the launch of a specific tracking process instance for that VUE in the *Location and Mobility Tracking* module. It also informs the *Communication Service* module that either a session start command with the identification of the application requested (that will require an interchange of messages that provides user credentials and all details needed to start the service) or a query of the available MEC applications will be received. We have graphically described this communication process between the VUE and the MEO in Fig. 2 omitting the modules and entities that do not affect the control plane, that is for example the MEC host that provides the service and that belongs to the data plane of the system, or the UALCMP whose functionality is transparent in the process we are modelling. Although the instantiation of a VUE application can also be done through the Operations Support System (OSS), it would not change the procedure more than just skipping the interaction with the *Location and Mobility Tracking* module until a VUE asks for the application. The rest of the procedure would be the same.

The *Communication Service* module maintains a constant connection to the VUE application by sending him periodical high level protocol messages while the network connectivity is controlled by the access point that will inform the *Location and Mobility Tracking* module when it detects the VUE has left the coverage area.

There are therefore two invariants when dealing with the specification of the communication protocol that need to be taken into account: on one hand, the application requested from the VUE should be active during the whole process and, on the other hand, the VUE should not go out of the MEO coverage area during the communication process. In order to detail these two conditions, we define the operations *req* and *resp* as the request and response operations, respectively, and τ as a value similar to the Round Trip Time (RTT) between two communicating processes X and Y that will be established as the time-out limit. We have formalised the following expression that should be fulfilled during the whole V2I communication process by following Algebra of Communicating Processes (ACP) terminology for timed transition systems [23], where the binary operators $+$ and $\cdot$ represent an alternative and sequential operation, respectively, and notation $a^c t$ denotes an action a that takes place at time t. Therefore, $\sum_{t:\mathbb{R}} \text{req}^c t$ invokes the alternative composition of any request that arrives at time $t \in \mathbb{R}$. The use of the conditional operator $c \rightarrow p \diamond q$ expresses "if c then p else q", being the sequential operation of previous summation expression either a response from the VUE in a time instant lower than RTT or a timeout.

$$\sum_{t:\mathbb{R}} \text{req}^c t \cdot \left[\sum_{u:\mathbb{R}} (u \leq \tau) \rightarrow \text{resp}^c u \diamond \text{timeout}^c u \right]$$

This formalisation ensures then that any time a request arrives during the interval $[0, t], \forall t \in \mathbb{R}$, another action in time $u > t, \forall u \in \mathbb{R}$, will follow that will be either a

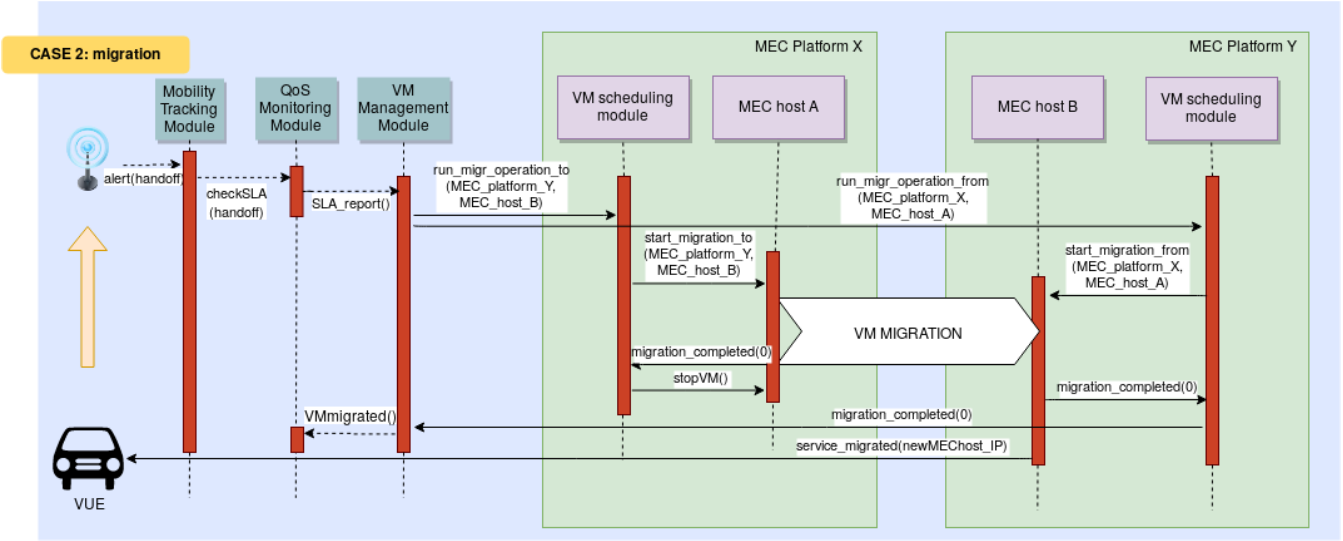


FIGURE 3. Communication process between MEC micro-datacentres that are performing a service migration.

response if $u \leq \tau$ or a time-out, otherwise. In this way we ensure that no matter what external action the communication protocol receives, there is always a response in the form of well defined actions. In other words, for every received input, the implementation of the rules guarantees the protocol consistent behaviour.

