Modelling Edge Computing in Urban Mobility Simulation Scenarios

Katja Gilly, Salvador Alcaraz
Department of Computer Engineering
Miguel Hernandez University
Elche, Spain
{katya,salcaraz}@umh.es

Noura Aknin
Department of Physics
Abdelmalek Essaadi University
Tetouan, Morocco
noura.aknin@uae.ac.ma

Sonja Filiposka, Anastas Mishev
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Skopje, North Macedonia
{sonja.filiposka,anastas.mishev}@finki.ukim.mk

***Abstract*—One of the main challenges in edge computing service management is to continuously maintain a high level of quality of service for the end-user mainly in terms of low latency. This can only be achieved with intelligent management of the edge infrastructure virtualised resources making sure that the edge services are always hosted on resources located as close to the end-user as possible. Thus, the problem of resource allocation and service migration in edge computing is intricately related to the user location and mobility pattern. Therefore, it is of great importance to analyse the edge services performance of proposed management techniques in an urban scenario with realistic mobility model. For these purposes, in this paper we discuss the problem of edge computing simulation and describe an integrated solution environment based on the combination of two proven simulators for datacentre services and urban mobility modelling.**

***Index Terms*—mobile computing, simulation, performance measures, modelling**

I. INTRODUCTION

The growing demand for high speed networking and low latency real-time services started a paradigm shift in the design and architecture of next-generation solutions resulting in the 5G ecosystem vision [1]. At the core of the new 5G landscape is an all encompassing architecture that ensures access to services from all types of devices anytime, anywhere [2].

Most of the service categories promoted as the 5G flagship services such as real time high resolution video services or autonomous driving are not only demanding high performance networking, but also require lots of computing power. These requirements have resulted in a synergy between the Multi-Access Edge Computing (MEC) architecture [3] and 5G networks. By following the MEC architecture idea and locating computing and storage resources close to the 5G network access points, the low latency targets can be reached.

The design and performance of the multi-access edge orchestrator solution [4] has raised a vibrant interest in the research community and there are already quite a few proposals that deal with the problems of service offloading, resource management, algorithms for dynamic service placement and migration, delay optimisation, high quality of service policies management, etc. The common first step in all of these related problems is to analyse the performances of the proposed solution in a simulated environment. For these purposes, a number of edge computing simulators have been proposed. The goal of this paper is the creation of simulation scenarios that will realistically reflect the use of edge computing in a 5G environment. We focus on dense scenarios with different actors: pedestrians, different vehicle types including bicycles, cars and public transportation that move according to the typical rules established in an urban environment. From that point of view we discuss the possibility of creating realistic simulation scenarios for 5G MEC services by using a combination of well known and proven tools: the urban traffic simulator Simulation of Urban Mobility (SUMO) [5] and the cloud management simulator CloudSim [6] extended with support for MEC services. This combination enables researchers to focus on their MEC oriented work and yet reap the benefits of the results analysis based on complex mobility scenarios.

II. RELATED WORK

There are several options to simulate the behaviour at the edge of the network. Some are specific for fog environments while others are based on cloud infrastructure simulation extended for multi-tier scenarios. This is the case of fog and edge computing simulators such as iFogSim, EdgeCloudSim and IOTSim, all based on the popular simulator CloudSim, already proven for federated datacentres and cloud scenarios. iFogSim [7] extends CloudSim with additional classes that model the behaviour of sensors and actuators that represent Internet of Things (IoT) devices. EdgeCloudSim [8] includes the implementation of nomadic mobility and dynamic network modelling to manage edge-resources via its own edge orchestrator module. IOTSim [9] focuses on application processing modules using the MapReduce model for concurrent processing of data collected from IoT devices. However, none of them support the creation of complex mobility modelling for the end-devices, using only simplified random mobility models such as Random Walk or Random Direction.

There are also specific fog simulators that do not cover cloud implementation like FogNetSim++, based on OMNet++, that includes the implementation of several mobility models

Part of this work is funded by ”Cooperacion Universitaria al DEsarrollo 2018-Universidad Miguel Hernandez de Elche” project, convenio UMH-GV.

Annex to ISBN 978-3-903176-28-7 2020 IFIP

and fog communication protocols, or FogTorchII [11] that is a prototype Monte Carlo based simulator for development of QoS-aware applications in a fog infrastructure.

