Enabling Autonomous and Connected Vehicles at the 5G Network Edge

Estefanía Coronado*, Gabriel Cebrián-Márquez‡, and Roberto Riggio*
*Smart Networks and Services (SENSE), Fondazione Bruno Kessler, Trento, Italy
Email: {e.coronado, rriggio}@fbk.eu
‡E.T.S. de Ingenieros Informáticos, Universidad Politécnica de Madrid, Madrid, Spain
Email: gabriel.cebrian@upm.es

***Abstract*—Connected and automated vehicles currently rely on on-board resources to implement autonomous functions, leaving the mobile network for non-mission-critical applications. At the same time, the ultra-low latency, the increased bandwidth, and the softwarization and virtualization technologies of 5G systems are opening the door to multiple applications in the context of connected and automated vehicles. The deployment of applications at the edge of the mobile network under the Multi-access Edge Computing (MEC) paradigm becomes an excellent option for meeting the latency requirements imposed by connected mobility. In this context, this demonstration showcases how remote and autonomous driving applications, such as lane tracking and object detection, can be offloaded to a MEC-enabled 5G network without impairing their effectiveness, and the change in the latency perceived by end-users with respect to a cloud deployment.**

***Index Terms*—CCAM, 5G, MEC, Cloud, NFV, SDN.**

I. Introduction

While self-driving vehicles seemed a science-fiction scenario at the beginning of the century, they have become a tangible reality at the present time. These *autonomous* vehicles have introduced the need to interchange information to cooperate and ensure driving safety in what is known as Cooperative, Connected, and Automated Mobility (CCAM). Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) and Vehicle to Network (V2N) are just a few examples of models that enable intercommunication between vehicles. In this context, the ultra-low latency, increased bandwidth and inherent softwarization and virtualization technologies of 5G networks have become key enablers for the smart mobility ecosystem. As a matter of fact, 3GPP includes the support for automotive requirements from Release 14 [1].

Autonomous vehicles generate enormous amounts of data that can be used to recognize road features and external elements such as lanes, signs, and pedestrians [2]. However, the operations involving the cooperation of multiple vehicles, such as lane merging, require data processing from participant and non-participant vehicles from a global perspective. Typical computer vision tasks used for autonomous driving such as feature extraction require significant computing power that may not be available on all vehicles, e.g., low-end cars and motorbikes. However, on-board resources have been leveraged so far for implementing computer vision operations, relegating the network to infotainment applications. The advent of 5G and the Multi-access Edge Computing (MEC) paradigm, however, allows offloading computing-intensive and latency-sensitive tasks to the network edge, thus opening up new horizons for autonomous driving. Software-Defined Networking (SDN) and Network Function Virtualization (NFV) have become fundamental enablers of this paradigm where network functions are decoupled from the underlying hardware, and they are deployed and orchestrated in the form of software.

In this paper, we set to demonstrate how typical computer vision applications like lane tracking and object detection can be offloaded to a MEC-enabled 5G network without impairing the effectiveness of the application. To this aim, we have implemented a computer vision model for autonomous driving, which is executed on different scenarios as an application (known as ME App Virtual Network Function in the ETSI MEC reference architecture [3]) in the MEC Host or in a cloud data center to showcase the latency reduction experienced by a car in a small-scale track. Alternatively, the car can be driven manually emulating remote driving to allow personally experiencing the impact in both performance and latency. All the code is available under a permissive APACHE 2.0 license[1].

II. System Architecture

A. Mobile Network Design

The reference network architecture leveraged in this demo is sketched in Fig. 1. Radio access nodes are implemented using the srsLTE suite [4] and coordinated by the 5G-EmPOWER Software-Defined Radio Access Network (SD-RAN) controller [5]. This implementation decision is due to the non-existence of open-source 5G stacks at the moment of writing. The communication between the radio access nodes and the SD-RAN controller is done using the OpenEmpower protocol [6]. The Evolved Packet Core (EPC) is implemented using nextEPC [7]. Finally, following a bump-in-the-wire approach [8], the MEC Host is located midpoint between the radio access node and the EPC.

B. MEC Host Deployment

The MEC Host leveraged in this work is based on lightweight virtualization technologies such as Docker containers and Click processes [9]. The traffic routing capabilities of

[1]Online resources available at: http://5g-empower.io/.

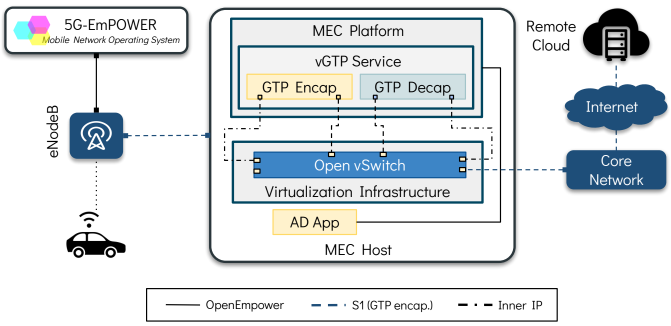


Fig. 1. High-level view of the MEC Host architecture.

