Sarrigiannis, I., Contreras, L. M., Ramantas, K., Antonopoulos, A. y Verikoukis, C. (2020). An Intelligent Edge-based Digital Twin for Robotics. *IEEE Network,* 34 (5), pp. 120-126.

DOI: https://doi.org/10.1109/MNET.111.2000476

# Fog-enabled Scalable C-V2X Architecture for Distributed 5G and Beyond Applications

Ioannis Sarrigiannis, Luis M. Contreras, Kostas Ramantas, Angelos Antonopoulos and Christos Verikoukis

## ABSTRACT

**The Internet of Things (IoT) ecosystem, as fostered by fifth generation (5G) applications, demands a highly available network infrastructure. In particular, the internet of vehicles use cases, as a subset of the overall IoT environment, require a combination of high availability and low latency in big volumes support. This can be enabled by a network function virtualization architecture that is able to provide resources wherever and whenever needed, from the core to the edge up to the end user proximity, in accordance with the fog computing paradigm. In this article, we propose a fog-enabled cellular vehicle-to-everything architecture that provides resources at the core, the edge and the vehicle layers. The proposed architecture enables the connection of virtual machines, containers and unikernels that form an application-as-a-service function chain that can be deployed across the three layers. Furthermore, we provide lifecycle management mechanisms that can efficiently manage and orchestrate the underlying physical resources by leveraging live migration and scaling functionalities. Additionally, we design and implement a 5G platform to evaluate the basic functionalities of our proposed mechanisms in real-life scenarios. Finally, the experimental results demonstrate that our proposed scheme maximizes the accepted requests, without violating the applications' service level agreement.**

## INTRODUCTION

The Internet of Things (IoT) ecosystem is a collection of billions of devices, such as sensors, that are connected among them and with the Internet. According to Ericsson's mobility report, the 10.8 billion IoT connections of 2019 are expected to reach the number of 24.9 billion by the end of 2025, which means a compound annual growth rate (CAGR) of 15 percent [1]. When it comes to cellular networks, and as the fifth generation of wireless communications (5G) is being gradually introduced, the amount of 1.3 billion cellular IoT connections of 2019 is expected to experience an even higher CAGR of 25 percent, reaching the number of 5 billion by the end of 2025.

Vehicle-to-everything (V2X) communications is a major area of IoT that will enable communication between vehicles and between vehicles and infrastructure. The cellular V2X (C-V2X) application layer model includes the vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) and vehicle-to-network (V2N) operational modes that require a combination of high reliability and low latency. C-V2X was initially introduced by the 3rd Generation Partnership Project (3GPP), under the Release 14 [2], using LTE-based radio access network (RAN) for V2X communications, while Release 16 is anticipated to feature 5G support for the V2X services.

The national highway traffic safety administration of the United States (U.S.) predicts that by fully adopting only two V2X safety applications, 1000 lives per year could be saved and a half million crashes could be prevented in the U.S., while a reduction of 14 percent of the global greenhouse emissions, due to transportation, could be also achieved [3]. Therefore, it is imperative to create an intelligent transportation system (ITS), or an internet of vehicles (IoV) environment, in order to provide crucial and non-crucial services.

Traffic safety, reduced air pollution and regulation of vehicle traffic flows, are only few examples for improving the quality of life. Traffic management applications, for example, could collect real-time weather information from road condition sensors, such as surface conditions (e.g., temperature, humidity, salinity, and so on.). A traffic safety application will have to take into consideration the aforementioned metrics and, along with sensors on the vehicles (e.g., proximity or break sensors), issue an action that could be a warning (e.g., information

*Ioannis Sarrigiannis and Kostas Ramantas are with Iquadrat Informatica; Luis M. Contreras is with Telefónica I+D / Global CTIO Unit; Angelos Antonopoulos and Christos Verikoukis are with CTTC/CERCA.*

for cooperative road safety), or an immediate action (e.g., emergency break for collision avoidance).

Depending on the service, V2X crucial services (e.g., autonomous driving) can tolerate a maximum latency between 10 ms and few seconds, while non-crucial services (e.g., vehicle software update) can tolerate up to few minutes of latency [4]. LTE RAN is unable to support such low latency values in big volumes, which makes the 5G RAN the perfect enabler for C-V2X applications [5]. The majority of the published works propose edge computing solutions where processing power becomes available at the edge of the network. These proposals often refer to multi-access edge computing (MEC)-enabled architectures [6], where processing power is offered both at the core and edge network. Moreover, applications, in accordance with the network function virtualization (NFV) paradigm (i.e., the virtualization of the physical infrastructure), are considered as virtual network functions (VNFs) that can be executed at these two layers [7]. While these solutions are able to offer the required low latency for many C-V2X applications to function properly, they struggle to guarantee the ultra-low latency requirements (i.e., equal or less than 10 ms) in big volumes for some crucial services [5].

