Journal of Computer Engineering
ISSN 10003428

Original Research Article
Vol 14, Issue 10, 2025 | ISSN: 1000-3428

Migration of Virtual Network Functions in a Multi-Access Edge Computing-Enabled 5G Network Architecture

Emilia Rodriguez, Ioannis Karampelas

Department of Electrical and Computer Engineering, University of California, Los Angeles (UCLA); Networking Research Group, University of Amsterdam, the Netherlands

***Abstract*—The upcoming fifth generation (5G) of mobile communications brings new paradigms, such as network slicing and automation, to the forefront. Cloud computing and network function virtualization (NFV) constitute two fundamental key enablers towards the implementation of these new paradigms. On the one hand, computational functionalities are no longer limited to distant servers, but they are coming closer to the end user through Multi-access Edge Computing (MEC) technology, thus creating a multi-tier cloud architecture in modern networks. On the other hand, going beyond the existing rigid architectures, NFV enables the flexible and on-the-fly creation and placement both of application and network functions, aiming at satisfying the diverse application requirements and optimizing the management of the heterogeneous (network, computational and storage) resources. In this paper, we discuss the challenges that arise in a MEC-enabled 5G architecture that supports the flexible placement/migration of network and application virtual network functions (VNFs), orchestrated by an NFV Orchestrator (NFVO) with admission management functionalities, able to manage on-the-fly the network functions and resources. Finally, we describe in detail a MECenabled testbed architecture that facilitates the development and testing of such solution in the context of 5G networks.**

Network Slicing

I. Introduction

The emerging fifth generation (5G) of wireless communications brings new services and applications that can be generally classified into three categories, namely enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC). Despite the fact that the user equipment (UE) devices become more powerful and sophisticated, 5G services cannot be executed solely in isolated devices. The constantly increasing need for vast, uninterrupted and agile information exchange among the peers and the central locations requires the appropriate infrastructure upgrade and the reshaping of the traditional Long Term Evolution (LTE) network architecture, in order to overcome the current limitations, e.g., fronthaul and backhaul capacity and delay or radio interface conditions.

Towards overcoming the aforementioned obstacles and, in order to serve the increased requirements of the future 5G services, a Multi-access Edge Computing (MEC)-enabled architecture that enables cloud computing capabilities at the edge of the access network is widely accepted by the literature [1]. Key enabler elements of such architecture are the virtualization of the applications and network functions, as well as the on-demand decentralization of the computing and network resources. The restrictions applied by the legacy LTE networks, e.g., the fixed placement of the network functionalities, are eliminated with the aid of Network Function Virtualization (NFV) technologies [2]. Hence, application and network functionalities are handled as Virtual Network Functions (VNFs) and managed by a NFV Orchestrator (NFVO) [3], able to be executed within the various locations of a distributed system and, in our case, in the UE proximity. Furthermore, the concept of network slicing [4] ensures that the isolation among the services is guaranteed.

Within this new paradigm, the on-demand allocation of network resources in the form of VNFs becomes feasible, a

concept similar to the novel distributed Evolved Packet Core (EPC) architecture [5]. Having a closer look to the literature solutions for the MEC implementation, a small cell cloud enabled architecture in the context of LTE for mobile offloading is proposed in [6], a Software Defined Network (SDN)/NFV enabled architecture integrating the cloud services into the mobile

How to cite: Emilia Rodriguez, Ioannis Karampelas (2025). Migration of Virtual Network Functions in a Multi-Access Edge Computing-Enabled 5G Network Architecture. Journal of Computer Engineering, 14(10), 107–114. DOI: https://doi.org/10.5281/zenodo.19358991

networks is described in [7], while the concept of replacing the centralized core network by a distributed one, "following the users on the move" is presented in [8]. In all the aforementioned work proposals though, the resources are completely distributed, without exploring the potential benefits stemming from the joint exploitation of centralized and edge resources. In terms of implementation, a 5G-aware evaluation testbed with MEC capabilities has been described in detail in [9], with no reference, however, to the softwarization of the network functionalities. To the best of our knowledge, there is no related work that combines the interplay of the edge with the core, in a virtualized manner, exploiting VNF migration capabilities for both meeting the computational and network needs of the UEs and respecting the restrictions of the network, accompanied by an open-source software deployed testbed.

In order to meet the stringent 5G requirements and especially the latency constrains, including, but not limited to, the URLLC paradigm, we present a MEC-enabled 5G architecture, distributing the computational and network resources in the UE proximity. This self-aware network of services is being supported by both core and edge computational capabilities, allowing the smart interplay between the two locations, respecting the isolation that network slicing requires. Furthermore, we discuss the potential of this NFV-based design with regard to the efficient and joint orchestration of the MEC and cloud resources by an NFVO. As an example, we present a policy for (re)allocation and dynamic migration of application and network VNFs, aiming to maximize the utilization of the MEC resources for latency-driven applications. Finally, we enhance the dynamic of the aforementioned architecture by outlining an NFV-enabled testbed implementation, deployed with open-source software over generic purpose hardware, and able to provide tangible results on the demanding 5G scenarios.