In order to simulate a vehicular network scenario connected to a cloud and edge infrastructure, we also need realistic mobility models implemented on actual urban maps. An editable open map of the world is supplied by the online tool named OpenStreetMap [13], that can be used as the input to mobility simulators such as SUMO. It provides an on-demand modelling to generate a road traffic simulation based on the imported map. SUMO has already been associated with network simulators [12], such as the project iTetris that connects SUMO with NS-3 to implement vehicle to vehicle and vehicle to infrastructure communication technologies or TraCI that links SUMO with OMNet++ and MATLAB. Similar approach that combines network and traffic simulators is Veins, a vehicular network simulation framework that uses TraCI and OMNet++. VANETsim 2.02 is another simulator for vehicle mobility based on NS-3 and is focusing on the implementation of security and privacy concepts. These tools are mostly oriented towards the performance study of communication protocols on vehicular networks.

When studying the resource and service management from both computing and mobility points of view, these tools are not adequate because they either lack detailed edge/network infrastructure implementation for application lifecycle management, or detailed mobility models for realistic urban scenarios.

III. SIMULATING VEHICULAR MEC SERVICE ORCHESTRATION

The problem of service orchestration in MEC is to be tackled by the Multi-access Edge computing Orchestrator (MEO) as described in the ETSI MEC framework architecture [3]. The MEO is responsible for orchestrating the workflows of creation, change and deletion of user requested services while working closely with MEC platform managers. While the MEO has a system-wide view, the MEC platform manager is in charge of a single cluster of server hosts, i.e. micro datacentres, and directly interfaces with their infrastructure virtualisation manager. The complete MEC system can be viewed as a collection of geographically distributed sets of virtual resources, each managed by a separate MEC platform manager, all orchestrated together via the MEO, see Fig. 1.

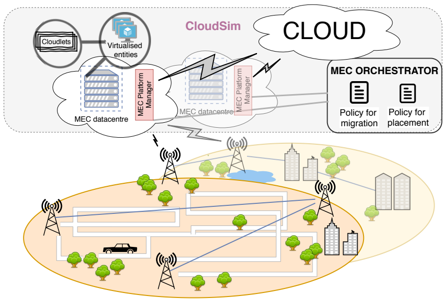


Fig. 1: Example MEC infrastructure deployment.

When combining the MEC system with 5G, the geographical distribution of micro datacentres is aligned with the distribution of 5G base stations. Each base station is directly attached to its own MEC micro datacentre on which end-user services are offloaded for maximum efficiency. Depending on the user density and size of the coverage area for a given base station, the size of its attached MEC micro datacentre can vary.

The end-user mobility will trigger a base station handover event in the moment when the user exits the coverage area of its current base station and enters the coverage area of a different base station. This handover event in the 5G network, should also trigger the MEO that will try to migrate all user's active MEC services to the nearest, in network latency terms, MEC micro datacentre based on the new user location. The migration process will involve relocation of MEC services from one MEC management platform to another, preferably collocated with the new base station in use by the user. In the case this is not possible due to limited available resources at the desired MEC management platform, the MEO needs to decide whether there is another potential location for the MEC service that will improve the experienced latency, or the service will continue to reside on its original platform until resources at a more optimised location become available.

The high-level resource management functionalities of the MEO are, thus, one of the most important ones and are invoked during service creation and service optimisation. This is the problem of initial service placement, where the MEO needs to decide on which MEC platform manager to place the requested new service, and the problem of service migration whenever the user mobility results in a latency increase and lower performances, respectively. A number of papers in the research literature are tackling these two problems by suggesting strategies and algorithms that will ensure high quality of experience for the end-user [15].

To be able to simulate this behaviour of the MEC orchestrator and its MEC platform managers, the simulator needs to support the creation of the clusters of hosts that act as micro datacentres and the MEC infrastructure network that interconnects them. Different strategies for virtualisation of these resources should also be available. For the logic of the MEO and platform managers to be implemented, the simulator must support the possibility to be extended with custom algorithms for service placement and migration and provide resource usage reports. Based on these requirements, we have chosen CloudSim as the simulator of choice.

A. Modelling MEC services with CloudSim

The MEO main functionalities can be implemented by extending the virtual machine (VM) initial placement and VM migration policy classes with customised algorithms. By choosing to work with the already available host types within CloudSim, the simulator also offers the capability to track the energy consumption of the MEC infrastructure which is an added bonus and requires no additional extensions or coding. The MEC infrastructure can be built by instantiating clusters of server hosts that will serve as the MEC micro datacentres.