The *VM Management* module, that handles the global management of virtualised resources, is informed of the requested service requirements. High level allocation and migration algorithms are implemented in this module, and are called to find the optimal MEC micro-datacentre and host to allocate the demanded service. In Fig. 2 we have summarised both cases in this process, considering the allocation algorithm is called when a new service starts and the migration algorithm is run when the service was already attended by other MEC host also ruled by the same MEO and has to be migrated to a closer MEC host. Once the service is provided, the *Cost Analysis* module computes the associated cost of the used MEC resources by polling for the resource usage details from the MEC micro-datacentre that hosts the service while the *QoS Monitoring* Module continuously tracks the status of the Service Level Agreements (SLA).

Once the service is provided, the VUE can either receive a connection start that comes directly from the MEC host where the service is allocated, or the vehicular client application discovers the MEC application instance via DNS look-up. We have not given evidence of this process in Fig. 2 in order to keep it clear and simple.

We have exemplified a special case of VMs migration in the message sequence chart of Fig. 3. We consider the VUE has moved to a coverage area governed by a different radio end-point attached to a different branch of the MEC network infrastructure. In this case, the MEO will try to find a suitable MEC micro-datacentre and host to migrate the associated VM to. In order to do this, it will search for a closer MEC micro-datacentre by running the migration algorithm (case 2 in Fig. 2), thus aiming to reduce the latency between the VUE and the VM that provides the service. The *VM Management* module is the responsible module to find the best destination MEC host and establishes a direct communication with both MEC host levels to inform them about the need to handle the migration. The *VM Scheduling* modules of both the original and destination MEC micro-datacenters are involved in the migration operation from the original MEC host to the destination one. This operation is performed autonomously by both host levels following the instructions of each *MEC Platform Manager (MEPM)* that commands the necessary operations of resource visualisation to the *Virtual Infrastructure Manager (VIM)* in order to migrate the application. The *VM Management* module is informed of the migration operation result. The MEC host where the service is now running informs the VUE that the service has been migrated and provides its own IP address in order to establish a new data service connection between the VUE and itself. Please note that neither the MEPM nor the VIM are represented in Fig. 3 but are defined as entities of the MEC host level in the multi-access edge system reference architecture [21].

We do not intend to cover all cases involving MEC micro-datacentre communications in this paper. By presenting these use cases we aim to acknowledge that an additional effort of the scientific community is required to work on the specifications of the communication protocols involved in vehicular MEC communication.

IV. RESULTS - A CASE STUDY

For the purposes of analysing the performances of the proposed low latency offloading proposal in the context of a realistic urban environment, a use case study was developed focusing on a specialised scenario. The aim of the use case is to setup an urban 5G environment that will support

a vehicular network with varying density that can use the advantages of a specific pilot edge service that works in real-time and has the requirements of minimum latency for analysis and streaming of multimedia content. Towards this goal, we have chosen to simulate a vehicular network case study that is using a tourist infotainment service based on augmented reality.

A. CASE STUDY DESCRIPTION

The location of the simulation scenario is chosen to be the city centre of Alicante, a Mediterranean city in the South East of Spain. The city centre area considered is 1.8 km wide per 2.0 km high and within, 9 different locations have been chosen for the placement of 5G base stations with a standard 200m radio range. The locations of the 5G base stations have been chosen from a list of well known tourist attractions in the analysed area, and the final location and number of base stations have been defined so as to have the whole city centre area covered with the signal with minimum overlaps based on the Voronoi cell technique. The whole urban area of interest has been imported into the simulation environment using the original OpenStreetMaps description.

The simulations scenario considers that all vehicles that enter the coverage area would connect to the infotainment system through the nearest base station. At this moment an edge service will be instantiated to serve the requesting vehicle and, as it moves around the coverage area, the assigned virtualised resources that provide the infotainment service would follow each vehicle (in order to minimise latencies) employing transparent dynamic migrations orchestrated under the control of the VM management module of the MEO. Migrations would be performed only if there are enough resources on the MEC micro-datacentre that is located in that radio range. Otherwise, the MEO would abort the migration leaving the connection to the actual MEC host to save the overhead produced by migrations in terms of energy, computation and networking since the resulting latency would not improve. Once the vehicle exits the coverage area, the corresponding infotainment service is cancelled and the resources are released for future use by other vehicular entities. In the described scenario, one service per vehicle is considered as it is taken into account that this service is part of the vehicle's integrated multimedia system.

The described scenario has been implemented by integrating two different simulators: SUMO [24] together with OpenStreetMaps in order to simulate the area and the vehicles movement, and Cloudsim [25] with a special extension (based on [26]) that supports the proposed mobile edge service architecture and the proposed algorithms within the described modules. The integration of the two simulators is implemented so that the output of SUMO represents input for CloudSim triggering the events of new service request, service migration analysis and service cancellation within the corresponding modules: location and mobility tracking and QoS monitoring. Fig. 4 illustrates this simulation process.