II. MEC Architecture

We consider a MEC-enabled 5G architecture depicted in Figure 1. A heterogeneous Radio Access Network is considered, including standalone evolved NodeBs (eNBs), small cells and a Cloud Radio Access Network (C-RAN) deployment, where Base Band Units (BBUs) are connected with Remote Radio Head (RRH) units that serve the UEs. We consider heterogeneous backhaul and fronthaul connections with either fiber of mmWave links, whereas the LTE air interface is employed for the UE access connectivity.

The considered architecture includes two tiers of computational capabilities: the core tier cloud and the edge tier cloud. On the one hand, the core tier cloud can be considered to have practically infinite computing resources (i.e., through on-demand upscaling) and constitutes a shared multi-tenant resource, which can host both application VNFs (residing in the Application Cloud) and network VNFs. On the other hand, the edge tier clouds contain limited computing resources, which are allocated to the MEC entity that hosts application VNFs and to network VNFs that should be placed closer to the UE-side to satisfy specific service requirements. We also consider that the Edge Clouds are connected to the core tier, while Edge Cloud interconnectivity may also be possible in some cases.

Fully adopting the NFV paradigm, we consider that the 5G applications are composed by sets of VNFs with different latency constraints. In particular, we distinguish between latency-crucial VNFs, which should be executed with the minimum possible latency, and latency-tolerant VNFs that can tolerate a higher degree of delay. Accordingly, the applications can be classified intro three categories: i) Real-Time Applications (RTAs), consisting only of latency-crucial VNFs, ii) Non Real-Time Applications (NRTAs), consisting only of latency-tolerant VNFs, and iii) Hybrid Applications (HAs), containing both types of VNFs. An example of an RTA could be the transportation safety for autonomous driving vehicles [10], where decisions need to be taken within a few ms, in order for the system to be responsive and safe. An online storage service can be considered as an NRTA, since access to the storage can tolerate some latency. Finally, an HA example could be a smart surveillance system, where the image recognition functionalities for the detection of suspicious movements should be executed with minimum delay (RTA part), whereas storage functionalities or more computational-

heavy functions (e.g., face recognition) could be offloaded to the core (NRTA part). In general, the MEC entity can host and execute all three types of applications. However, due to the limited MEC resources, some of the latency-tolerant VNFs can be migrated to the Application Cloud at the core tier, or to another MEC entity. Specifically, RTAs can only be executed at the closest MEC, due to the very stringent latency constraints. NRTAs have loose latency requirements, thus they can be migrated to the Application Cloud, or to another MEC, when MEC resources become depleted and offloading is needed. Finally, HAs can be partly offloaded by moving the non-critical VNFs to another MEC or to the Application Cloud.

With respect to the network functions, traditionally, the core network functions of LTE networks, known as EPC, have been deployed in a static and centralized manner at a central location at the core tier. However, the recent NFV advances have permitted the virtualization of the EPC, by transforming the rigid core architecture into a set of network VNFs that can be flexibly placed at different parts of the network [11]. Following this trend, in the proposed architecture, the EPC VNFs can be deployed either on the core or on the edge side, depending on application-specific requirements.

In terms of implementation, each VNF is hosted by a

Virtual Machine (VM) located either at the Edge Cloud or at the Core Cloud, thus creating an isolated and stable environment for the VNFs to run. This approach enables the seamless migration of VNFs between the Edge and/or the Core Clouds in real-time and without any service interruption, exploiting the advances in VM migration, which have been gaining a lot of ground in the context of Cloud Computing. Finally, all network and computing resources are centrally managed by an NFVO entity that resides at the core tier. The NFVO handles all incoming application requests, by allocating the most suitable resources and orchestrating the migration of application or network VNFs whenever necessary.

III. NFV Orchestrator

The NFVO resides in the core tier and can be considered as the central controller of the system, in terms of filtering the incoming requests and (re)allocating the computational and network resources. It executes periodical checks in order to monitor the current availability of computing and network resources. More specifically, the role of the NFVO consists in:

- Accepting or rejecting the incoming requests that have predefined requirements in network resources (e.g., latency and throughput), computing resources (e.g., CPU, RAM and storage) and execution duration (lifecycle).
- Defining the location (i.e., core or edge) where the accepted request should be executed.
- Executing the application and network VNF migration.

Regarding the computing resources allocation, the priority is always given to the edge side. In particular, in case of an RTA/NRTA, a new VM will be created on the MEC server, allocating the required resources, while, in case of an HA, 2 new VMs will be created on the MEC side, hosting the crucial and the non-crucial latency parts, respectively. In case there are no available resources at the MEC side, NRTAs and the latency-tolerant parts of the HAs may be executed in the Application Cloud. However, this does not apply for RTA requests, which should be rejected, as their latency requirement cannot be met if the RTA is executed in the Application Cloud or a neighboring MEC.

