Unraveling Edge-based in-vehicle infotainment using the Smart Highway testbed

Nina Slamnik-Kriještorac and Johann M. Marquez-Barja

University of Antwerp - imec, IDLab research group, Sint-Pietersvliet, 2000 Antwerp, Belgium.
E-mail: {name.surname}@uantwerpen.be

***Abstract*—With the advancements in SDN and NFV, both applications and network functions can be re-designed, and deployed at more appropriate locations. Thanks to the MEC platforms, cloud-alike service deployments are offered to the users/vehicles at closer proximity. However, MEC deployments are usually i) constrained in resources, ii) contain heterogeneous and distributed network and computing resources, and iii) cover narrower region that constrains service continuity due to the high mobility of vehicles. Thus, in this paper, we present our approach on collocating MEC platforms with roadside infrastructure (i.e., RSUs) in order to improve the QoS of infotainment services for vehicles on the smart highway. We tackle both challenges presented above by deploying MEC platforms along the highway, thereby having distributed control over each MEC host in the form of Kubernetes master nodes, and one powerful and yet centralized orchestrator in the cloud. Our approach is one of the earliest attempts to collocate MEC with the RSU, and to test the benefits of the smart application placement in a realistic vehicular environment.**

***Index Terms*—MEC, RSU, vehicular system, containerized applications, smart placement, Smart Highway.**

I. Introduction and Motivation

Vehicular systems are fostering the synergy between various transportation standards (Intelligent Transportation System (ITS-G5), Vehicle-to-Everything (V2X)), communication technologies (802.11p, LTE, 5G, etc.), altogether with the cloud and edge computing, to provide enhanced services to vehicle users (drivers, passengers, and even pedestrians). Thus, in such system, a vehicle is empowered and equipped to be *smarter*, i.e., to better observe the circumstances on the road, and contextual driving information, and to exchange information with other participants in transportation in order to increase its capabilities. This exchange of information refers to other vehicles, pedestrians, roadside infrastructure, and computing clouds, enabling all of them to make more intelligent decisions that ensure safety and energy efficiency [1]. Furthermore, some of the essential vehicular services are those that address safety and demand an ultra-low latency (e.g., lane change warning, situation awareness, emergency electronic brake warning, etc.), but there is also a group of infotainment services that are more tolerable in terms of latency comparing to the safety use cases, but also require a high level of Quality of Service (QoS) (e.g., video streaming, Internet content sharing, IPTV, etc.) [2].

Due to the programmability of communication networks, which is achieved by applying Software Defined Networking (SDN) and Network Function Virtualization (NFV) techniques, the traditional network functions in legacy networks can be virtualized and placed at appropriate locations (e.g., closer to the users, in the overcrowded places, etc.). Moreover, the capacity limitation and increase in delay impose severe challenges to mobile users when the applications are placed in cloud servers [3]. Hence, the robust applications that were previously deployed only in the cloud context now can be redesigned and deployed at the network edge. Thanks to the Multi-Access Edge Computing (MEC) platforms, cloud-alike service deployments are offered to the users at a closer proximity, enhancing the user experience and increasing network performance. The same applies to the vehicular networks, where the significant research effort [2, 4, 5] has been invested already to assess the opportunities of coupling MEC with vehicular systems.

However, MEC deployments impose several challenges that need to be properly addressed. Firstly, MEC platforms usually consist of heterogeneous and distributed network and computing resources, belonging to different operators/vendors. Secondly, deployments at the edge belong to the narrower regions than the cloud ones, and thus, the high mobility of vehicles triggers service re-association frequently to connect to the *closest* deployed service, which ultimately leads to service discontinuity. Thus, in this paper we present our research approach on how to collocate MEC platforms with roadside infrastructure, i.e., Roadside Units (RSUs), in order to improve QoS of infotainment services for vehicles on the highway. We tackle both challenges presented above by deploying distributed MEC platforms along the highway, thereby having *distributed* control over each MEC host in the form of Kubernetes (Kubernetes)[1] master nodes, and one powerful and yet centralized orchestrator in the cloud, i.e., Centralized MEC Application Orchestrator (cMEAO). The main role of cMEAO is to perform resource management in the distributed network edges by applying smart application placement mechanism. This smart mechanism is defined as a decision-making problem influenced by multiple criteria, such as availability of resources in MEC hosts, geographical location of the vehicles and the RSU, latency requirements, mobility patterns, etc., and it results in a selected MEC host, i.e., underlying Kubernetes cluster that will be used for application deployment. Our approach presented in this