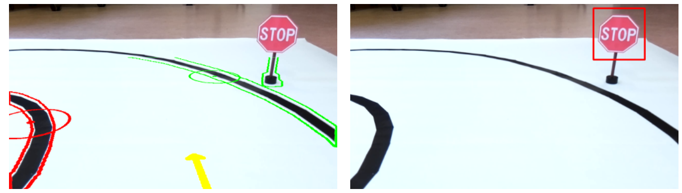
the MEC Host are provided by Open vSwitch. A Click-based process, named vGTP, analyzes the traffic between the radio access nodes and the EPC. Both the OpenFlow controller and the vGTP service are deployed as Docker containers, while other applications can run encapsulated in additional containers or as services in external machines connected physically to the MEC Host, which ensures the scalability of the system.

The traffic between radio access nodes and the EPC is intercepted by Open vSwitch for further processing by the vGTP service as follows. Control plane traffic, encapsulated using the SCTP protocol, is utilised to gather context information from the User Equipments (UEs) and the GTP-U tunnels created to exchange traffic between such UEs and the core network. Once analysed, these packets follow their path in the network. User plane traffic, encapsulated in GTP packets, is decapsulated using the aforementioned context information, and the underlying IP packets are delivered to the corresponding destination. If the packet destination is one of the applications managed by the MEC Host, Open vSwitch steers the IP packet to the corresponding virtual or physical port of such a application. If not, the packet is (re-)encapsulated in GTP packets, along with any other traffic originated by the MEC applications, so that they can reach again the mobile network.

The stateful decapsulation/encapsulation of GTP packets enables seamless communication between UEs and any service, regardless of the placement in the MEC Host or in a cloud data center. Additionally, the monitoring of the SCTP traffic allows performing efficient session and mobility management of UEs, e.g,. after a handover, the context information of the UE is updated. Note that these operations can be performed with no modifications to either the access nodes or the EPC protocol stacks, making this solution completely vendor-agnostic.

III. Autonomous Driving Application

Lane line detection and on-road object recognition algorithms represent the basis of autonomous driving. To demonstrate how these latency-sensitive operations can be offloaded to the network, we have developed a simple application, named Autonomous Driving (AD) application, comprising the two aforementioned algorithms. This application processes the video stream fed by vehicles using OpenCV [10], and returns the corresponding driving directions [11].



(a) Lane line detection. (b) On-road object detection.

Fig. 2. Deployed autonomous driving application.

A. Lane Line Detection

Lane line detection allows determining the path and the relative position of vehicles in the road. Therefore, the accuracy of this algorithm is key to ensure safe vehicle maneuvers. However, there are many factors that may hinder this accuracy, such as adverse weather conditions, worn-out roads, or dirt. It is thus necessary to process the images to minimize errors.

After applying a Gaussian filter to reduce granularity, the images are transformed to the Hue, Saturation and Lightness (HSL) color space to extract the tones of the lane. The binarization of these pictures and the application of gradient filters, including Sobel, results in the highlighting of the areas in the images that correspond to the lane lines. To determine the curvature of the lanes, the picture is perspective-transformed. Finally, the interpolation of the points conforming the lane lines are grouped and interpolated to estimate the curves. The result of this process can be seen in Fig. 2a.

B. On-Road Object Detection

This algorithm enables the detection of entities within the view of the vehicle. As in the case of lane line detection, this algorithm involves some significant challenges, such as the understanding of text plates and the recognition of non-regulated signs. For the sake of simplicity, however, the implemented application only recognizes a limited set of traffic signs, resulting in the prediction shown in Fig. 2b.

Typically, object detection has been widely performed using the so-called Haar feature-based cascade classifiers due to their ability to detect objects on the basis of the features extracted from a set of positive and negative images [12]. Once the model is trained, it can be used off-line. Given that this classifier groups the extracted features into layers, the object detection process can be terminated prematurely, which results in very low classification times. Because of these properties, this model has been used in the detection of traffic signs in the proposed application.

IV. Demonstration

In this demonstration, we illustrate how the autonomous driving application described in the above section can be offloaded from the vehicle to the network. In particular, we show the ability of the MEC paradigm to comply with the strict latency requirements imposed by applications for autonomous driving in comparison with a cloud deployment.