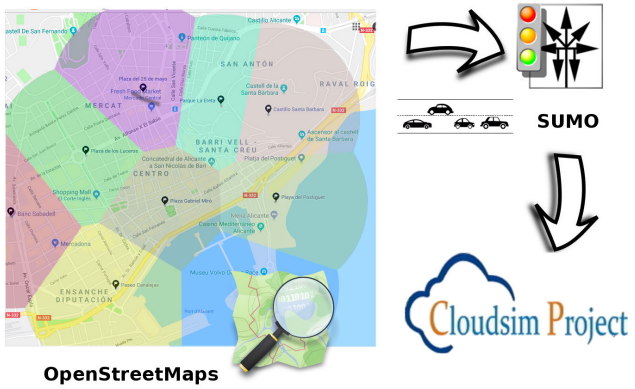


FIGURE 4. Simulation process description.

The SUMO simulator has been chosen because it excels in providing realistic simulations of urban traffic scenarios. By importing the OpenStreet map representation of the city centre of Alicante, the complete representation of the traffic rules is provided to SUMO including details such as: one-way streets, street lanes, traffic lights, speed limits, roundabouts and right of way. Thus, the random traffic generated using the internal traffic generators of SUMO adheres to all real-life traffic rules defined in the city. In addition, the simulator also takes into account vehicular behaviour such as acceleration and deceleration at the start and stop of a movement, as well as using the brakes whenever needed and dealing with traffic jams. The simulation parameters where defined in the way that all vehicles start their journey from a random point within the target area and move towards a predefined goal which can be inside or outside the coverage area of interest. Each vehicle must move for at least 2km within the area of interest before it is removed from the simulation scenario. A set of different simulation scenarios has been created by varying the density of vehicles in each simulation which is described with the number of vehicles generated per second parameter and the total number of vehicles and simulation time. The detailed output of the SUMO simulation is post-processed based on the additional information of the location of base stations. In this way, the available description of the location of each vehicle at specific time points from the SUMO trace is transformed into service information input for CloudSim:

- the initial moment the service is requested, corresponding to the moment the vehicle is generated and starts to move,
- the timestamp of handover that triggers service migration based on the base station coverage area according to the Voronoi assignment and
- the moment the service is terminated, corresponding to the vehicle exits the simulation area or stops within the simulation area.

As previously mentioned, this output is then provided as input to the CloudSim simulator, wherein it is used to schedule the creation and termination of services, and to tigger the call of the migration policies that will handle the handover events.

TABLE 2. Simulation input variables for SUMO and CloudSim.

simulation parameter	possible values
area size	1.8 km x 2.0 km -> Alicante's city centre
total number of base stations	9 -> 9 local communities of MEC hosts, 1 base station per community
total simulation time	10000 s
average number of vehicles	{49xx, 57xx, 69xx, 89xx}, where xx may differ in different simulation runs
total number of MEC hosts	{45, 63, 81, 99, 117} with {5, 7, 9, 11, 13} hosts per local community, correspondingly
MEC host type 1	1 - 4 CPU cores, 8 GB RAM, 1 Gb network
MEC host type 2	2 - 6 CPU cores, 12 GB RAM, 1 Gb network
average delay per link type	edge: 1.57 ms, aggregation: 2.45 ms, core: 2.85 ms
resources per infotainment service	2 CPU cores, 2 GB RAM, 100 Mbps network

The CloudSim simulator has been extended with additional functions that allow for the creation of a dynamic MEC environment. The MEO functions of interest have been implemented by extending the existing policies for VM allocation and VM migration. For the results presented in this paper the community based placement and migration algorithm in combination with load balancing have been used. For details on their implementation please refer to [19]. The complete MEC network set-up has been defined using a community based grouping of hosts that represents the hierarchical tree structure used by the algorithms to make the decision on choosing the MEC micro-datacentre and host for a given service. For each base station in the coverage area, a separate lowest level MEC hosts community is defined and interconnected with edge links creating the MEC micro-datacenters. These are then connected with aggregation and core network links via one or multiple switches/routers so that the multi-layered tree is created, with its root encompassing all available MEC hosts in the provider's network. An additional cloud data centre is connected to the tree root via a WAN link, to be used in the case when there are no available resources for hosting services in the MEC network. The simulation scenario description scripts have also been extended to support the dynamic nature of the creation and cancellation of services based on the output received from SUMO. The available virtualised resources are thus dynamically reassigned during allocation and live migration processes. For the purposes of supporting the computing intensive requests from the infotainment service, the resources are assigned using the space sharing method. The CloudSim simulator supports two types of MEC hosts that differ in the resources available and in the energy efficiency that have been used to analyse the energy consumption of the hosts during the simulations. For the delay calculations, the internal delay constants per link type as defined in CloudSim have been used. All simulation scenarios have been run multiple times and the averages of the results have been calculated.