Each host is defined with its operating system platform and its available physical resources in terms of CPU cores, RAM size, disk storage and network bandwidth.

The simulator supports the creation of clusters of different sizes, and its integrated networking elements enable designing each cluster as a set of hosts attached to an edge switch. It is assumed that each hosts cluster, or each edge switch, is directly connected to a corresponding base station. By instantiating aggregation and core switches, but also WAN links if necessary, one can recreate any type of networking infrastructure including parameters such as bandwidth and latency, which then affect the migration times.

VMs can be instantiated on the physical hosts, based on the amount of virtual resource that will be consumed and consumption strategy: time sharing or space sharing. The end-user services are defined as Cloudlets running in VMs, defined with start/end time and type of resource consumption.

The customised main simulation flow performs the initial setup of hosts and network elements and the scheduling of the simulation events in terms of starting and ending Cloudlets and events for handover notifications. Events can be defined in an external input file that needs to contain all necessary details such as relevant actor ids, event type and timestamp.

The CloudSim output is a trace file that serves as a log for the actions that are taking place in the defined MEC infrastructure. Each event is logged with a possibility for variable level of details. These include the complete process of VM migration with start time stamps and end time stamps, but also information on no migration events. Finally, logs contain information about the power consumption during the simulation, as well as additional detailed information about the resources utilisation of the MEC infrastructure.

B. Adding Urban Mobility Using SUMO

When the goal of the simulation is to analyse the performance of the proposed MEC service orchestration logic in a 5G urban environment, using simple mobility models is not sufficiently realistic. Thus, we have decided to move away from generic random mobility models, to a realistic specialised urban vehicular simulator that will enable the representation of the mobility of end-users in an urban area of choice. Based on these requirements, for modelling mobility we have chosen SUMO, being a free, scalable simulator that works with real OpenStreetMaps information about the area wherein the traffic is simulated. SUMO supports different types of vehicles including pedestrians, bicycles, public transportation such as buses and trams, cars, emergency vehicles, etc. The simulation area is described using the OpenStreetMaps already available information including speed limit, one way streets, pedestrian crossings, stop lights, precedence traffic rules, etc.

When defining the SUMO simulation one can also control total simulation time, types of vehicles and their frequency of occurrence, or minimum trip lengths per vehicle. More details to the simulation definition can be added: vehicle start location, maximum parking duration, etc. In other words, SUMO is an urban mobility simulator that focuses on creating realistic urban vehicular scenarios in great detail which enables defining scenarios that can match many of the MEC + 5G special use case services including emergency services.

C. Information flow between CloudSim and SUMO simulators

The SUMO simulator was used to simulate the mobility patterns of the MEC users, and its output trace file was then post-processed in order to use it as the basis for the external input files fed to the CloudSim simulator.

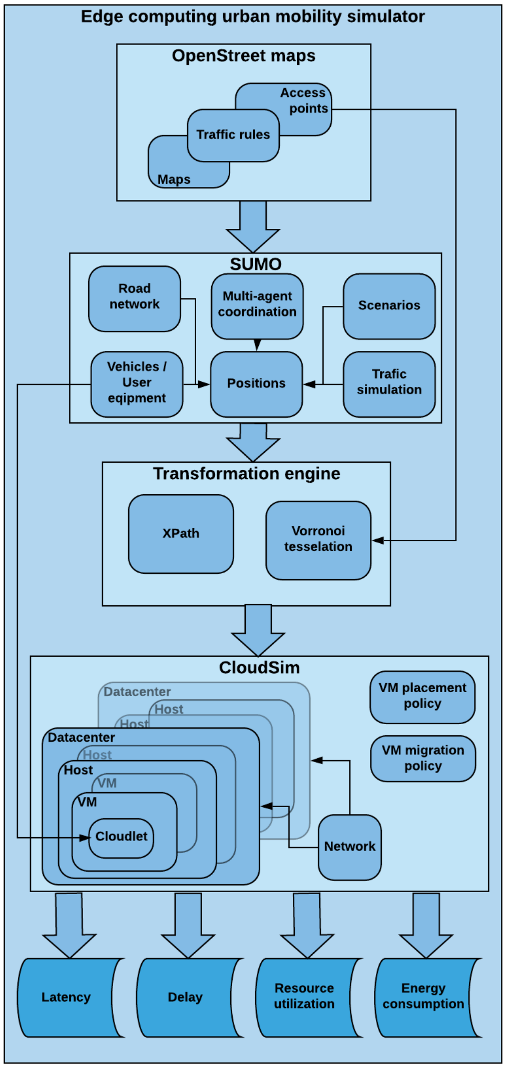


Fig. 2: Integration.