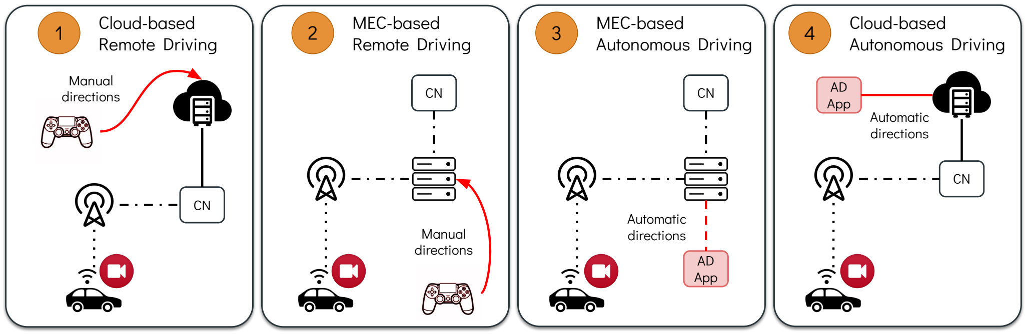


Fig. 3. Scenarios showcased in the demonstration: remote and autonomous driving at cloud-based and MEC-based sites.

Based on the system architecture shown in Fig. 1, a toy car equipped with an LTE dongle and a video camera represents a UE connected to the mobile network requesting driving directions in a small-scale track. Using this setup, four different scenarios are showcased, as depicted in Fig. 3. On the one hand, *Scenario 1* and *Scenario 2* aim to demonstrate a remote driving use case, allowing users to experience the latency and performance that would be obtained when issuing driving directions from a MEC Host or a cloud site, respectively. On the other hand, in *Scenario 3* and *Scenario 4* the described autonomous driving application is deployed as a Docker container on the MEC Host and on the remote data centre, respectively. The application receives the video stream sent by the vehicles connected to the access node, processes the images, and returns the corresponding driving directions. The autonomous functionality allows observing the behaviour of the vehicle depending on the time taken to receive the instructions. For example, if there is high delay from the cloud site, the emergency break is performed too late and the vehicle gets off track.

V. Conclusions

Virtualization and network sofwarization technologies have transformed the operation of mobile networks. The interplay between MEC and NFV enables not only advanced customization but also a radical latency reduction. With these enablers in mind, this demonstration have shown how latency-sensitive computing operations can be offloaded to the network edge. To this end, we have deployed a cloud-native application for autonomous and remote driving on a MEC Host, and we have demonstrated the significant gains obtained by comparing the response time and the user experience with the deployment on a cloud site. The demonstration is based on open-source software and platforms, thus evoking the research community's interest and collaboration in this challenging field.

Demo Requirements

The equipment consists of an Intel NUC connected to an Ettus Research Universal Software Radio Peripheral (USRP) b210, two laptops, one Ethernet switch and a toy car equipped with a video camera and an LTE dongle. The setup fits on a regular table, and an additional space on the floor of roughly four square meters. The setup time is lower than two hours.

Acknowledgements

This work has been performed in the framework of the European Union's Horizon 2020 project 5G-CARMEN co-funded by the EU under grant agreement No 825012. The views expressed are those of the authors and do not necessarily represent the project. The Commission is not liable for any use that may be made of any of the information contained therein. In addition, this work has been supported by the Spanish Ministry of Science, Innovation and Universities (RTI2018-098156-B-C52, FEDER funds), and the Spanish Regional Government of Castilla-La Mancha (SBPLY/17/180501/000353).

References

[1] 3GPP, "TR 23.711. Enhancements of Dedicated Core Networks selection mechanism (Release 14)," Sep. 2016.

[2] A. Uçar, Y. Demir, and C. Güzeliş, "Object Recognition and Detection with Deep Learning for Autonomous Driving Applications," *Simul.*, vol. 93, no. 9, Sep. 2017.

[3] European Telecommunications Standards Institute (ETSI), "Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV Environment," *ETSI GR MEC 017*, Feb. 2018.

[4] "Software Radio Systems (SRS)," Accessed on 03.04.2020. [Online]. Available: http://www.softwareradiosystems.com/

[5] E. Coronado, S. Kahn, and R. Riggio, "5G-EmPOWER: A Software-Defined Networking Platform for 5G Radio Access Networks," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 715–728, Jun. 2019.

[6] "The OpenEmpower Protocol," Accessed on 03.04.2020. [Online]. Available: https://github.com/5g-empower/5g-empower.github.io/wiki/EmPOWER-Protocol

[7] "NextEPC," Accessed on 03.04.2020. [Online]. Available: http://nextepc.org/

[8] T. Subramanya, G. Baggio, and R. Riggio, "lightMEC: A Vendor-Agnostic Platform for Multi-access Edge Computing," in *2018 14th International Conference on Network and Service Management (CNSM)*, Rome, Italy, Nov. 2018, pp. 198–204.

[9] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router," *ACM Trans. Comput. Syst.*, vol. 18, no. 3, pp. 263–297, Aug. 2000.

[10] "OpenCV Library," Accessed on 03.04.2020. [Online]. Available: https://opencv.org/

[11] E. Coronado, G. Cebrian-Marquez, and R. Riggio, "Enabling Computation Offloading for Autonomous and Assisted Driving in 5G Networks," in *Proc. of IEEE Globecom*, Waikoloa, HI, USA, 2019.

[12] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. I, Dec. 2001, pp. 511–518.