Addressing some of the core and edge limitations, fog computing "distributes computing, storage, control and networking functions closer to the users, along a cloud-to-thing continuum" [8]. Fog takes advantage of the infrastructure that lies along the cloud-to-thing path, such as servers, routers, switches, vehicles and smartphones. These devices, previously limited to operate autonomously, can now participate in a connected environment that efficiently manages their resources. Therefore, the ability to leverage all the available resources, especially at the thing proximity, can guarantee the ultra-low latency in big volumes that the C-V2X applications require.

Another significant topic that has caught both industry's and academia's attention is the VNF placement problem in such distributed architectures. On the one hand, there are various works that try to tackle the placement problem within a generic MEC-enabled environment considering a VNF either as a virtual machine (VM) [7], an environment that fully virtualizes a physical computer, or a container [9], a lightweight virtual environment that enables application-level virtualization. Compared with unikernels, an even more shrunken virtualization environment that contains only the minimum amount of operating system (OS) services, kernel and libraries for a specific application to run, VMs and containers could add significant delay to the IoV infrastructure, making them inappropriate for ultra-low latency C-V2X services.

On the other hand, there are very few works focusing on the service placement, specifically for C-V2X communications. The authors in [10] consider the C-V2X service placement problem in a MEC architecture, taking into consideration the computational resource availability at the nodes, offering a low-complexity greedy-based heuristic algorithm in order to solve this problem. Yet, they are limited to a two-layer architecture and their building block is considered to be the VM. Furthermore, the authors in [11] provide a fog-enabled platform in order to support the distribution of IoV applications. While they leverage the fog infrastructure, they are also limited to the container as their virtualization environment.

Within the context of the fog computing paradigm, we propose a new 3GPP compliant fog-enabled architecture with three different layers of processing power; the core, the edge and the vehicle. Our architecture, apart from the V2N communications, supports direct V2V communications, where, combined with the processing power at the vehicle, enables the execution of ultra-low latency applications, saving, at the same time, resources for the V2N communications. A distributed application model is adopted, where applications consist of virtual environments (i.e., VNFs) that can run as VMs, containers or unikernels. The combination of one or more heterogeneous virtual environments, distributed across the three different processing layers, creates an application-as-a-service function chain (AaaSFC).

Supporting this architecture, we explain two important lifecycle management (LCM) functionalities, the live migration and the horizontal scaling of the VNFs. Furthermore, we provide an experimental 5G platform implementation, based on open source software and common hardware, while we enhance the shortcomings of an open source NFV orchestrator (NFVO) in order to support migration decisions. Additionally, we present four distinctive

examples of C-V2X use cases and we provide a comprehensive guide that elaborates the main properties of the three different virtualization environments. To the best of our knowledge, there is no other work that assumes such distributed architecture, utilizing heterogeneous virtual environments, in order to deploy, place and manage a distributed AaaSFC within the premises of a three-layered fog-enabled environment for C-V2X use cases, supported by experimental results that are based on an experimental platform implementation.

## C-V2X USE CASES

The IoV aims to provide crucial and non-crucial applications for an ITS. While in the V2V, V2N and V2I operational modes the 5G base station (gNB) can be used as the communication hub, there is also the option for direct communication between the vehicle and another vehicle or device. This direct channel can be used when the latency requirements of a service would be violated if the intermediate gNB was used, or to save infrastructure and network resources. In this section we will describe four C-V2X use cases, as well as their latency requirements, based on [[4],[12]].

### Vehicle type warnings

A vehicle type warning service could be on emergency situation to create awareness of the presence of emergency vehicles in proximity. This application allows a vehicle of an emergency service (e.g., ambulance, police car, etc.) to indicate its presence (event). The emergency vehicle notifies the vehicles on its path about its presence (notification) and the drivers should be warned to clear the road (action). This will save important time for the emergency services to reach their destination. The maximum latency for such application is 100 ms, from the occurrence of the event until the notification of the involved vehicles [12].

### Co-operative road safety

An emergency breaking application could be considered as a co-operative road safety service. The vehicle that uses a hard break (event) signals the hard breaking to the following vehicles (notification) in order to notify them for possible collision. Based on the speed, distance and road conditions, the following vehicles should decide if they should slow down in order to avoid a possible collision (action). The maximum latency for such application is 100 ms, from the occurrence of the event until the decision for action of the involved vehicles [12].

### Navigation and traffic jam avoidance

A map download and update application can be considered as an infotainment service. The vehicle requests Internet access in order to download the required map and find the best route, based on its current location and destination. The maximum latency for such application is 500 ms [12]. Traffic jams normally happen in a long time period. In case a traffic jam occurs (event) on the navigation's predefined route, depending on the setting, a delay of 2 s, in the case of urban areas, up to a few minutes, in the case of rural areas, can occur for the vehicle to get notified (notification) and change its route (action) [4].

### Autonomous driving

An autonomous driving application is enabled by high definition (HD) sensor sharing. It provides mechanisms for vehicles to share HD sensor data, such as lidar that measures the distances by illuminating targets with laser light and HD cameras, in order to enable better coordination for platooning and intersection management. An emergency vehicle that has to make a turn into an intersection (event) signals its course and speed to all nearby vehicles (notification) in order to adjust their speed to give priority to the emergency vehicle (action). The maximum latency that such application can suffer cannot exceed 10 ms, from the occurrence of the event until the actions taken by the involved vehicles [4].

