A Novel Design for Advanced 5G Deployment Environments with Virtualized Resources at Vehicular and MEC Nodes

Published Version:
Feraudo A., Calvio A., Bujari A., Bellavista P. (2023). A Novel Design for Advanced 5G Deployment Environments with Virtualized Resources at Vehicular and MEC Nodes. 345 E 47TH ST, NEW YORK, NY 10017 USA : IEEE [10.1109/VNC57357.2023.10136327].

Availability:
This version is available at: https://hdl.handle.net/11585/953701 since: 2024-01-22

Published:
DOI: http://doi.org/10.1109/VNC57357.2023.10136327

A Novel Design for Advanced 5G Deployment Environments with Virtualized Resources at Vehicular and MEC Nodes

Angelo Feraudo, Alessandro Calvio, Armir Bujari, Paolo Bellavista
Department of Computer Science and Engineering
University of Bologna
Bologna, Italy
name.surname@unibo.it

***Abstract*—IoT and edge computing are profoundly changing the information era, bringing a hyper-connected and context-aware computing environment to reality. Connected vehicles are a critical outcome of this synergy, allowing for the seamless interconnection of autonomous mobile/fixed objects, giving rise to a decentralized vehicle-to-everything (V2X) paradigm. On this front, the European Telecommunications Standards Institute (ETSI) proposed the Multi-Access Edge Computing (MEC) standard, addressing the execution of cloud-like services at the very edge of the infrastructure, thus facilitating the support of low-latency services at the far-edge. In this article, we go a step further and propose a novel ETSI MEC-compliant architecture that fully exploits the synergies between the edge and far-edge, extending the pool of virtualized resources available at MEC nodes with vehicular ones found in the vicinity. In particular, our approach allows vehicle entities to access and partake in a negotiation process embodying a rewarding scheme, while addressing resource volatility as vehicles join and leave the resource pool. To demonstrate the viability and flexibility of our proposed approach, we have built an ETSI MEC-compliant simulation model, which could be tailored to distribute application requests based on the availability of both local and remote resources, managing their transparent migration and execution. In addition, the paper reports on the experimental validation of our proposal in a 5G network setting, contrasting different service delivery modes, by highlighting the potential of the dynamic exploitation of far-edge vehicular resources.**

***Index Terms*—Multi-access Edge Computing, MEC, Vehicular Computing, VANET, 5G**

I. Introduction

Thanks to the vast improvements in computing technologies and the pervasive deployment of next-generation communication networks, it is estimated that every new vehicle will be connected in the near future [1], embodying the potential of a fully-fledged mobile computing platform where vehicles serve as computation nodes for a diverse range of services. Different from vehicular networking [2], which serves as a communication enabler for applications associated with transportation, vehicle computing focuses on the computation function and emphasizes the promising role it embodies towards the implementation of a pervasive context-aware computing environment. This environment can play a major role in future ICT systems for supporting applications like Intelligent Transportation Systems (ITS) and full-scale smart cities [3].

Mobility-as-a-service and high-definition (HD) map generation are examples of such services, provisioned spanning cloud-to-vehicle resources where computationally heavy tasks are offloaded to resource-hungry cloud-based nodes. However, as the number of embedded sensors and smart vehicles grows, the amount of in-vehicle generated data may represent a serious problem for the infrastructure [4]. At the same time, cloud-based solutions are generally unfit to serve delay-sensitive application scenarios, which are currently served by shifting the computation burden to the edge of the network while providing services and data preprocessing functions.

On this front, the European Telecommunications Standards Institute (ETSI) has worked on the standardization of a cloud platform co-located at the edge of the network, including the Radio Access Network (RAN), named Multi-Access Edge Computing (MEC) [5]. The standardized architecture includes functional components tasked with the management and orchestration of edge resources, managing MEC applications' life-cycle, and providing standardized reference points to access the services. ETSI MEC augments legacy radio units with cloud-like computing capabilities, thus representing a pillar technology for (beyond) 5G cellular networks, allowing application deployment spanning cloud/edge resources [6], [7].

New opportunities and challenges arise as a growing number of businesses start to exploit the shared edge-cloud environment. In contrast to traditional cloud deployment environments in data centers, the edge has limited resources and may not always be able to satisfy application demands for resources and associated QoS. Moreover, the technical challenges associated with advanced edge infrastructures are exacerbated by the convergence trends meant to make sure an end-user can access the whole range of subscribed services. For these reasons, the need has emerged to identify new resources that can support the edge infrastructure dynamically, thus enabling service availability in dense and congested deployment scenarios. To this end, some examples of applications that may benefit from such a dynamic scenario involve decentralized learning contexts, where opportunistic resources can improve the overall learning process in terms of data quality and training/prediction speed. Additionally, in a smart city context, the availability of more resources could enable the creation of edge-enabled digital twins to keep track of assets present in the city and execute computationally demanding simulation tasks.