[1] Kubernetes: https://kubernetes.io/

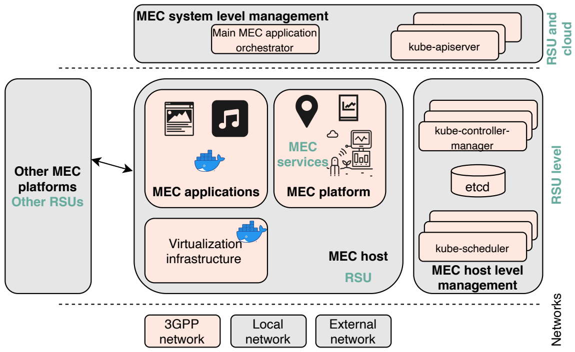

Fig. 1: The mapping of our MEC-backed vehicular system with ETSI MEC standard framework

paper is pragmatic, and we show the opportunities of experimentation with an orchestrated vehicular system in a realistic environment, such as the Smart Highway[2] testbed, which is a test site built on top of the E313 highway, located in Antwerp, Belgium. To the best of our knowledge, our approach is one of the earliest attempts to collocate MEC platforms with the road infrastructure, and to test the benefits of the smart application placement in deployment in a realistic vehicular environment.

II. The MEC-backed vehicular platform

In this section, we briefly present the vehicular communication platform that consists of roadside infrastructure (i.e., RSUs), MEC platforms, cloud infrastructure with the cMEAO, and the smart highway with connected vehicles. Firstly, we tackle the alignment of this platform with the current standards, as ETSI is the pioneering standardization body for MEC. Secondly, we present the architecture and all components of this vehicular platform, and finally, we tackle the virtual Content Delivery Network (CDN) service deployment for infotainment type of the vehicular use case.

a) Alignment with ETSI MEC standards: Being designed and developed in line with the current standards is important for all MEC-based platforms because of the opportunities to have common capabilities over the edge infrastructure of the operators. Taking into account the standards presented by ETSI MEC [6], we map the architectural components of our platform to the MEC framework, as shown in Fig. 1. In Fig. 1, we can see that the MEC system level management is in our case dispersed to both cloud and edge (i.e., RSUs along the highway). To extend the view of the cMEAO on the overall MEC/RSU infrastructure, we deploy it in the cloud. This orchestrator is then connected to Kubernetes API server components that are deployed in each Kubernetes cluster on the RSUs, as the anchor of Kubernetes master node. The management and control of each particular MEC platform (i.e., the MEC/RSU host level management) is performed by controller managers and schedulers that make sure that MEC applications have sufficient amount of resources to run, and that they operate correctly. As we host the MEC within the RSU device, the RSUs contains MEC helper services, such as location services, sensor data fusion services, services that analyze monitored data from Kubernetes cluster, and services that parse and decode/encode the messages from/to vehicles. All these services are used as helper functions for the MEC applications that are developed for a certain use case. Furthermore, all MEC applications are deployed as Docker containers in PODs, which are the atomic units of deployment in Kubernetes.

The inter-MEC communication that is shown on the left-hand side of the Fig. 1 is enabled by the cMEAO. Furthermore, depending on the type of the MEC application, it can be developed as a data plane communication between application themselves, allowing these application instances to collaborate and share data in order to expand their scopes. Finally, the connectivity with vehicles in the Smart Highway testbed can be obtained via hybrid communication modules, either 3GPP LTE, or ITS-G5 and V2X.

b) The platform architecture: As shown in Figure 2, the MEC platforms are collocated within RSUs, the wireless communication devices that are located along the road to provide connectivity to the vehicles. The RSUs are installed as a part of the Smart Highway testbed [7], which is the test site built on top of the E313 highway in Antwerp, Belgium. For instance, the map in Fig. 4 showcases the locations of some of the RSUs that are deployed along the highway site. In order to differentiate edge domains in our internal experimentation, we mimic the multi-site deployment by dividing the RSUs into the groups (as shown in blue boxes of Figure 2). Therefore, the deployment of applications in different RSU domains mimics the scenario with multiple administrative and technological domains, with multiple edge domain per each.