## DISTRIBUTED HETEROGENEOUS APPLICATION MODEL

In accordance with the NFV paradigm, a distributed application model is adopted, where applications are hosted in VNFs. VNFs can run as VMs, containers or unikernels, while their combination creates an AaaSFC. In a traditional environment, the memory is divided in the kernel and the user space, for memory and hardware protection. The OS and kernel are executed in the kernel space, while the libraries and the application software are running in the user space.

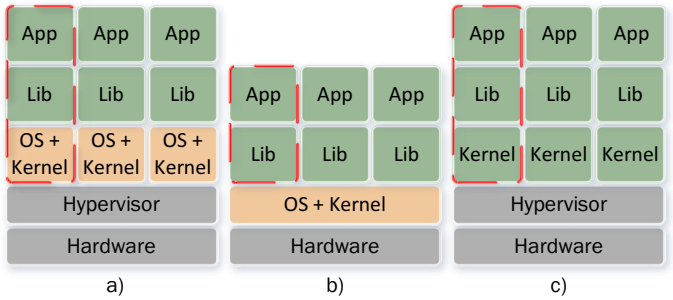



*Figure 1 - Comparison of a) virtual machine, b) container and c) unikernel system architecture*

A virtual environment follows similar principles. In some cases, though, a hypervisor is used for the virtualization of the underlying physical resources or the kernel is deployed to the user space. Figure 1 demonstrates the system architecture of each virtual environment, namely the VM, container and unikernel. The grey color illustrates the hardware or hypervisors, the orange indicates the kernel space and the green designates the user space.

### Virtual Machines

VMs are virtual environments that act as virtual computers with their own central processing unit (CPU), memory, storage and network interfaces, and have their own OS and kernels. A VM includes both user and kernel space, while the virtualization and allocation of the underlying physical resources is enabled by a hypervisor software (Figure 1-a). VMs can support resource demanding applications and isolate them properly over the same infrastructure, but their instantiation time could be up to few minutes, depending, among other factors, on the footprint of the VM.

### Containers

Containers can also provide an isolated environment for the applications and their runtime dependencies to run, using the common infrastructure. Contrary to the VMs, containers include only the user space where the applications and the library files reside, while the kernel space is shared (Figure 1-b). Containers are more lightweight, compared to VMs, but they require an underlying OS and kernel that provide the basic services. Their instantiation time could be up to few seconds and they are mostly used to host lightweight applications.

### Unikernels

Unikernels can be considered as shrunken VMs, as they contain the minimum required OS services and dependencies for a single task to run. They do not require a kernel space, as the kernel commands are executed in the user space and they do not need and underlying OS; they can be executed directly at the hypervisor or on bare metal (Figure 1-c). Since they contain only the essential information for the application to run, the size of the unikernels is very small, while the reduced kernel complexity makes the unikernels to run faster than the VMs. Thus, they can be instantiated almost instantly. In contrast, unikernels can only run a single process and the development of applications able to run on this environment is trickier. Finally, containers are prone to hack attacks, since their kernel uses the user space.

## FOG-ENABLED C-V2X ARCHITECTURE

We propose the fog-enabled C-V2X architecture depicted in Figure 2. This architecture includes computation, network and storage resources on each fog node, with the core having the most, and the vehicle the least resources. The MEC resides in the edge fog node. The proposed architecture fully supports NFV by enabling the virtualization of physical resources at the hypervisors that are located at the three nodes, and by applications running as connected VNFs (i.e., AaaSFC). The virtualized infrastructure manager (VIM) is responsible for the management and control of the heterogeneous resources of the NFV infrastructure, while the NFVO performs the resource orchestration. They both reside in the core fog node.

In this architecture, the 5G RAN is considered for the communication of the three different fog nodes. The 5G core control functions reside in the core fog node and the user plane function (UPF) that steers the data traffic to desired applications and network functions resides in the edge fog node. The gNBs communicate among them using the NG interface, with the UPF using the N3 interface and with the vehicles using the Uu interface. The vehicle fog nodes can also communicate between them using the direct PC5 interface. The PC5 interface, as described in 3GPP Release 15, enables the direct communication where the infrastructure does not participate. Finally, each vehicle fog node acts as an IoT gateway, as it collects and aggregates the data from the sensors deployed at the vehicle.