Taking a step towards the implementation of the aforementioned pervasive computing environment, our proposal extends the MEC resource pool so to leverage resources at the far-edge layer (i.e., the outermost edge of the network), exposed and made available in a standardized way. To this end, we propose a novel ETSI MEC-compliant architecture that can tap into far-edge resources within an Area of Interest (AoI). Node resources are registered at the edge resource pool, exposed and accessed via standardized interfaces. To handle node mobility, the architecture assists the application migration by triggering the user context transfer to MEC applications running on nodes leaving the AoI. To demonstrate the viability of our proposal, we have developed a simulation model that relies on Simu5G, an OMNeT++ simulation library. This model extends the MEC infrastructure with vehicular onboard resources by supporting MEC application deployment on vehicle resources and manages their volatility as vehicles enter and leave the AoI. Our simulation model is readily available for researchers in [8]. We evaluate the proposed approach by benefiting from two real-world datasets (about user mobility and parking lot occupancy) where there is the need to manage service migrations triggered by vehicle mobility. Finally, let us highlight that this work represents the first effort at enabling the deployment of vehicular resources in a multi-vendor and multi-operator scenario, through the exploitation of (the original extension of) an established ETSI standard.

II. Background

This section provides a brief description of the main functional elements of the ETSI MEC architecture used as the standard reference basis in our design, along with details about the simulation tools adopted to develop the extended model.

A. ETSI MEC

The MEC standard by ETSI proposes an architecture [5] introducing cloud computing capabilities at the very edge of the network, offering the ability to run so-called MEC Applications (MEC-App) within a virtualized and multi-tenant environment. The MEC architecture comprises two main parts: the system and host levels. The system level represents the entry point for users to request service execution and it is usually deployed in the core network. The central element, the MEC Orchestrator (MEC-O), has a view of the MEC Hosts (MEC-Hs) present in a particular area and is responsible for choosing the best one according to the requirements of the requested service, triggering both the instantiation and termination of applications. In this context, the User Application LifeCycle Management Proxy (UALCMP) acts as an intermediary for the users by supporting instantiation and termination requests forwarded to the MEC-O.

TABLE I: Table of acronyms for MEC elements

Abbreviation	Definition
AMS	Application Mobility Service
MEC	Multi-access Edge Computing
MEC-App	MEC Application
MEC-H	MEC Host
MEC-O	MEC Orchestrator
MEC-P	MEC Platform
MEC-PM	MEC Platform Manager
UALCMP	User Application LifeCycle Management Proxy
VI	Virtualisation Infrastructure
VIM	Virtualisation Infrastructure Manager

The host level lies at the edge of the network and realizes and manages the virtualization platform where applications are deployed. At this tier of the network, the MEC-H is the main entity providing computational, network, and storage resources for the MEC-Apps by exploiting the MEC Platform (MEC-P) and the Virtualization Infrastructure (VI). The MEC-P exposes a service registry in which applications can discover, offer and consume standard MEC services defined by ETSI; at the moment of writing, the standard supports four types of services that each MEC-H might maintain: Radio Network Information Service, Location Service, Traffic Management Service, and Application Mobility Service (AMS) [9] conceived to handle MEC-App migrations between edge nodes. In this context, each application runs as a virtual machine or container on top of the VI, and it optionally interacts with the MEC-P and its registry to take advantage of the standard MEC services. Finally, alongside the MEC-H, the Virtualization Infrastructure Manager (VIM) and the MEC Platform Manager (MEC-PM) serve as access points for the MEC-O to the lower layer. The VIM has the task of managing and releasing the virtualized resources and appropriately configuring the VI to run software images, while the MEC-PM handles all the components functions related to a specific MEC-H.

B. OMNeT++ and Simu5G

OMNeT++ [10] is a discrete event simulator framework used to model and build general-purpose simulations. It proposes a modular architecture based on components that can be arranged to create simulation models easily and effectively.

Build on top of OMNeT++ is Simu5G [11], a simulation library containing a collection of models and components useful for creating arbitrarily complex end-to-end scenarios involving 5G radio networks. Simu5G models both the core network and the RAN of a 5G network through the implementation of 3GPP-compliant protocols and a physical transmission system based on a set of customizable channels. The simulations created with this library can also leverage heterogeneous models related to gNBs base stations while supporting handover and inter-cell interference coordination. Indeed, Simu5G can be used to analyze transition scenarios, from 4G to 5G networks, thanks to the ability to use dual connectivity (X2) between