Each RSU consists of a large electrical cabinet that accommodates different modules, such as: i) modules for wireless communication (ITS-G5 and V2X with PC5 interface on 5.9GHz, and the long-range communication based on 4G on 3.5GHz, i.e., Uu interface), ii) modules for local computing processing on the RSU, and iii) modules that allow us to remotely manage and orchestrate the RSUs [7]. In particular, the General Purpose Computing Units (GPCUs) enable the edge computing, and extend the capabilities at the edge, thereby enabling deployment of distributed MEC applications closer to the users (i.e., vehicles) and decreasing the latency to dozens of milliseconds. The full palette of features that characterize the RSUs deployed on the Smart Highway testbed is the following: i) GPCU, a unit that processes the incoming data and controls the RSU, ii) a radio unit that consists of the radio communication technologies, iii) antenna unit containing the antennas to support the wireless transmission, iv) recovery unit that recovers the RSU if any failure occurs, v) power unit, controlling the power to the different sub-units in the RSU.

For the deployment of the MEC platform in RSU, we utilize the GPCU that is based on the machine of Intel Xeon 8 Cores with 32 GB RAM. In particular, each GPCU has its own Kubernetes cluster, configured and monitored by Kubernetes

[2]Smart Highway: https://www.fed4fire.eu/testbeds/smart-highway/

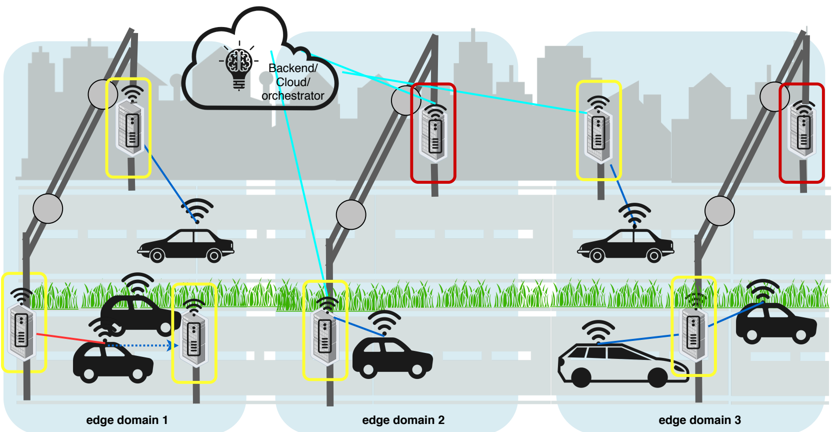

Fig. 2: The collocation of MEC platforms with RSUs on the Smart Highway testbed

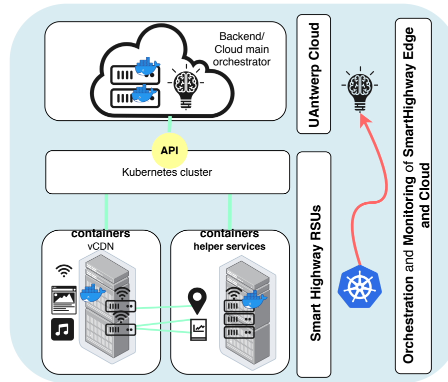

Fig. 3: The logical architecture of our vehicular platform

internal monitoring service that is based on Prometheus[3]. As presented in Fig. 3, one Kubernetes cluster covers both containerized vCDN MEC applications that serve directly the vehicles on the road, and helper containers, which are in charge of: i) processing the real-time monitoring input from MEC applications running in cluster, thereby generating statistics on applications and MEC host for the cMEAO, ii) retrieving data from sensors that can be attached to the roadside infrastructure (e.g., cameras, traffic lights, etc.), iii) tracking the location of certain vehicles, and iv) processing the messages from/to vehicles on the road (e.g., Cooperative Awareness Message (CAM) and Decentralized Environmental Notification Message (DENM)).