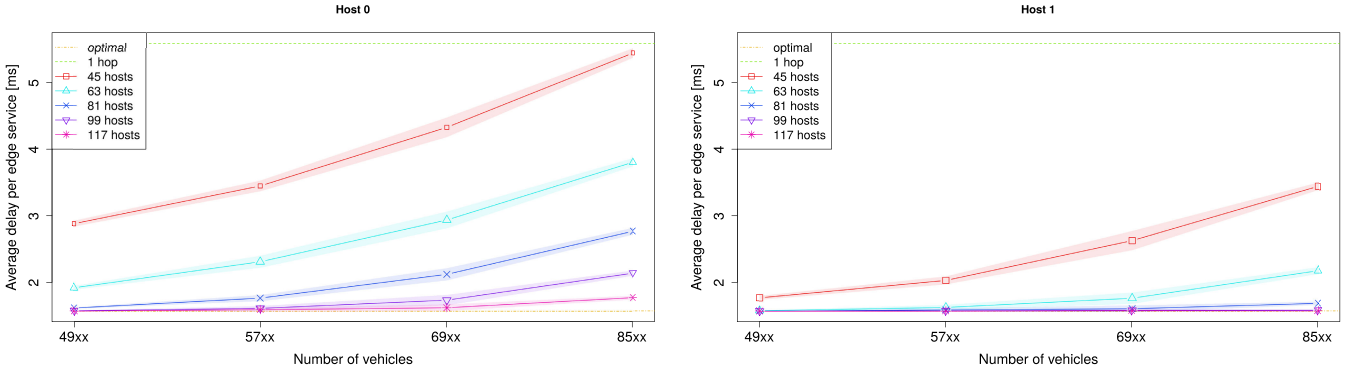
Table 2 summarises the variable scenario parameters used in the simulations.

B. RESULTS AND DISCUSSION

We analysed the performances of various aspects of the proposed MEC solution employed to host an infotainment service for a fleet of tourist vehicles in the described urban vehicular network scenario. The presented results focus on the performances of the MEC network, with the base station where the VUE is currently connected as entry point to the network and the VM where its corresponding service is hosted in the MEC infrastructure as the destination point. The main goal of the presented analysis is to provide insight in the potential of the proposed solution for the purposes of design of the MEC hosts distribution and locating the vulnerable spots in the coverage area where increased capacity is needed.

The effectiveness of the proposed solution compared to other more traditional approaches to resource management in terms of decisions on where to place the requested VMs hosting user services and when to migrate them have been analysed in [19], [27] and [**?**] wherein the pseudo code and detailed explanation of the resource management policies and their implementation are also discussed. The results in this paper, on the other hand, focus on a much more realistic urban scenario on a grander scale.

In Fig. 5 the averaged delay induced by the MEC network to the services that need to be migrated due to location change is presented together with the 95% confidence intervals. The results are provided for varying network load in terms of number of vehicles requesting a service. In order to provide a reference point for visual assessment of the performance two more lines are presented: 'optimal' and '1-hop'. The bottom 'optimal' line represents the perfect case of continuous minimum latency provided by the MEC network, which can only happen when during the whole time period all services are located on servers which are co-located with the currently used base station connected on the same edge switch. In other words, this is the network latency caused by the edge switch traversal which amounts to 1.5 ms based on the network constant values in CloudSim. The '1 hop' line at the top of the figure represents the latency induced by the MEC network in the case when the service is located