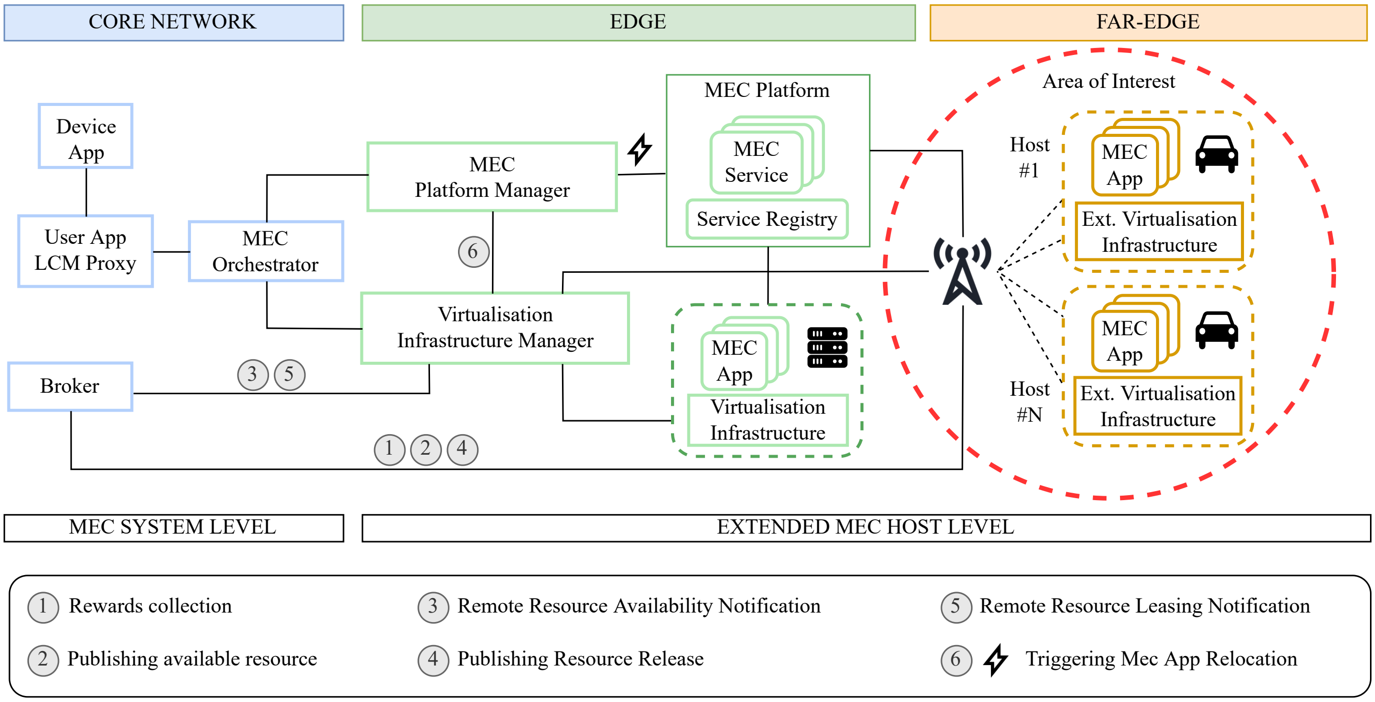


Fig. 1: MEC extension architecture diagram

eNB and gNB access types. Among the various implemented models, Simu5G provides an implementation of the ETSI MEC standard with its main components (i.e., MEC-O, MEC-H, etc.) and the ability to create applications that communicate via ETSI-compliant interfaces with other elements of the MEC ecosystem. Running applications can either be self-contained or take advantage of the presence of standard MEC services; currently, the framework implements the Location Service and the Radio Network Interface Service. A significant feature of the tool is that Simu5G can also be employed as a real-time emulator to replace simulation elements with real devices and thus use the same code for simulation and prototyping. Additionally, this allows for more realistic and reliable data collection.

Our simulation model works within the OMNeT++ simulation tool and relies on the Simu5G library. It provides a new interpretation of the ETSI MEC functional elements enabling dynamic resource acquisition at MEC Host level.

While preserving the general aspect of our study and without loss of generality, in the following section, we survey prior research effort on key design features our solution embodies.

III. Related Work

Considerable research effort has been devoted in the past, advocating for the use of vehicular resources to improve service delivery at the edge [12]. The concept of Vehicular Cloud Computing was initially proposed in [13] and [14]. Abuelela and Olariu [13] introduced the concept of Vehicular Cloud (VC), where vehicular resources are exploited as a mean to provide diverse community services. Similarly, Gerla in [14] outlined two applications of VC in which vehicles not only act as datacentres but also as observers of the environment. The service delivery model discussed in these works embodies new challenges when compared to the traditional, infrastructure-based one, such as task scheduling and distribution, resource volatility and acquisition, to be attributed to the unpredictability of the environment, i.e., node mobility.