In terms of virtual environments deployment, the VMs can run either at the core or edge fog nodes, the

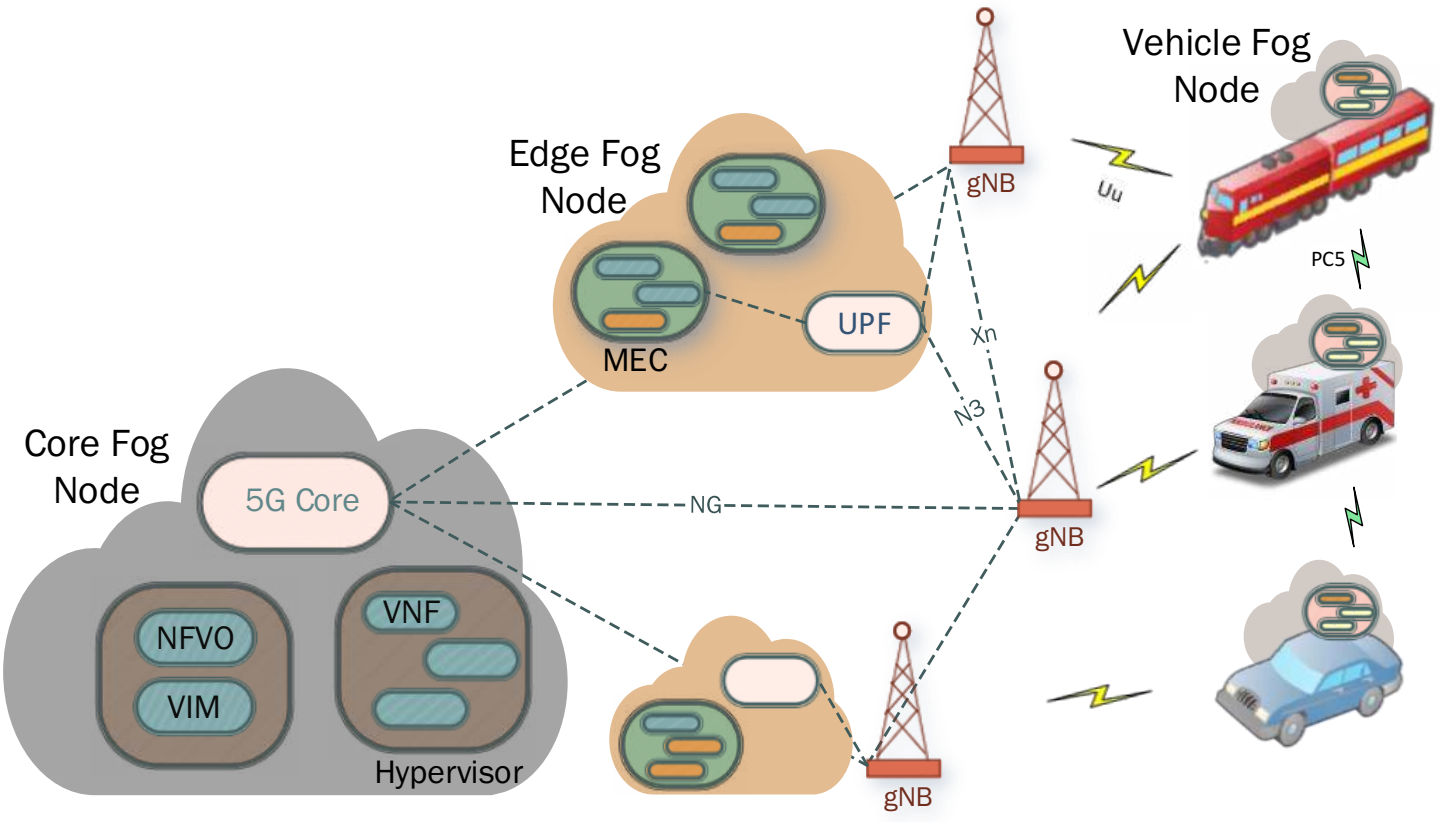



*Figure 2 – Fog-enabled C-V2X Architecture*

containers at the edge or vehicle fog nodes and the unikernels are able to be executed at the vehicle fog nodes only. Regarding their connectivity, the VMs of the core fog node can communicate with VMs of the core and edge fog nodes, and with containers of the edge and vehicle fog nodes. Furthermore, the containers of the edge nodes can also communicate with the containers of the vehicle nodes. Finally, containers of the vehicle fog nodes can communicate with the unikernels and with containers on other vehicle fog nodes, while the unikernels can only communicate with the containers and sensors of the vehicle fog nodes. All the communications are enabled by virtual links.

There are numerous applications that can be executed in this architecture. Crucial services, such as autonomous driving could be combined with non-crucial services, such as traffic jam avoidance or infotainment services, running in parallel. In some cases, the initial allocation of the VNF resources can prove to be insufficient. Therefore, LCM actions, such as live migration and scaling, need to be performed to ensure a smooth service operation.

**Lifecycle management**

Each individual VNF has a lifecycle, which is governed by the NFVO that can be considered as the central controller of the system, in terms of filtering the incoming requests and (re)allocating the physical resources. The NFVO executes periodic checks in order to monitor the current availability of resources and ensures that the NFV infrastructure adapts to data traffic variations. Moreover, the NFVO is responsible for two important actions, namely the live migration and the scaling:

- **Live migration**: the process that involves moving a VNF to a different hypervisor for resource optimization purposes, without service interruption [13]. This is achieved by running the instances of the old and new hypervisor simultaneously while service migration is performed, and only migrating RAM contents as a final step.