(a) MEC host type 1

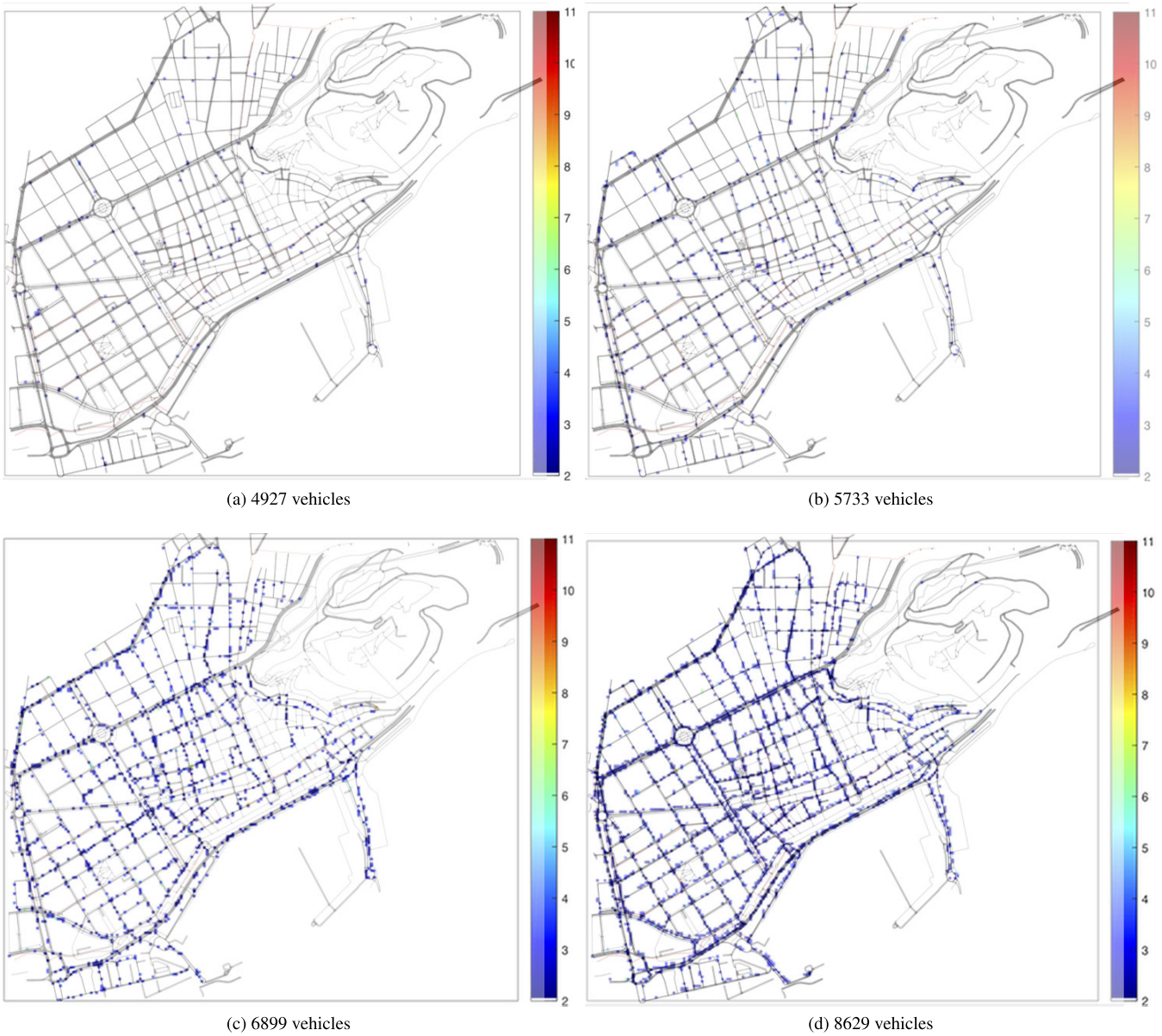
(b) MEC host type 2

FIGURE 5. MEC service to VUE latency for migrating services.

in a neighbouring MEC hosts community relative to the currently used base station. In order to reach the service VM from the base station, the traffic must pass through the edge switch, up to an aggregation switch, and then down to a different edge switch, which yields to the total amount of around 5.7 ms. Based on the structure of the hierarchical tree network that connects all MEC hosts, the highest number of hops that can be found between two edge points in the network is 3 hops, (going up to the core switches and then down to the edge) corresponding to a total of 11.5 ms delay. As expected, in the cases when the capacity of a single local community of hosts is higher (the case of using MEC hosts type 2), the average latency experienced is dropping due to the possibility to locate a larger number of services in the same local community. Please also note that this increased capacity alleviates the migration process since during the live migration process the service uses two sets of virtual resources: the initial resources at the old host, and the newly allocated resources at the new host. Only after the completion of the migration process, the initial resources are deallocated and freed for future use. The presented results also show that as the number of MEC hosts increases, the average delay drops and comes close to the optimal latency, that is offered to an increasing number of services throughout the simulation time. The worst performances are obtained for the minimum number of hosts (45) in a heavy loaded scenario with more than 8500 vehicles. When taken into account that these number of vehicles represents a situation of a traffic jam when reviewed in SUMO, it can be concluded that even this scenario copes well with the demands of the vehicular fleet.

In order to analyse how the events of non-optimal latency are spread throughout the coverage area, the obtained results from CloudSim have been combined with the information obtained from SUMO and plotted as an overlay on top of the OpenStreet map of the area. The results are presented in Fig. 6 where the dots represent events in CloudSim where suboptimal latency is detected (any latency higher than the optimal value). In order to obtain these results, all reported optimal latency events have been filtered out of the images. The colour of the dots represent the recorded latency according to the legend provided in ms. Please note that the upper right corner of the map represents a pedestrian only area as it is Benacantil hill, where stands one of Spain's largest mediaeval fortresses, Santa Barbara's castle, that is avoided by vehicles in all simulation scenarios except for the smaller number of streets given in darker shade. As it can be expected, the positions where non-optimal latencies are experienced increase in density as the load in the simulation increases (starting with 4927 vehicles in Fig. 6a and increasing up to 8629 vehicles in Fig. 6d). It is interesting to note that the distribution of the reported location is somewhat uniform across the coverage area, with no distinctive locations that would be considered as 'gray' areas, or dead-spots. Another observation should be made on the reported values for the experienced non-optimal latency. All scenarios represented in Fig. 6 report latencies that are very close to the 1 hop latency values (blue tones), compared to the worse case scenarios of 2 hops (yellow tones) and 3 hops (red tones) which are reported in less than 1% of the cases, and are very difficult to spot on the presented figures. The same type of analysis can be made using a higher threshold for latency during the filtering phase in order to easily spot the problematic reported latencies in the area. Our analysis has not found any particular pattern in the reported latencies other than temporary traffic increase in the area that has resulted in events of allocating sub-optimal resources for new service requests. All of these are then migrated to an optimal location once the vehicle reaches a new base station coverage area.