Counteracting the mobility phenomenon, the authors in [15], [16] proposed to use parked vehicles as relay nodes, which can help improve connectivity and augment the chances of message delivery. Conversely, other works focused on exploiting vehicle resources for edge application execution [17]–[20]. On this front, Huang *et al.* [17] proposed a centralized architecture, where the central node (i.e., nodes at the edge of the network) receive task request which are then distributed as sub-tasks on selected parked vehicles. Similar in its objective, the work in [18] proposes a decentralized approach, offloading task execution to vehicle resources available nearby. Addressing a practical challenge to service delivery, Li *et al.* [21] defined a contract-based incentive mechanism to persuade vehicle owners to rent out their resources. Similarly, the authors in [16] propose an auction-based model where participating nodes compete to lend their resources in an extended vehicular resource pool.

Due to mobility, the vehicular resource pool might be subject to continuous changes in terms of capacity and task offloading decisions require carefully consideration. The necessity for this might also arise due to an inaccurate estimate of residual resource availability. In this context, most of the works propose algorithmic strategies used to evaluate the probability of nodes to complete task execution [17]–[20], [22]. However, these approaches are probabilistic, neglecting also practical considerations such as nodes refusing to partake in the resource pool. In this direction, the authors in [23] present a two-stage algorithm that handles service migration from one service provider to another in a vehicular network context. The algorithm relies on several metrics, such as average latency and energy consumption, to properly select the next service provider. Although there has been some research effort addressing task scheduling and resource allocation problems in the vehicular network tier, these proposals neglect and make no consideration of the challenges arising in multi-vendor and multi-domain environments. Furthermore, the migration problem does not seem to concern many authors.

In this paper, we propose an ETSI MEC-compliant architecture and an accompanying simulation model, extending the edge resource pool to contemplate far-edge (vehicular) resource infrastructures. Vehicular resources can be transparently accessed and made readily available through standardized interfaces. Our proposal has built-in mechanisms capable of deploying and distributing applications on the available resource pool, while addressing resource volatility issues (i.e., nodes joining/leaving) via a transparent migration mechanism that exploits already available constructs. As an extension of the ETSI MEC standard, the model allows us to deal with some of the aforementioned challenges, providing better integration with cloud resources and enabling coexistence among heterogeneous technologies.

IV. Our Proposed MEC Extension for the Dynamic Exploitation of Neighbor Vehicular Resources

Our work stems from the observation that MEC-Hs (i.e. edge nodes) have limited capabilities when compared to cloud-backed ones. The proposal turns the MEC-H into a logical entity that can leverage multiple VIs, dynamically adding and removing localized computational resources.

Referring to Fig. 1, our proposal allows the inclusion of the far-edge (vehicular) layer in cloud continuum deployment environments. Thus, in addition to locally defined hosts (edge nodes), it involves also remote host resources that are reachable and added dynamically via the RAN (e.g., 5G RAN). The approach entails some changes in the MEC traditional architecture in terms of structure and interactions (steps ①-⑥ in Fig. 1). The main component to be affected is the VIM that is in charge of administering the host resources. In our design, it handles a heterogeneous pool of distributed resources and is aware of the single contributions that each host brings in terms of capacity. In this context, it is desirable to differentiate between infrastructure resources and the more transient ones, dynamically joining the MEC-H thanks to vehicle availability in the neighborhood.

Our architecture proposal implies the creation of a mechanism to handle far-edge resources joining and leaving the MEC-H resource pool. Thus, each MEC-H defines an AoI within which far-edge nodes (hosts) can decide whether to provide their onboard resources. To model this mechanism, we decided to involve a new external component in the core network, named Broker, which handles the resource pooling of several MEC-Hs. To this end, the Broker relies on a publish-subscribe model to collect MEC-H AoI subscriptions and manage notifications whenever new devices enter the area. The AoI may depend on where MEC-Hs are located, i.e., at the network edge or network aggregation points [6]. Hence, each MEC-H, at bootstrap time, subscribes to an AoI that might coincide with one or more zones, which typically correspond to the coverage of the associated gNBs.

A reward system encourages far-edge nodes to lease their computational capacity and join the resource pool. Two different schemes of the procedure are envisioned: network- or vehicle-initiated. The former requires the Broker to provide each far-edge node in the AoI a set of rewards to incentivise resource leasing. The latter (steps ①-③) expects far-edge nodes to manifest their intention to join the pool, by asking for available rewards contextualized to the AoI. In both cases, if the device finds acceptable terms, it can confirm the intent to participate by communicating to the Broker the set of leased resources. The current design adopts the second approach, whereby vehicles obtain available rewards and can autonomously decide whether completing resource registration. Similarly, when one of them leaves the AoI, it notifies the Broker, which forwards the release request to the VIM managing that area. The latter in turn removes the concerned resources from those available in the pool.