- **Scaling:** the ability of the VNFs to scale-out upon increased resource utilization and scale-in upon resource underutilization. In this architecture, the scaling option that is selected is the horizontal scaling (i.e., the instantiation of

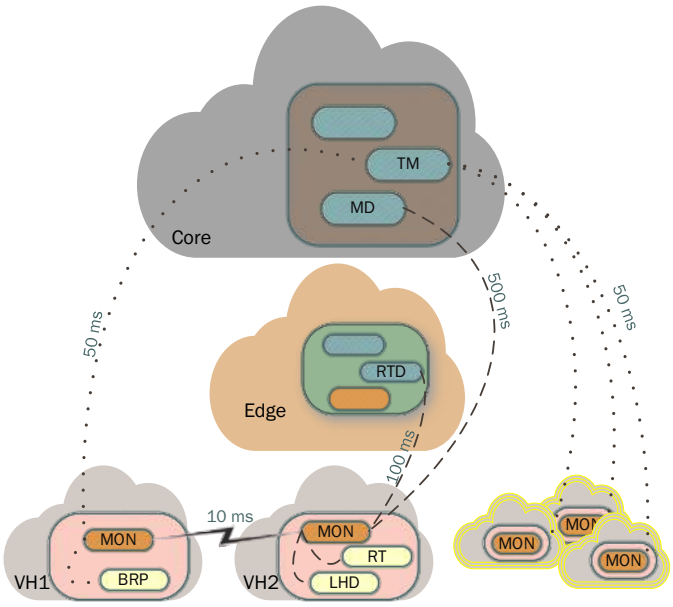



*Figure 3 - AaaSFC for co-operative drive safety (dotted line) and for autonomous driving (dashed line)*

more VNFs). Compared with the vertical scaling (i.e., the increase of the resources allocated to a VNF), the horizontal scaling provides the distributed applications with the required elasticity, redundancy and continuous availability.

## C-V2X APPLICATIONS AS A SERVICE FUNCTION CHAIN

In this section, we provide two examples of how C-V2X use cases can be developed as AaaSFCs, as well as their LCM.

### Co-operative road safety

Let us consider the co-operative road safety example, combined with a vehicle warning feature that were described earlier. If we want to deploy it as a simplified AaaSFC (Figure 3 – dotted line), it involves:

- One VM module that runs at the core and hosts the traffic manager module (TM).
- One container module that runs at the vehicle, monitors the data of the unikernels and notifies possible involved parties (MON).
- One unikernel module that runs at the vehicle where the data of the breaking and proximity sensors can be collected (BRP).

In case a hard break occurs (BRP) on vehicle 1 (VH1), the MON checks for any possible vehicles in proximity (VH2) and notifies them for the hard break (through the direct interface) in order for the latter to decide an emergency break action. At the same time, the MON sends the notification to the TM, through the 5G network, in order for the latter to issue and broadcast a warning notification to the vehicles that follow VH1, but are out of range for direct V2V communication.

### Autonomous Driving

Let us consider the autonomous driving application as part of many applications that cooperate. In a simplified version, we could involve two of the previously described applications, the HD sensor data and the navigation and traffic jam avoidance. If we want to deploy it as an AaaSFC (Figure 3–dashed line), it involves:

- One VM module that runs at the core and hosts the latest map data of a geographical area (MD).
- One VM module that runs at the edge and monitors the real-time vehicle traffic data of an urban area (RTD).
- One container module that runs at the vehicle, monitors the data of the unikernels and notifies possible involved parties (MON).
- Two unikernel modules that run at the vehicle where the data of the lidar and the HD cameras are collected (LHD) and the route is determined (RT).

In case there is a request for an emergency service, the MON requests from the RTD the best route, based on real-time vehicle traffic data, and updates the RT with the route. When the vehicle arrives to an intersection, the data from the LHD will be used by the MON to notify all nearby vehicles using the direct interface.

While on route, a traffic jam is detected. The RTD forwards an alternative route to the MON but the RT map data are not updated; thus, the MON requests the additional map from the MD and forwards it to the RT. In the last part, the edge acts as the intermediate in the vehicle-to-core communication.

### AaaSFC lifecycle management

Each VNF has specific virtual resources allocated to it and each virtual link that interconnects the VNFs has a maximum latency that can tolerate, based on