To be able to share aforementioned statistics with the cMEAO in the cloud, for each Kubernetes cluster in RSU we develop an Application Programmable Interface (API) that is based on REpresentational State Transfer (REST) protocol (3). This way, the cMEAO is subscribed to monitoring services in Kubernetes clusters, monitoring the CPU, memory, storage of the MEC hosts, but also the application containers that are running in clusters. The type of service that we deploy is a virtual CDN, which we expose to vehicles, but cMEAO is making decision where to place it depending on the availability of the distributed MEC hosts, and more importantly the location of the vehicles. In the following subsection, we provide more information on the service type, and finally briefly describe the main features of our smart placement algorithm in Section III.

c) Deployment of vCDN services: Due to the opportunities to deploy services and virtualized network functions closer to the road, the distributed cloud/edge architecture is going to be a cornerstone for vehicular communications, providing low latency services tailored to various 5G automotive use cases. In this experimental vehicular platform we focus on the infotainment use case, such as virtual CDN as a service, with cache CDN servers placed within MEC in order to decrease

[3]Prometheus: https://prometheus.io/

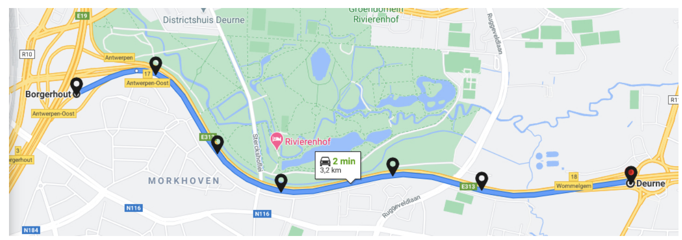

Fig. 4: The locations of RSUs deployed on the E313 highway in Antwerp, Belgium

the overall latency in accessing popular websites (e.g., Google maps) [2]. Thus, CDN as a Service (CDNaaS) represents a containerized service instance of CDN, which is strategically instantiated and placed over the edge/RSU nearby users taking into account the number of users in certain region, content popularity, mobility patters, among others. For instance, in case there is an insignificant number of vehicles in the edge domain of a certain RSU (e.g., red-boxed RSUs in Fig. 2), taking care of the energy-aware green computing principles, there is no need to burden RSU with running a CDN service. Accordingly, the service can be muted/decommissioned in order to release resources for other more urgent MEC applications, and the vehicles from this domain can be redirected to the nearest RSU (e.g., yellow-boxed RSUs in Fig. 2) that hosts the same service instance.

III. Smart Placement of Vehicular Services

As explained in Section II, the smart placement algorithm is running in the cMEAO that is receiving statistics from each RSU, i.e., Kubernetes cluster. These statistics are taken as an *input* for the algorithm, and they collect: i) the aggregated information about QoS parameters for each running MEC application, ii) geographical positions of the vehicles on the highway, and iii) CPU/memory/storage consumption for each MEC host (algorithm 1). Tackling the geographical position of the vehicles, we are currently considering two different options for determining the location of vehicles: i) by receiving the periodical CAM messages that will be parsed and decoded in each Kubernetes cluster, and ii) by deploying sensors along the highway, which will collect information and expose it to RSUs and Kubernetes master nodes. One of the next steps in our ongoing experimentation is to test the performance of both variants, and choose the one which is more efficient.

Algorithm 1 The baseline of Smart placement algorithm

$Input \leftarrow$ {set of distributed MEC hosts (c); statistics collected from all hosts via Kubernetes monitoring services: host CPU, memory, storage, MEC applications aggregated statistics, geographical positions of MEC hosts and vehicles on the highway}
$Output \leftarrow NodeSelectors$
while true **do**
  foreach $c \in Input$ **do**
    $r_1 = checkResources(NodeSelectors(c))$
    $r_2 = checkAppStatus(fNodeSelectors(c))$
    $r_3 = checkGeographicLocations(NodeSelectors(c))$
    if $NodeSelectors(c)$ is activated **then**
      $optimization(r_1, r_2, r_3)$
      if $vehiclesDensity$ is low **then**
        $NodeSelectors(c) = 0$
    else
      $activate(NodeSelectors(c))$
  MONITORFORWRITES($Input$)
end

The overall information presented above is important to be collected, because the orchestrator needs to select the most suitable RSU for application deployment, i.e., the one that is serving a certain number of vehicles and has a sufficient amount of NFV Infrastructure (NFVI) resources to run the service. Depending on the type of application (e.g., CPU-bound, or memory-bound), the smart placement algorithm is assigning weights to parameters, and makes decision that is tailored to a specific application type. Finally, all the collected information is being stored and used a training data-set that enables prediction of future trends for resource consumption, as well as the density of vehicles on the highway.