The resource release procedure (steps ④-⑥) requires more attention as the departing host may have applications running on it. In such a case, after receiving the release notification, the VIM triggers the migration procedure (step ⑥) to move running apps from one host to another and thus maintaining service continuity with very low service interruptions. Both registered hosts in the AoI and the local infrastructure of the MEC-H are eligible to support the migration operation and embrace the new application(s). The AMS, defined in the ETSI standard, currently supports app migration in environments encompassing multiple edge nodes. Our extension has been designed to work in this perspective by extending the service to further support intra-host migration in a standard way. Furthermore, it is MEC-PM that is identified as the main component that, during the procedure, acts as an intermediary node between the AMS and the VIM for new app allocations.

Finally, concerning the resource allocation and scheduling approach, the VIMs initial selection results in a set of hosts eligible for app deployment; the scheduling phase, leading to the identification of a single host, can be done by ordering nodes depending on strategies that might favor certain aspects over others. Some metric examples are the average latency time between a host and the central infrastructure or the probability of a node to further contribute to the resource pool (i.e., based on historical data).

The proposed design approach paves the way for innovative application scenarios, which go beyond the state-of-the-art far-edge computing ones targeted nowadays, such as task offloading and content caching. For example, our proposal could be a key enabling element for the hosting of Federated Learning [24] enabled environments at the far-edge layer. Specifically, a MEC-App (federated server) running on MEC-H local resource infrastructure can coordinate other MEC-

Apps (federated clients) deployed on remote resources during the training on local data. The former can choose federated clients by using the model descriptors and information collected through the MEC standard API, while the latter, after receiving a federated model, can start the training procedure by relying on their status, local data, and received rewards.

V. Evaluation Results

We built a simulation model of the proposed ETSI MEC extension in the OMNeT++ simulation tool, using Simu5G as a communication library, so to demonstrate the viability of our approach in (beyond) 5G scenarios. In the current release, available in [8], vehicular nodes are modeled and identified as 5G-enabled User Equipment (UE). The exposed vehicular resources consider the CPU in terms of instructions/second, RAM, and disk space available for the lease. Furthermore, the vehicle model includes a MEC VI managing its onboard resources and applications' life-cycle and a module that manages rewards, resource registration, and releasing.

At model startup, each MEC-H identifies its AoI corresponding to an area within the associated gNB coverage. Thus, when a vehicle enters the area, it starts interacting with the Broker, according to the protocol described in Sec. IV. As a result, the vehicle joins the MEC-H resource pool and can be considered by the VIM scheduling logic, which in the current implementation of our model uses a simple Round Robin algorithm. Note that when vehicles leave the AoI, corresponding app migrations might be triggered: we have decided to leverage only local MEC-H infrastructure as a target platform where to migrate MEC-Apps. The rationale behind this choice is to avoid expensive and inefficient domino effects where a vehicle receiving a migrated app leaves the AoI, thus triggering a new migration.

In our environment, we adopted a 5G standalone network, using a numerology index $\mu = 2$ and encompassing a single MEC-H connected to a gNB and a cloud datacenter to enable comparisons among multiple service delivery modes. The simulation scenario includes a parking area, where vehicles can enter and leave as they want. We currently provide a basic reward scheme using integer values, which is accepted by all vehicles partaking the resource acquisition procedure.

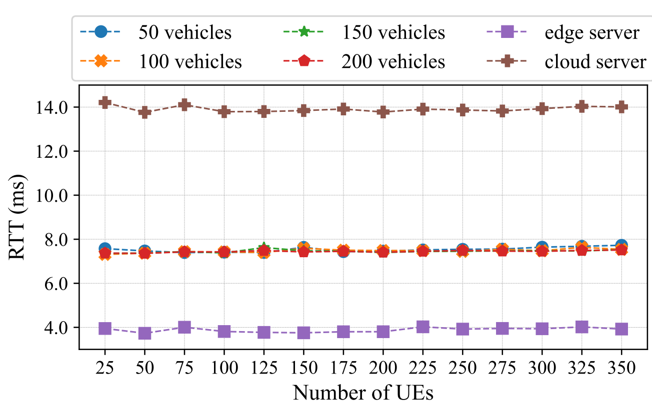


Fig. 2: RTT variation with varying number of UE requests in the considered service delivery scenarios. The x-axis denotes the number of UEs requesting an app execution, while the y-axis denotes the average Round-Trip Time between UEs and the MEC-App. In the far-edge delivery mode, MEC-Apps requested by the UEs are deployed onboard the vehicle VI.

We have conducted an extensive set of experiments on a Linux VM running OMNeT++ having 16 CPUs and 64 GB of RAM. It is noteworthy to point out that according to the definition of discrete event simulator, OMNeT++ does not consider the processing time spent to run the code of any module; hence, all the evaluations concern network-related delays. In the following, we evaluate our model and provide an experimental analysis to assess model performance under dynamic conditions.