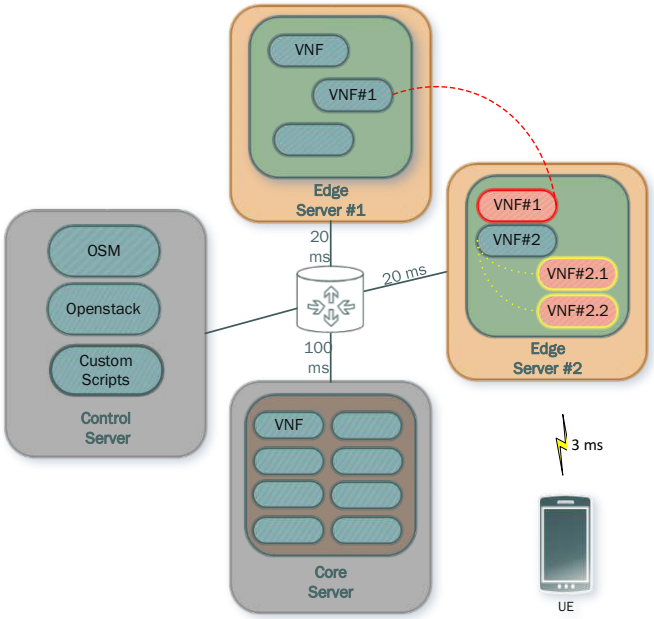



*Figure 4 - 5G experimental platform implementation*

| | Control server | Core server | Edge server#1 | Edge server#2 |
|---|---|---|---|---|
| **CPU** | Intel i5-8500 | Intel i5-8500 | Intel i5-7400 | Intel i5-7400 |
| **Cores** | 6 | 6 | 4 | 4 |
| **RAM** | 32 GB | 32 GB | 16 GB | 16 GB |
| **Storage (SSD)** | 250 GB | 2x 250 GB | 120 GB | 120 GB |
| **NIC** | 2x 1Gbps Ethernet | 2x 1Gbps Ethernet | 2x 1Gbps Ethernet | 2x 1Gbps Ethernet 1x 802.11n |

*Table 1 - Hardware specification*

the application. Depending on real-time data traffic conditions, the application's SLAs might be violated. In what follows, we explain the conditions under which a migration or a scaling decision might occur, in order to prevent such violation.

**Core to edge migration decision:** The example of the co-operative road safety uses the TM/VM of the core. In case of high data traffic (e.g., during rush hours), the required for the service latency could be violated. Thus, we propose a migration action from the core to the edge, in order for the TM/VM to be closer to the vehicles. In the case of insufficient edge resources, a migration from the edge to the core of non-crucial VNFs could take place in order to free up the needed edge resources for the TM/VM accommodation. When the data traffic is restored to normal, the VNFs are migrated back to their original hosts.

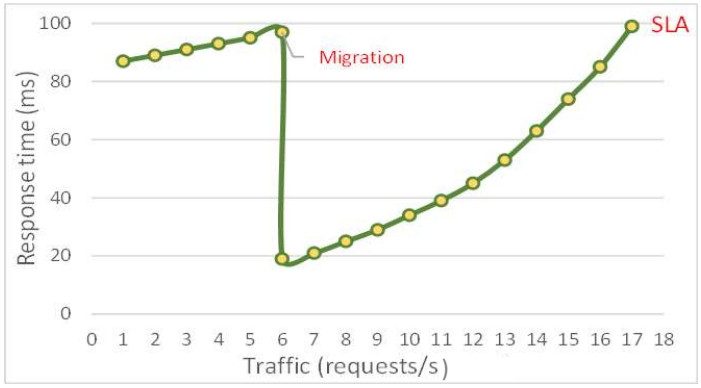
**Edge to edge migration decision:** Another action that could result in migration is due to the vehicles' mobility. Since the vehicles move from one gNB to another, if the two gNBs are not served by the same edge fog node, then a migration of the service is needed from the old to the new edge fog node, following the user's handover process between the gNBs. Since the edge fog nodes communicate between them, the VM could have remained at the old node. However, this would result in increased latency, since more hops are added between the vehicle-to-edge communication.

**Scaling decision:** In case of high data traffic, one or more of the VNFs that are part of the AaaSFC can reach their maximum utilization capacity. In this case, the NFVO will decide a scale-out operation by creating a new copy of the overutilized VNF, balancing the load between the two VNF instances. This process can be repeated (as long as there are sufficient physical resources) until the data traffic can be handled properly. In the case of scaling, the total VNFs of an AaaSFC are consequently increased. The NFVO is also responsible for the reverse process, that is, the scale-in, when the data traffic is restored to normal.

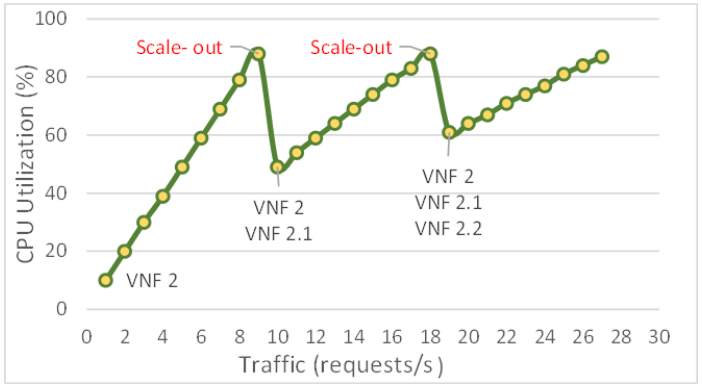
## 5G EXPERIMENTAL PLATFORM IMPLEMENTATION

In order to demonstrate a proof-of-concept of our architecture, we developed a 5G experimental platform (Figure 4). The hardware of the platform (Table 1) consists of one control server for the management and orchestration of the physical and virtual infrastructure, one core and two edge servers for the core and edge processing needs respectively. In terms of compute resources, the physical servers at the edge site have lower computational power compared to the server at the core. In terms of networking, all physical servers have two network interface cards (NIC) and are connected to two routers through 1 Gbps Ethernet interfaces, while edge server#2 has an additional 802.11n interface where a smartphone that acts as the UE can be connected.