In Fig. 7 the percentage of service events where non-optimal latency is detected out of all events observed during the simulation time is presented using a 3D bar chart for the two different types of MEC hosts evaluated. The effects of designing a MEC infrastructure solution with increased capacity is evident when comparing the presented results. In this case, as given in Fig. 7b, the maximum percentage of events reaches around 65% in the worst case scenario of only 45 hosts coping to host over 8900 services. These are then rapidly dropping to around 45% when decreasing the number of services in the range of 69xx, and 35% when increasing the number of hosts to 63. It is clear that in most



(a) 4927 vehicles

(b) 5733 vehicles

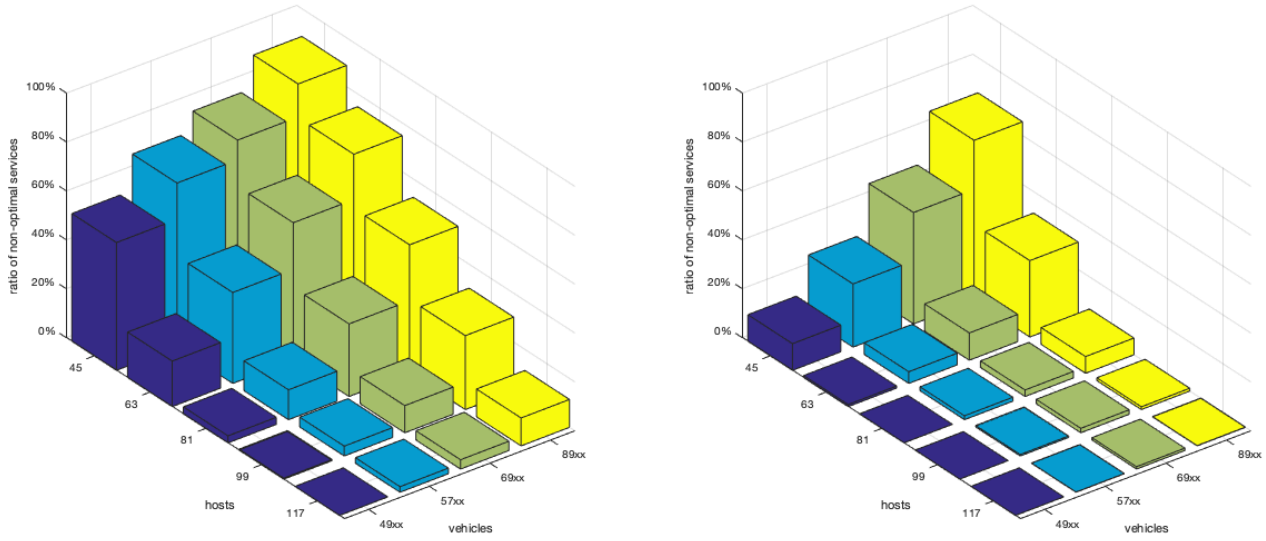
(c) 6899 vehicles

(d) 8629 vehicles

FIGURE 6. **Overlay of vehicles in the map of Alicante for a scenario with 45 MEC hosts of type 1.**

of the analysed scenarios (16 out of 20), the percentage of sub-optimal latency events are less than 10%. When combined with the fact that the provided resources ensure high performance computing in terms of not allowing CPU time sharing, these results clearly point to the high potential of the proposed resource management functionalities of the MEC orchestrator. Of course, results also show that careful planning is needed so that there are enough resources to answer customers needs. However, it must be noted that although the case of using MEC hosts of type 1 leads to an obvious increase in the number of sub-optimal latency events, the provided latency is still well under the 1-hop delay of 5.7 ms, as presented in Fig. 5a.