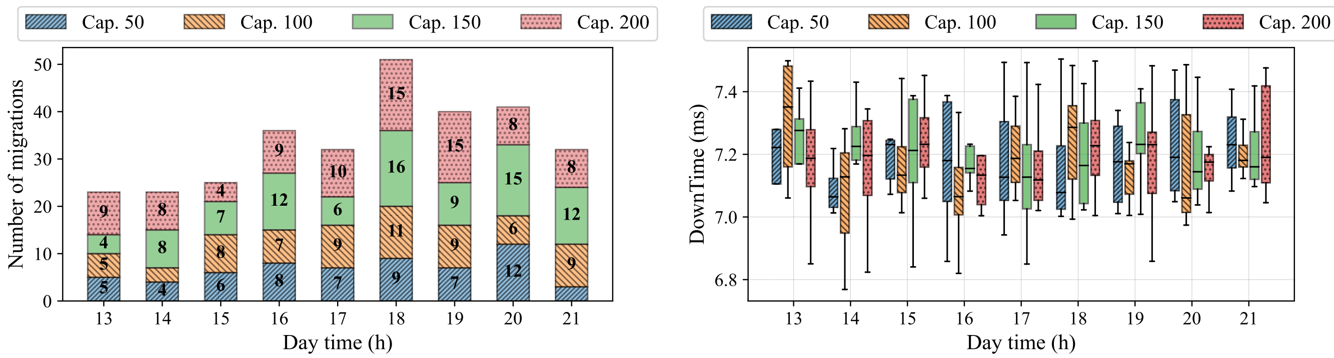
A. Model validation

To validate the effectiveness of our model, we evaluate the network-induced delays in three service delivery schemes, namely at the cloud, edge of the network, and onboard the vehicle. In the cloud scheme, the MEC-Apps are hosted on a simulated cloud datacenter, thus involving data transmissions spanning the 5G RAN, edge, and core network. The second service delivery scheme embodies delays introduced by running MEC-App directly on the MEC-H, thus those due to 5G RAN and MEC local User Plane Function (UPF) co-located with the gNB [6]. The last scheme involves MEC-App running on onboard vehicle resources, i.e., remote host in Fig. 1. In the latter case, network delays are affected by data transmission between the network (i.e., gNBs) and devices (i.e., vehicle running MEC-Apps and UE requesting their execution).

Figure 2 shows the Round Trip Time (RTT) of the above-mentioned schemes. On the x-axis, we considered the number of clients requesting the execution of a MEC-App to assess the behavior of the system under different load scenarios. In this experimental setup, we assume that each UE requests precisely the execution of a single MEC-App as soon as it enters the gNB coverage. Additionally, UEs are spawned at the same time, simulating a flash-crowd phenomenon. To reduce bias in the results, each experiment is repeated 5 times.

The analysis of the third delivery scheme has been conducted by considering different vehicle quantities participating in the resource pool of the MEC-H namely, 50-100-150-200 vehicles. In this scenario, the MEC-App execution triggered by the UEs requests is executed onboard the vehicle. This setting allows us to get further insights about scalability and the relationship between delays and the number of apps deployed on a single vehicle. The figure shows how latency is steady even with more than 500 devices (i.e., 350 UEs and 200 vehicles) within the coverage of the MEC-H associated gNB, almost independently on the number of vehicles when in these intervals of practical interest (note that computing-related delays are not considered in the reported simulations).

Deploying MEC-Apps on vehicles brings the advantage of reducing the RTT by around 50% as opposed to cloud deployment. On the other hand, the far-edge scheme increases the RTT by 3ms if compared with the edge mode, as it



(a) Migration frequency

(b) Downtime

Fig. 3: Migration related metrics

exploits fully wireless communications via the 5G network infrastructure. However, it should be noted that Simu5G handles device-to-device (D2D) communications through the gNB base station, thus employing a network-mediated communication model also in D2D scenarios. We expect that the adoption of *sidelink* mode of 5G network would further reduce the RTT, as this enables direct communication between two devices without the participation of a gNB in data transmission and reception.

B. Migration Study

As mentioned in Sec. II, far-edge nodes may leave the resource pool once their resources have been allocated, resulting in a service disruption. To cope with this problem, the MEC-H initiates a migration procedure, moving running apps from the host leaving the resource pool into another one. However, migration of stateful MEC-Apps may generate a downtime period in which the involved app becomes unavailable.