IV. Conclusion and Future Work

In this paper we present our approach for the experimental setup of MEC-backed vehicular platform that includes MEC collocation with RSUs on the highway. For the realistic testing we use the Smart Highway testbed. The ongoing activities include: i) configuration of Kubernetes clusters in RSUs with the APIs that expose statistics to orchestrator in the cloud, ii) instantiation of vCDN Docker containers, and iii) optimization of the smart placement algorithm in the cMEAO. Out future work includes study of the feasibility of connecting real vehicles on the highway to the edge services running in RSUs, as well as the deployment of sensors on the highway. Such experimental setup that we are developing is going to be a resourceful pool of opportunities to conduct realistic and experimental research both in vehicular context, but also in the context of edge and cloud computing.

V. Acknowledgement

This work has been carried out with the support of the Horizon 2020 project 5G-CARMEN co-funded by the EU under Grant Agreement No. 825012, the Horizon 2020 Fed4FIRE+ project, Grant Agreement No. 723638, and the Horizon 2020 5G-Blueprint project, under Agreement No. 952189. The views expressed are those of the authors and do not necessarily represent the projects.

References

[1] Y. Ni, L. Cai, J. He, A. Vinel, Y. Li, H. Mosavat-Jahromi, and J. Pan, "Toward Reliable and Scalable Internet of Vehicles: Performance Analysis and Resource Management," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 324–340, 2020. doi: http://dx.doi.org/10.1109/JPROC.2019.2950349.

[2] N. Slamnik-Kriještorac, E. de Britto e Silva, E. Municio, H. Carvalho de Resende, S. Hadiwardoyo, and J. Marquez-Barja, "Network Service and Resource Orchestration: A Feature and Performance Analysis within the MEC-Enhanced Vehicular Network Context," *Sensors 2020*, vol. 20, 2020. doi: https://doi.org/10.3390/s20143852.

[3] H. El-Sayed and M. Chaqfeh, "The deployment of mobile edges in vehicular environments," in *2018 International Conference on Information Networking (ICOIN)*, pp. 322–324, 2018. doi: http://dx.doi.org/10.1109/ICOIN.2018.8343133.

[4] Z. Ning, J. Huang, X. Wang, J. J. P. C. Rodrigues, and L. Guo, "Mobile Edge Computing-Enabled Internet of Vehicles: Toward Energy-Efficient Scheduling," *IEEE Network*, vol. 33, no. 5, pp. 198–205, 2019. doi: http://dx.doi.org/10.1109/MNET.2019.1800309.

[5] A. Moubayed, A. Shami, P. Heidari, A. Larabi, and R. Brunner, "Edge-enabled V2X Service Placement for Intelligent Transportation Systems," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020. doi: http://dx.doi.org/10.1109/TMC.2020.2965929.

[6] ETSI, "Multi-Access Edge Computing (MEC); Framework and Reference Architecture," *ETSI ISG MEC, ETSI GS MEC 003 V2.1.1*, 2019. Online [Available]: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf.

[7] J. Marquez-Barja, B. Lannoo, D. Naudts, B. Braem, C. Donato, V. Maglogiannis, S. Mercelis, R. Berkvens, P. Hellinckx, M. Weyn, I. Moerman, and S. Latre, "Smart Highway: ITS-G5 and C-V2X Based Testbed for Vehicular Communications in Real Environments Enhanced by Edge/Cloud Technologies," *28th European Conference on Networks and Communications (EuCNC), Valencia, Spain*, pp. 1–2, 2019. Online [Available]: http://www.marquez-barja.com/images/papers/eucnc-SmartHighway.pdf.