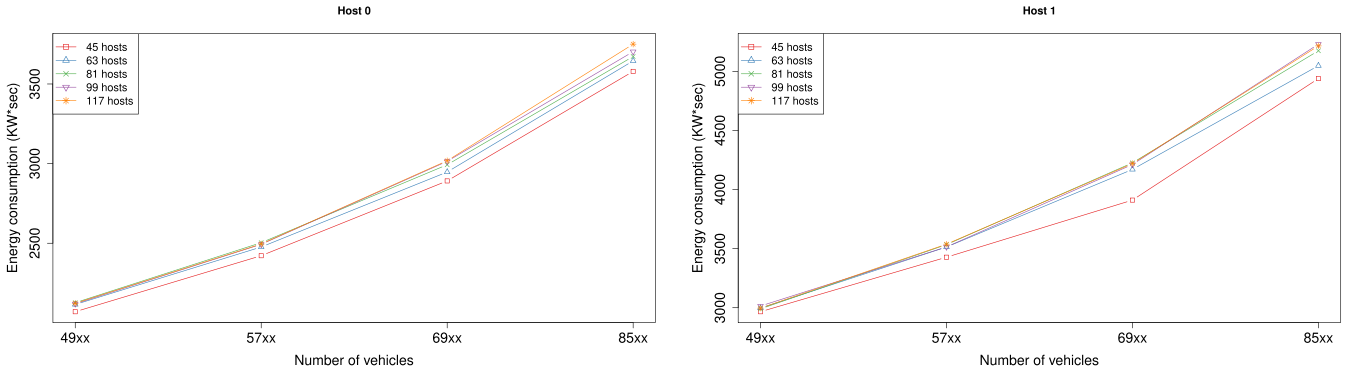
The presented results show that the performance in terms of provided latency by the MEC orchestrator for various loads in the simulation scenarios are complemented with the analysis of the energy efficiency aspect of the proposed solution. In Fig. 8 the overall energy consumption of the whole MEC network and hosts is represented for various number of vehicles and MEC hosts available. Please note that in all simulation scenarios the MEC hosts in each community are used in a consolidated approach that minimises the number of hosts needed to provide the allocated services. The hosts that are not used for any services are turned off in order to save on energy consumption. These hosts are brought online only when not enough resources can be provided by the already running hosts. The presented results show that the increasing number of hosts in the MEC system slightly affects the overall energy consumption with the difference being more pronounced in scenarios of high loads (85xx vehicles). On the other hand the change of host type has a more significant effect on the energy consumption. Based



(a) MEC host type 1

(b) MEC host type 2

FIGURE 7. **Percentage of MEC services provided with non-optimal latency.**



(a) MEC host type 1

(b) MEC host type 2

FIGURE 8. **Energy consumption of the MEC system.**

on the presented results, the amount of energy spent for the scenarios involving around 69xx vehicles when using MEC hosts of type 1 (about 3 MWs) is comparable to the amount of energy spent by MEC hosts of type 2 for the case of around 49xx vehicles. The increase in energy consumption between the two types of hosts moves from 15% up to 30% in the case of high loads. However, the smooth low increase with the number of hosts in both cases indicates that the solution can scale in terms of number of hosts without incurring a steep increase in the energy consumption.

V. CONCLUSION

This paper has covered several steps towards the design of an offloading solution for vehicular networks considering the MEC standard. On one hand, we provide the definition and functionality of a modular structure for a MEC orchestrator and the general lines of the control plane communication specification among the modules of the MEO and the vehicular user equipment, considering the description of the communication process for a service migration between two MEC micro-datacentres as a special case. The analysis of our proposal has been done using a case study of a tourist infotainment MEC service offered in the city centre of Alicante, Spain implemented by combining two simulation tools: SUMO for the vehicular entities and Cloudsim for the MEC environment. The presented results show that the resource allocation and migration technique implemented in the VM management module of the orchestrator provides a close to optimal performance in terms of delay time in most of the coverage area even when considering high load scenarios such as 8500 vehicles, and that the solution is energetically scalable with an increase in the number of hosts.

REFERENCES

[1] D. Sabella, H. Moustafa, P. Kuure, S. Kekki, Z. Zhou, A. Li, and C. Thein, "Toward fully connected vehicles: Edge computing for advanced automotive communications," 5GAA Automot. Assoc., White Paper, Dec. 2017. Accessed: Jan. 2020. [Online]. Available: http://5gaa.org/news/toward-fully-connected-vehicles-edge-computing-for-advanced-automotive-communications