The high level design of the integrated simulation environment is shown in Fig. 2 along with their logical connections and mapping. The SUMO output is processed by the transformation engine, preparing it for the CloudSim simulator.

The transformation engine post-processing step creates the glue layer between the two simulators. In addition to the SUMO output trace, the geolocation of the 5G base stations that cover the simulated area of interest are also provided to the engine. The main logic involved is to analyse the provided vehicle mobility trace and augment the available data with information about the active base station that covers the area wherein the vehicle is located. This is done using Voronoi tessellation [16], where each point on the OpenStreetMap is assigned to the nearest base station.

Once all locations in the SUMO trace output are assigned to a base station, the second part of the post-processing engine work is to transform the information into the definition of corresponding cloudlets and migration events. For these purposes different mappings can be used based on the vehicular type and post-processing settings such as assigning one cloudlet per vehicle, or assigning multiple cloudlets for vehicles such as buses. The rest of cloudlets' characteristics can be uniform or randomised, or again typified based on vehicular type.

The final output of the transformation engine is the creation of the external input files for CloudSim: the definition of cloudlets and their creation and termination timestamps, accompanied with the base station id where they initially appeared; and the scheduling of handover events where the corresponding vehicle new location belongs to the area served by a different base station compared to the previous location. Care must be taken that the elements ids correspond, so that the final stitching between the two simulators is achieved.

The proposed approach can be extended to include the 5G wireless communication. To create a simulation of the complete ecosystem the same methodology can be applied by combining CloudSim with a wireless network simulator. We propose the use of SimuLTE with Veins [14], to build on the SUMO mobility capabilities and provide the missing layer.

IV. EXAMPLE USE-CASE RESULTS ANALYSIS

By combining the output results from both the SUMO and the corresponding CloudSim simulation, location based performance analysis can be conducted related to the MEC infrastructure and its resource usage. For these purposes, we have conducted a number of simulation scenarios with a variable set of parameters, see Table I.

TABLE I: SUMO + CloudSim simulation parameters

Parameter	Value
OpenStreetMap area	Alicante city centre, Spain
simulation time	10,000 s
min path length	200 m
vehicles arrival time	1.2 per second
vehicle type	passenger car
pause time	0
number of base stations	9
number of hosts per base station	5 or 9
base stations location	co-located with touristic attractions
virtual resource usage type	space sharing
MEC network	3 layer fat-tree
MEC network switches	9 edge, 3 access, 2 core
MEC network latencies	1.57 ms, 2.45 ms, 2.85 ms
number of server hosts	45 or 81
server host type	0: 8 cores, 16 GB RAM, 100 Mbps 1: 16 cores, 32 GB RAM, 100 Mbps
initial placement policy	hierarchical communities
migration policy	follow-me
VM characteristics	2 cores, 2 GB RAM, 100 Mbps
Cloudlet complexity	100% of CPU time used
Vehicle to cloudlet mapping	1-to-1

The use case is based on the idea of interactive touristic guide available in the vehicles moving throughout the city centre of Alicante, Spain. In this case, each vehicle is considered to be a passenger car, mapped to only one cloudlet. The whole city centre is covered using 9 base stations, collocated with main touristic attractions, where the highest density of users is expected. The provider's network is designed to be a 3 layer, fat-tree, most commonly used in such models.

Aiming to create a more demanding simulation, we have used the space sharing approach for virtual resources usage type. As the vehicles move, a follow-me migration policy is used to minimise the distance between the moving vehicle and the MEC datacentre hosting its VM.