To measure the associated performance indicators, we build a realistic and innovative scenario involving vehicle volatility and UE activities in a parking area context. To make this scenario as close as possible to reality, we need information regarding vehicle activity (i.e., vehicles joining and leaving an AoI) and UEs activity exploiting a network connection in the surrounding area. To the best of our knowledge, there are no datasets related to a specific area and collecting information similar to the scenario highlighted above. Thus, in order to recreate such an environment, we ground our study and make use of two real-world datasets [25], [26]. The first dataset captures vehicle join and leave times in three parking garages in the city of Arnhem, while the other one describes the usage of public WiFi networks in the city of Bologna, recording the number of users joining the WiFi network for each day hour.

To synthesize the aforementioned dynamics, we looked for a relationship between these datasets, analyzing each parking garage and each network available. Therefore, we selected one parking garage context and location, and filtered Bologna's dataset using this information. The location chosen is the *Central garage*, a parking area able to host more than 1000 vehicles and close to the train station of Arnhem. Successively, we extracted data from WiFi networks within the train station of Bologna. After the pre-processing step, we studied the distributions of the occupancy time, i.e., the time a vehicle stays parked, the average number of vehicles entering per hour, and the average number of UEs joining the network per hour. Based on the fitted distribution found for the occupancy time, we decided to approximate it with a normal distribution with mean $\mu = 202.80$ minutes and standard deviation $\sigma = 135.07$. Other metrics, i.e., the number of entering vehicles and UEs per hour, are better captured by a Poisson distribution as they are independent events that occur randomly within an hour, but with a known average rate. The Poisson interval time considered is 3600 seconds, with the average value λ varying depending on the hour of the day.

We use the obtained distributions in our simulation model to spawn parked vehicles, hence the requesting UEs. This way, we build a dynamic scenario and simulate vehicles joining and leaving the AoI, and UE requests for MEC-App execution. The simulation settings are the same as in the prior scenario, i.e., each vehicle lends its onboard resources, and each UE issues a request to run a MEC-App. The employed stateful MEC-App logic, provided by Simu5G library, allows the user to define a circular geographic area as a warning zone. The app is responsible for notifying the user whenever it enters and leaves that zone. More complex and meaningful location-based applications could be built, considering innovative scenarios such as decentralized (federated) learning etc.

Figure 3a illustrates the number of migrations generated adopting the distributions described above. We considered four arbitrary parking lot capacities (i.e., 50, 100, 150, and 200) used to scale quantities generated by the Poisson distribution of the original dataset describing vehicles' entrances in the parking lot. As MEC-App s are equally distributed among the parked cars using Round Robin algorithm, lower park capacities may correspond to a greater number of migrations, for instance, during day time at 15:00 and 20:00. Note that the number of migrations also depends on the user activities and the amount of parking cars in that hour of the day, both generated through probability distributions. Thus, the relocations occurred may vary according to the number of users requesting for app execution and available car resources, which are not illustrated in this paper. Furthermore, we analyzed the interval from 13:00 to 21:00, as it corresponds to day hours with more network activity. Finally, the downtime has been measured as the time interval between the shutdown of the app on the leaving host and the end-user receiving the new MEC-App location. Generally speaking, the downtime may be affected by latency between the two involved entities, i.e., remote and local VI, and bandwidth [27]. In our experiments, the former includes the delays due to MEC-H distance from the gNB and 5G radio delays, while the latter does not affect the service interruption time, as the state sent by MEC-Apps is smaller than 30B. Hence, as shown in Figure 3b, the downtime remains stable around 7 ms.Overall, despite the number of MEC-App relocations (Fig.3a) and the network activity increase, the downtime remains stable when migrating MEC-Apps from one host to another.

VI. Conclusions and Future Work

In this paper, we propose a novel architecture extending the ETSI MEC standard to enable the dynamic negotiation and acquisition of far-edge resources at MEC-H level. The presented scheme allows device owners to access a reward system and addresses resource volatility problems as devices join and leave the MEC-H resource pool. To demonstrate the viability of our approach, we built a simulation model that extends ETSI MEC towards scenarios with dynamic availability of vehicular resources. The model is validated with three service delivery schemes, thus demonstrating how our approach introduces good performance in MEC-enabled environments. Finally, we also proved its robustness in real-world dynamic conditions.

Although the nature of our architecture considers both physical and mobile nodes, in this discussion we have limited the simulation-based evaluation to parked vehicles and local migrations within the MEC-H. As future work, we envision expanding our simulation model to support resources provided moving vehicles in more complex scenarios, e.g., smart-city ones. These require tracing the vehicle's position to monitor its location inside the AoI since its resources are released and any task being executed on it is dynamically relocated once it moves beyond the borders of the AoI. Consequently, we plan to generalize the reward mechanism to support pluggable reward schemes, e.g., network-initiated and relying on user behaviors. Moreover, we plan to extend the support for pluggable scheduling modules so to study the effect of scheduling strategies that distribute and migrate MEC applications among far-edge nodes according to device parameters (e.g., energy consumption) and app requirements (e.g., max latency).