[2] *V2X White Paper: NGMN Task Force*, NGMN Alliance, Frankfurt, Germany, 2018.
[3] *View on 5G Architecture, Version 3.0*, 5G PPP Architecture Working Group, Jun. 2019. Accessed: Jan. 2020. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2019/07/5G-PPP-5G-Architecture-White-Paper_v3.0_PublicConsultation.pdf
[4] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, K. Wen, K. Kim, R. Arora, A. Odgers, L. M. Contreras, and S. Scarpina, "MEC in 5G networks," ETSI, Sophia Antipolis, France, White Paper 28, 2018. Accessed: Jan. 2020. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf
[5] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6398–6409, Jul. 2018.
[6] SAS Institute. *The Connected Vehicle: Big Data, Big Opportunities*. Accessed: Apr. 2019. [Online]. Available: https://www.sas.com/content/dam/SAS/en_us/doc/whitepaper1/connected-vehicle-107832.pdf
[7] A. Reznik, R. Arora, M. Cannon, L. Cominardi, W. Featherstone, R. Frazao, F. Giust, S. Kekki, A. Li, D. Sabella, C. Turyagyenda, and Z. Zheng, "Developing software for multi-access edge computing," ETSI, Sophia Antipolis, France, ETSI White Paper no. 20, 2017.
[8] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.
[9] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive offloading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.
[10] C. Huang, R. Lu, and K.-K.-R. Choo, "Vehicular fog computing: Architecture, use case, and security and forensic challenges," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 105–111, Nov. 2017.
[11] X. Huang, R. Yu, J. Kang, Y. He, and Y. Zhang, "Exploring mobile edge computing for 5G-enabled software defined vehicular networks," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 55–63, Dec. 2017.
[12] P. Dong, T. Zheng, S. Yu, H. Zhang, and X. Yan, "Enhancing vehicular communication using 5G-enabled smart collaborative networking," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 72–79, Dec. 2017.
[13] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for vehicular communications," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 111–117, Jan. 2018.
[14] T. S. J. Darwish and K. Abu Bakar, "Fog based intelligent transportation big data analytics in the Internet of vehicles environment: Motivations, architecture, challenges, and critical issues," *IEEE Access*, vol. 6, pp. 15679–15701, 2018.
[15] L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10190–10203, Nov. 2018.
[16] Z. Lamb and D. Agrawal, "Analysis of mobile edge computing for vehicular networks," *Sensors*, vol. 19, no. 6, p. 1303, Mar. 2019.
[17] T. Kondo, K. Isawaki, and K. Maeda, "Development and evaluation of the MEC platform supporting the edge instance mobility," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, Jul. 2018, pp. 193–198.
[18] K. Gilly, S. Filiposka, and A. Mishev, "Supporting location transparent services in a mobile edge computing environment," *Adv. Elect. Comput. Eng.*, vol. 18, no. 4, pp. 11–22, 2018.
[19] S. Filiposka, A. Mishev, and K. Gilly, "Mobile-aware dynamic resource management for edge computing," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 6, p. e3626, Apr. 2019, doi: 10.1002/ett.3626.
[20] W. Lu, X. Meng, and G. Guo, "Fast service migration method based on virtual machine technology for MEC," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4344–4354, Jun. 2019.
[21] *Multi-Access Edge Computing (MEC); Framework and Reference Architecture*, Standard ETSI GS MEC 003 V2.1.1 (2019-01), ETSI Group Specification, 2019
[22] D. Sabella, V. Sukhomlinov, L. Trang, S. Kekki, P. Paglierani, R. Rossbach, X. Li, Y. Fang, D. Druta, F. Giust, L. Cominardi, W. Featherstone, B. Pike, and S. Hadad, "Developing software for multi-access edge computing," 2nd ed., ETSI, Valbonne, France, ETSI White Paper 20, 2019.
[23] J. F. Groote and M. R. Mousavi, *Modeling and Analysis of Communicating Systems*. Cambridge, MA, USA: MIT Press, 2014.
[24] D. Krajzewicz, "Traffic simulation with SUMO simulation of urban mobility," in *Fundamentals of Traffic Simulation*. New York, NY, USA: Springer, 2010, pp. 269–293.
[25] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Jan. 2011.
[26] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency Comput., Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.
[27] S. Filiposka, A. Mishev, and K. Gilly, "Community-based allocation and migration strategies for fog computing," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018.

KATJA GILLY (Member, IEEE) received the Ph.D. degree in computer science from Miguel Hernández University, in 2009. She has been teaching at Miguel Hernandez University in Elche, since 2001. She is currently working as an Associate Professor and she has been the Head of the Department of Computers Engineering, since June 2019. She has participated as coauthor in more than 50 international research publications including conferences and journals. Her research is centered in cloud oriented and edge computing datacentres, web servers and networking performance, QoS, resource management, and virtualization.

ANASTAS MISHEV (Member, IEEE) received the Ph.D. degree in computer science, in 2009. His research focused on infrastructures for collaborative computing and research, primarily grid and high-performance computing systems. He is currently working as a Professor with the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje. His aim is to get these systems closer to all potential users, mainly the research communities, in order to fully use their enormous potential. He researched in the areas of computer architectures and networks, software engineering, the Internet technologies, and e-learning. He has coauthored more than 80 scientific articles published in international journals and proceedings of conferences.

SONJA FILIPOSKA received the Ph.D. degree in technical sciences with speciality in computer engineering—computer networking from the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University in Skopje, North Macedonia, in 2009. From 2013 to 2014, she held a postdoctoral position on the topics of energy efficiency in the cloud at the University of the Balearic Islands. She is currently working as a Full Professor with the Faculty of Computer Science and Engineering, Department of Computer Networks and Information Systems. She has authored or coauthored more than 100 scientific articles published in international journals or conference proceedings and has participated in a number of Horizon 2020 research projects on the topics of networkings and e-infrastructures. Her research interest include automation and orchestration of services in a multidomain environment, complex networks, next generation networks, and multifactor optimization problems.

SALVADOR ALCARAZ received the Ph.D. degree in computer science and communications from the University of the Balearic Islands, in 2015. He is currently teaching Computer Networks and Protocol Engineering as an Assistant Professor with Miguel Hernandez University in Elche. He researched in areas of networking performance and QoS in Internet. He is currently researching in topics about protocol engineering and formal specification of protocols and computer systems. He has participated as coauthor in several international research publications, conferences, and journals.