With respect to the software installation, Openstack [14], on its Queens release, is the open-source infrastructure-as-a-service platform that is employed as the VIM, in order to deploy and control the VNFs, while Open Source MANO (OSM) [15], on its sixth




a)




b)

*Figure 5 - Demonstration of a) live migration and b) scaling functionalities*

release, is the software that is used to enable the VNF management and orchestration. Finally, neither Openstack nor OSM are aware of the application that runs on the VNF or its SLAs. Therefore, we have deployed bash-based custom scripts that monitor the inter-VNF latency and trigger the migration decisions, based on custom-defined thresholds.

## EXPERIMENTAL RESULTS

In order to validate the described architecture, we conducted a set of experiments, leveraging the 5G platform.

### Experimental setup

In our setup (Figure 4), we assume service#1 and service#2 that are hosted on the VNF#1 at the edge server#1 and on the VNF#2 at the edge server#2 respectively. Service#1 has an SLA that can tolerate up to 100 ms of latency, while VNF#1 needs 5 percent of CPU resources in order to process each incoming request. Service#2 has an SLA that can tolerate up to 15 ms of latency and VNF#2 needs 9.8 percent of CPU resources in order to process each incoming request. Furthermore, if the CPU utilization of VNF#2 exceeds 90 percent of the CPU resources, the process time for each request becomes unstable and the latency SLA for service#2 is violated. Thus, the scale-out threshold is set at 88 percent of CPU utilization. In order to equally distribute the data traffic among the VNFs, we have already deployed a load balancer with round robin policy. Finally, we can define the number of requests/s that the UE, which serves as the vehicle, can send. For the live migration experiment, we start with 1 request/s and gradually increase to 17 requests/s. For the scaling experiment we start with 1 request/s and gradually increase to 27 requests/s. The duration that each experiment run was 12 h.

### Live migration experiment

The first experiment demonstrates the case of a vehicle leaving a gNB's premises and entering another's, and the two gNBs are not served by the same edge fog node. Figure 5-a depicts the response time versus the requests/s that the vehicle produces. Up to the first 6 requests/s, the VNF that serves this vehicle runs at the old edge fog node (i.e., edge server#1) and the service response time is 99 ms. If the vehicle makes more requests/s, the service#1 latency SLA will be violated. Thus, the solution is to migrate the VNF#1 to the nearest to the vehicle edge fog node (i.e., edge server#2). This reduces the overall response time, since the delay that was added by the edge-to-edge server communication is now eliminated. This way, we can support more requests, without violating the latency SLA, compared with an environment that does not support migration features.

### Scaling experiment

The second experiment shows the case of increased demand for a specific service (i.e., during rush hours) but the VNF's already allocated resources cannot cope with such high demand. Figure 5-b displays the average CPU utilization of the active VNFs versus the requests/s that the UE sends. Up to the first 9 requests/s, the process time at VNF#2 is stable, with a CPU utilization of 88.2 percent. Since the scale-out threshold has been exceeded, VNF#2.1 is instantiated. Hence, the load balancer equally distributes the data traffic and each VNF reaches approximately 45 percent CPU utilization. As the requests/s increase, the scale-out process is executed once more when the average CPU utilization of the two VNFs exceeds again the 88 percent threshold.

This results in the instantiation of VNF#2.2. Once the data traffic is reduced, the scale-in process will destroy the VNFs that are underutilized. In this way, the resources can be efficiently managed and the application can serve more users when needed, compared with monolithic architectures that do not support scaling functionalities.

## CONCLUSION

In this article we proposed a fog-enabled C-V2X architecture, able to exploit the interplay among the core, the edge and the vehicle layers. We presented specific examples of C-V2X use cases that can be deployed as AaaSFC and provided information on their LCM. Additionally, we deployed an experimental platform, where open-source software, along with custom-made scripts, were used to demonstrate experiments that take advantage of the proposed architecture and validate the usefulness of the live migration and scaling features, resulting in respecting the latency SLAs and maximizing the served users. As future work, we aim to provide online algorithms that will efficiently manage the lifecycle of the AaaSFC that are deployed across our proposed architecture.

## AKNOWLEDGMENT

This work has been supported in part by the research projects SPOTLIGHT (722788), AGAUR (2017-SGR-891), 5G-DIVE (859881), SPOT5G (TEC2017-87456-P), MonB5G (871780) and 5G-Routes (951867).

## REFERENCES

[1] "Ericsson Mobility Report", June 2019, Available: https://www.ericsson.com/en/mobility-report/reports/june-2019, accessed May 30, 2020.

[2] "Release 14 Description", 3GPP TR 21.914, Tech. Rep., 2018.

[3] National Higway Traffic Safety Administraion, Available: https://www.nhtsa.gov/speeches-presentations/traffic-safety-and-59-ghz-spectrum, accessed May 30, 2020.

[4] 5G Automotive Association (5GAA) , "C-V2X Use Cases: Methodology, Examples and Service Level Requirements", *White Paper*, 2019, pp. 1-77.