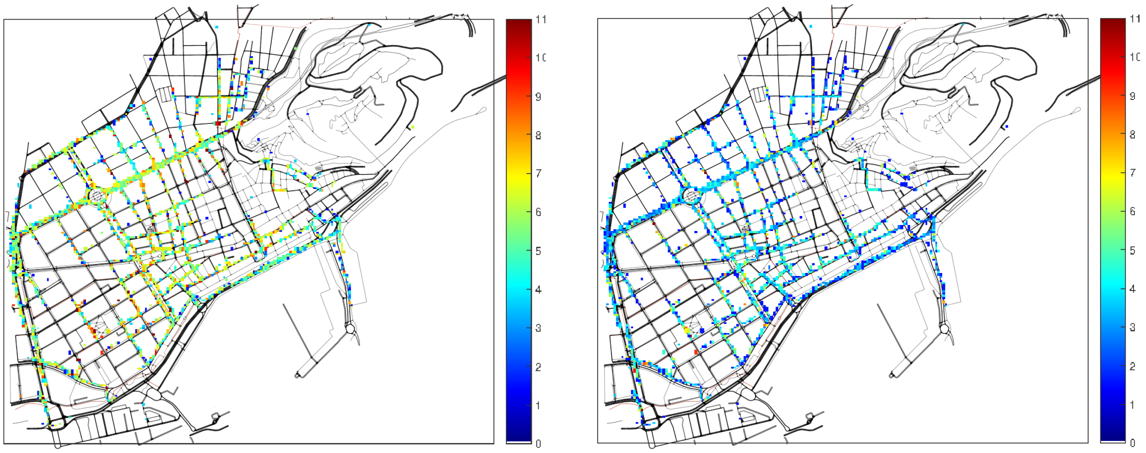
An example of the usage of the integrated simulation environment is shown in Fig. 3, where the average delay that is due to the network access in the MEC infrastructure is presented in each geolocation that has been reported as active during the SUMO simulation. The figure represents the average delay for four different cases that shows how the MEC infrastructure deals with the demands of 8629 vehicles when it is equipped with different number and types of hosts.

The presented results help locate the spots in the area where less than optimal delay is experienced. Namely, the high average delay in the first scenario is mainly due to failed migration attempts because of lack of resources at the newly connected base station during the handoff process. The migration procedure is very demanding resource-wise because CloudSim implements live migration of the VMs in order to ensure zero downtime for the end-user. However, live migration entails double usage of resources at the source and destination hosts. The source host resources will be freed only after the migration process has finished successfully. As the pool of available resources in the MEC infrastructure increases either by changing the host type or by changing the number of hosts, the average delay across all locations decreases. The few points with less than minimum average delay that are left in the final figure show that there is a problem with user density in the areas and that a higher number of base stations with smaller coverage area or an uneven distribution of server nodes across the network should also be considered to ensure minimum latency throughout the simulated area.

V. CONCLUSION

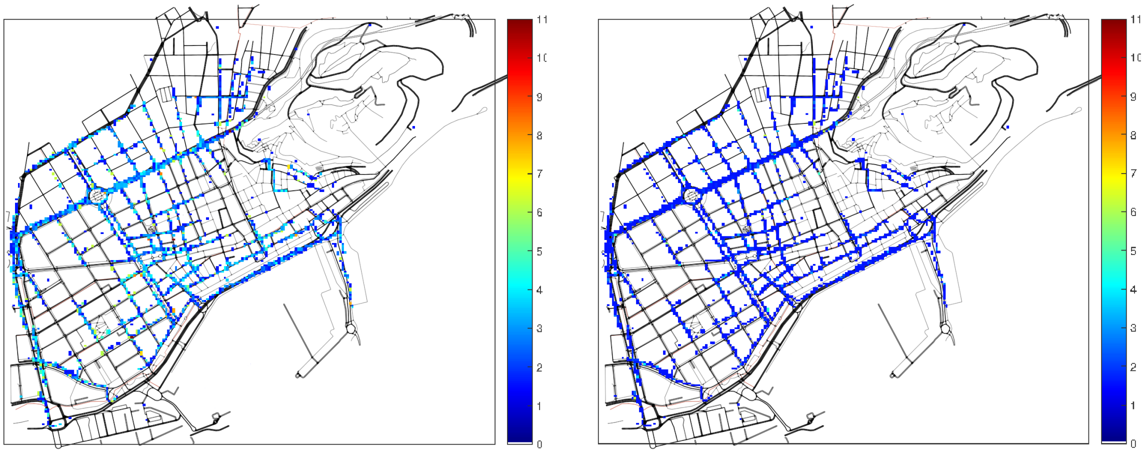
In this paper we discuss the benefits of using combination of proven, powerful simulators to create large scale simulations of MEC systems as a part of the 5G ecosystem.