References

[1] Elektrobit, "Connected vehicle software solutions," last Accessed: 2022-10-03. [Online]. Available: https://www.elektrobit.com/products/eb-connected-vehicle/

[2] K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9457–9470, 2016.

[3] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A Survey on Vehicular Cloud Computing," *Journal of Network and Computer Applications*, vol. 40, pp. 325–344, 2014.

[4] AECC, "Distributed Computing in an AECC System," Aug 2021, last Accessed: 2022-10-03. [Online]. Available: https://aecc.org/wp-content/uploads/2021/09/Distibuted_Computing_White_Paper_v1.0.0.pdf

[5] ETSI. (2022, 3) GS MEC 003 - V3.1.1. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/03.01.01_60/gs_mec003v030101p.pdf

[6] S. Kekki *et al.*, "MEC in 5G Networks," *ETSI white paper*, vol. 28, no. 28, pp. 1–28, 2018.

[7] A. Sabbioni, L. Rosa, A. Bujari, L. Foschini, and A. Corradi, "Diffuse: A distributed and decentralized platform enabling function composition in serverless environments," *Computer Networks*, vol. 210, 2022.

[8] A. Feraudo and A. Calvio, "MEC extension." [Online]. Available: https://github.com/aferaudo/Simu5G/tree/feat/vim-extension

[9] ETSI. (2022, 2) GS MEC 021 - V2.2.1. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/021/02.02.01_60/gs_mec021v020201p.pdf

[10] OMNeT++ Event Discrete Simulator. [Online]. Available: https://omnetpp.org/

[11] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G–An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks," *IEEE Access*, vol. 8, pp. 181 176 – 181 191, 2020.

[12] S. Olariu, "A Survey of Vehicular Cloud Research: Trends, Applications and Challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2648–2663, 2020.

[13] M. Abuelela and S. Olariu, "Taking VANET to the Clouds," in *Proc. of International Conference on Advances in Mobile Computing and Multimedia*. New York, NY, USA: ACM, 2010, p. 6–13.

[14] M. Gerla, "Vehicular Cloud Computing," in *Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2012, pp. 152–155.

[15] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: Stopped cars are not silent," in *Proc. of IEEE INFOCOM*, 2011, pp. 431–435.

[16] P. Qin *et al.*, "Energy-Efficient Resource Allocation for Parked-Cars-Based Cellular-V2V Heterogeneous Networks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 3046–3061, 2022.

[17] X. Huang, R. Yu, J. Liu, and L. Shu, "Parked Vehicle Edge Computing: Exploiting Opportunistic Resources for Distributed Mobile Applications," *IEEE Access*, vol. 6, pp. 66 649–66 663, 2018.

[18] C. Ma *et al.*, "Parking Edge Computing: Parked-Vehicle-Assisted Task Offloading for Urban VANETs," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9344–9358, 2021.

[19] N. Cha, C. Wu, T. Yoshinaga, Y. Ji, and K.-L. A. Yau, "Virtual Edge: Exploring Computation Offloading in Collaborative Vehicular Edge Computing," *IEEE Access*, vol. 9, pp. 37 739–37 751, 2021.

[20] W. Fan, J. Liu, M. Hua, F. Wu, and Y. Liu, "Joint Task Offloading and Resource Allocation for Multi-Access Edge Computing Assisted by Parked and Moving Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5314–5330, 2022.

[21] C. Li *et al.*, "Parked Vehicular Computing for Energy-Efficient Internet of Vehicles: A Contract Theoretic Approach," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6079–6088, 2019.

[22] F. H. Rahman *et al.*, "Off-Street Vehicular Fog for Catering Applications in 5G/B5G: A Trust-Based Task Mapping Solution and Open Research Issues," *IEEE Access*, vol. 8, pp. 117 218–117 235, 2020.

[23] S. Ge, M. Cheng, X. He, and X. Zhou, "A two-stage service migration algorithm in parked vehicle edge computing for internet of things," *Sensors*, vol. 20, no. 10, p. 2786, 2020.

[24] K. A. B. *et al.*, "Towards federated learning at scale: System design," in *SysML 2019*, 2019. [Online]. Available: https://arxiv.org/abs/1902.01046

[25] Open Parkeerdata portal of the Municipality of Arnhem. [Online]. Available: https://parkeerdata.nl/opendata/arnhem/parkeergarages/transactiedata-parkeergarages

[26] Open Data Comune di Bologna. [Online]. Available: https://opendata.comune.bologna.it/explore/dataset/iperbole-wifi-affluenza/information/

[27] T. Taleb, A. Ksentini, and P. A. Frangoudis, "Follow-me cloud: When cloud services follow mobile users," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 369–382, 2019.