[5] C. Bockelmann et al., "Towards Massive Connectivity Support for Scalable mMTC Communications in 5G Networks," *IEEE Access*, vol. 6, 2018, pp. 28969-28992.

[6] F. Giust et al, "MEC Deployments in 4G and Evolution Towards 5G", *ETSI White Paper*, 24, 2018, pp. 1-24.

[7] I. Sarrigiannis et al., "Online VNF Lifecycle Management in a MEC-enabled 5G IoT Architecture", *IEEE Internet of Things Journal*, vol. 7, no. 5, 2020, pp. 4183-4194.

[8] Consortium Architecture Working Group, "OpenFog reference architecture for fog computing.", Feb. 2017.

[9] H.-C. Hsieh, C.-S. Lee, and J.-L. Chen, "Mobile edge computing platform with container-based virtualization technology for iot applications," *Wireless Personal Communications*, vol. 102, no. 1, 2018, pp. 527–542.

[10] A. Moubayed et al., "Edge-enabled V2X Service Placement for Intelligent Transportation Systems", *IEEE Transactions on Mobile Computing,* 2020, pending publication.

[11] F.-E. da Silva Barbosa, F.-F. de Mendonça Júnior, K.L. Dias, A platform for cloudification of network and applications in the Internet of Vehicles. *Trans Emerging Tel Tech*, 2020, pending publication.

[12] "Intelligent transport systems (ITS); vehicular communications; basic set of applications; definitions," ETSI TR 102 638, Tech. Rep., 2009.

[13] A. Laghrissi and T. Taleb, "A survey on the placement of virtual resources and virtual network functions", *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, 2019, pp. 1409–1434.

[14] The Openstack foundation, https://openstack.org/, accessed May 30, 2020.

[15] Open Source Mano, https://osm.etsi.org/, accessed May 30, 2020.

## BIOGRAPHIES

Ioannis Sarrigiannis [S'10] (isarrigiannis@iquadrat.com) received his five-year diploma (MSc equivalent) in Information and Communication Systems Engineering in 2015 from the University of the Aegean, Samos, Greece. Currently, he is a Marie Curie Researcher at Iquadrat Informatica S.L., in Barcelona, Spain, and is pursuing his Ph.D. degree in Signal Theory and Communications (TSC) with the Polytechnic University of Catalonia (UPC), Barcelona, Spain. His research interests include Software Defined Networks, Network Function Virtualization, VNF Placement and Orchestration and Network Slicing, in the scope of cloud-edge architectures towards the realization of 5G concepts.

Luis M. Contreras (luismiguel.contrerasmurillo@telefonica.com) is a Telecom Engineer by the Universidad Politécnica of Madrid (1997) and M. Sc. on Telematics by the Universidad Carlos III of Madrid (2010). Since August 2011 he is part of Telefónica I+D / Global CTIO. Previously he worked for Orange Spain and Alcatel. His research interests are on 5G and scalable networks.

Kostas Ramantas (kramantas@iquadrat.com) has received his Diploma of Computer Engineering, his MSc degree in Computer Science and Engineering and his PhD degree from the University of Patras, Greece, in 2006, 2008 and 2012 respectively. He has been the recipient of two national scholarships and has participated in the EC funded ICT-BONE and ePhoton/One+ Networks of Excellence, conducting joint research with many European research groups. His research interests include modeling and simulation of network protocols, and scheduling algorithms for QoS provisioning. In June 2013, he joined IQUADRAT as a senior researcher and is actively involved in EU-funded research projects.

Angelos Antonopoulos [SM'15] (aantonopoulos@cttc.es) is a Senior Researcher at CTTC/CERCA. He has authored over 100 publications on various topics, including 5G wireless communications, content

caching and dissemination, energy efficient network planning and network economics. He currently serves as an Editor for IEEE Access, IEEE Networking Letters and Elsevier Computer Networks. He has received the Best Paper Award at IEEE GLOBECOM 2014, the Best Demo Award at IEEE CAMAD 2014, the first prize in the IEEE ComSoc Student Competition (as a Mentor), and the EURACON Best Student Paper Award at EuCNC 2016.

Christos Verikoukis [SM'07] (cveri@cttc.es) received his Ph.D. degree from UPC, Barcelona, Spain, in 2000. He is currently a Fellow Researcher at CTTC, Castelldefels, Spain, and an Adjunct Professor with UB. He has authored more than 134 journal papers and more than 200 conference papers. He has coauthored more than three books, 14 chapters, and two patents. He has participated in more than 40 competitive projects and has served as the Principal Investigator of national projects. He has supervised 15 Ph.D. students and five postdoctoral researchers.
Dr. Verikoukis was the recipient of the Best Paper Award at IEEE International Conference on Communications 2011 and 2020, IEEE GLOBECOM 2014 and 2015, and EUCNC/EURACON2016, and the EURASIP 2013 Best Paper Award for the Journal on Advances in Signal Processing. He is currently Associate EiC for the IEEE NETWORKING LETTERS, IEEE ComSoc EMEA Director and Member-at-Large of GITC.