We extended the capabilities of the CloudSim simulator with MEC orchestration functionalities and modelled realistic urban vehicular simulations with SUMO. By adding a transformation engine that maps between these two simulators, the SUMO output can be used as input to CloudSim and use the information about the user location relative to the MEC infrastructure to effectively implement a "follow-me" MEC services migration. The example results section provides a glimpse into the possibilities for advanced visualisation obtained from the combined output of both simulators.



(a) 8629 vehicles, 45 hosts, type 0 server

(b) 8629 vehicles, 45 hosts, type 1 server



(c) 8629 vehicles, 81 hosts, type 0 server

(d) 8629 vehicles, 81 hosts, type 1 server

Fig. 3: Combined output results showing the average service delay (colorscale in ms) experienced in each user location.

REFERENCES

[1] S.Y. Lien, C.C. Tseng, I. Moerman, and L. Badia, ”Recent Advances in 5G Technologies: New Radio Access and Networking,” Wireless Communications and Mobile Computing, 2019.

[2] L. D’Acunto, J. W. Kleinrouweler, J. Panneman, A. Gabriel, A. Adhikari, and R. Satta, ”Prosuming live multimedia content in 5G-enabied smart cities,” In Proceedings of the 10th ACM Multimedia Systems Conference, ACM, 2019, pp. 312–315.

[3] ETSI GS MEC 003 V2.1.1, ”Multi-access Edge Computing Framework and Reference Architecture,” ETSI Group Specification, 2019

[4] ETSI, ”MEC in 5G networks,” ETSI White Paper No. 28, 2018

[5] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, ”Recent develpment and applications of SUMO-Simulation of Urban MObility,” Int. Adv. Syst. Meas., vol. 5, nos. 3-4, 2012, pp. 128–138.

[6] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. 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, 2011, pp. 23–50. doi:10.1002/spe.995

[7] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, ”iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments,” Softw Pract Exper, vol. 47, 2017, pp. 1275–1296. https://doi.org/10.1002/spe.2509

[8] C. Sonmez, A. Ozgovde, C. Ersoy, ”EdgeCloudSim: An environment for performance evaluation of edge computing systems,” Trans Emerging Tel Tech, vol. 29:e3493, 2018. https://doi.org/10.1002/ett.3493

[9] X. Zeng, S. K. Garg, P. Strazdins, P. P. Jayaraman, D. Georgakopoulos, R. Ranjan, ”IOTSim: A simulator for analysing IoT applications,” Journal of Systems Architecture, vol. 72, 2017, pp. 93–107. https://doi.org/10.1016/j.sysarc.2016.06.008.

[10] T. Qayyum, A. W. Malik, M. A. Khan Khattak, O. Khalid, and S. U. Khan, ”FogNetSim++: A Toolkit for Modeling and Simulation of Distributed Fog Environment,” in IEEE Access, vol. 6, 2018, pp. 63570–63583. https://doi.org/10.1109/ACCESS.2018.2877696

[11] B. Marques, I. Machado, A. Sena, and M. C. Castro, ”A Communication Protocol for Fog Computing Based on Network Coding Applied to Wireless Sensors,” 2017 International Symposium on Computer Architecture and High Performance Computing Workshops, Campinas, 2017, pp. 109–114. https://doi.org/10.1109/SBAC-PADW.2017.27

[12] L. Wang, R. Iida and A. M. Wyglinski, ”Vehicular Network Simulation Environment via Discrete Event System Modeling,” in IEEE Access, vol. 7, 2019, pp. 87246–87264. https://doi.org/ACCESS.2019.2922766

[13] https://www.openstreetmap.org (accessed on 28th of November, 2019)

[14] G. Avino, M. Malinverno, F. Malandrino, C. Casetti, C. F. Chiasserini, G. Nardini, and S. Scarpina, ”A simulation-based testbed for vehicular collision detection,” in 2017 IEEE VNC, 2017, pp. 39–40.

[15] Z. Rejiba, X. Masip-Bruin, and E. Marn-Tordera, ”A survey on mobility-induced service migration in the fog, edge, and related computing paradigms,” ACM Computing Surveys (CSUR) vol. 52, no. 5, 2019.

[16] Q. Du, M. Emelianenko, and L. Ju, ”Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations,” SIAM journal on numerical analysis, vol. 44, no. 1, 2006, pp. 102–119.