Apparently, it is highly likely that the MEC's limits will be reached if its resources are constantly allocated to the VNFs. In order to prevent the requests from being rejected on the MEC side, the NFVO handles two types of migration: the application and the network VNF migration. On the one hand, the former is used to migrate the VMs when the computing resources are depleting on the MEC side, enabling the better utilization of the computing resources of the system and preventing resource depletion. On the other hand, the latter is used when there is the need to bring the network functionalities closer to the user by reshaping the system's network components, in order to meet the increased network-related requirements that 5G applications impose, especially in high traffic periods.

A. Application VNF migration

Each application request has a predefined lifecycle, i.e., its execution duration is defined upon receiving and accepting the request and the resources for the VM are allocated for this specific duration. When there is a need for releasing MEC resources, the VMs that are running NRTAs are the first ones that are being migrated, either to another MEC with low computing resources utilization, or to the Application Cloud. Priority is given to the applications with longer lifecycle and lower computational demands, in order for the migration to be fast. Then, following the same criteria (i.e., lifecycle and computational demands), the latencytolerant parts of the HAs are migrated to the Core Cloud, while RTAs and the latency-crucial parts of the HAs remain at the

MEC side. Subsequently, in the case that the resource utilization on the MEC server is still high, despite the aforementioned migrations, the NFVO also decides the migration of the VMs with lower lifecycle. However, during this process, the NFVO examines whether the duration for the migration is shorter or equal to the remaining duration of the VM's lifecycle, as there is no point in migrating a VM whose lifecycle is about to expire.

Following the aforementioned technique, we manage to efficiently allocate the computational resources of the MECenabled architecture, achieving offloading of the MECs to the Application Cloud (Figure 1), or to another MEC, without compromising the latency factor nor wasting the edge computational resources. It is also worth noting that the whole operation takes place in an isolated manner, without affecting the execution of the other VMs. This isolation is essential in recent paradigms (e.g., network slicing), where network operators need to guarantee the resources for the different services.

B. Network VNF Migration

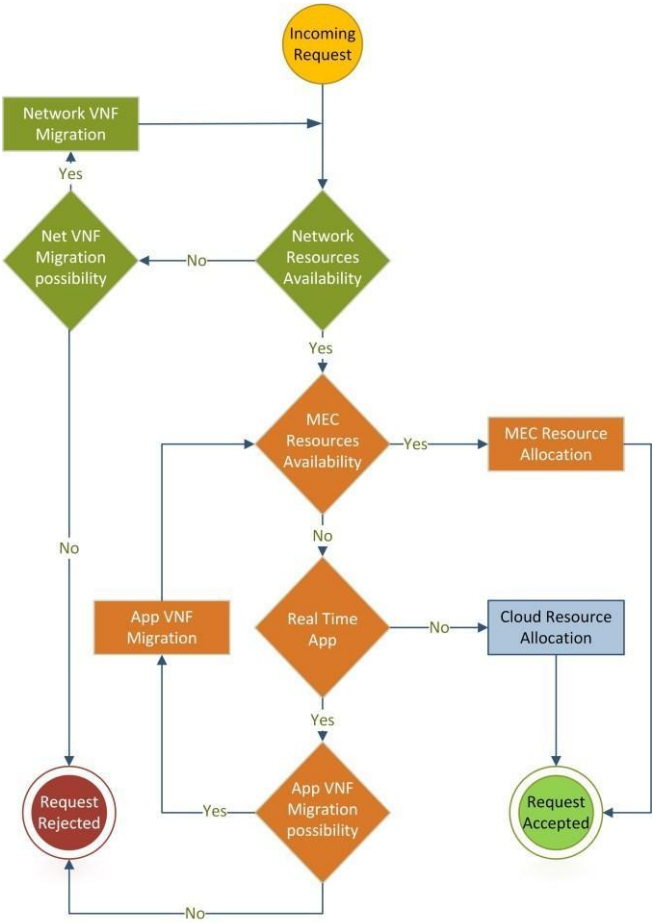
Regarding the network operation, in case of increased latency-crucial requests, the responsible VNFs for making the system susceptible to delays can be duplicated from the core to the edge. For example, we can copy some or all the EPC functions (Figure 1), increasing the edge's capabilities in order to meet the temporarily increased local requirements. The EPC that runs on the core tier continues to serve the rest of the system, or even specific latency-tolerant requests from the MEC that has the migrated "local" EPC, in accordance with the network slicing paradigm. Upon reduced traffic, the "local" EPC functionalities may be released from the edge tier.

C. Allocation and migration example

The supported reshaping of the architecture components (i.e., VNFs migration) could be initiated in three different ways:

- ***Proactively***, using for example traffic prediction algorithms,
- ***Retrospectively***, based for example on a rejection percentage, meaning that in case a threshold of rejected requests is reached (or redirections to the Cloud or to another MEC), a reshaping will be performed, based on whether the rejection was a result of limited computing resources or limited network resources (or both),
- ***On-the-fly***, a dynamic method based on the current incoming traffic and system's utilization.

Figure 2 describes an on-the-fly toy example of a dynamic resource management and VNF (re)allocation method. Upon an incoming request, the network resources availability is



primarily checked. If there are not sufficient resources, the NFVO checks whether some or all of the core network functions can be migrated on the edge side. If this is not possible, the request is rejected. If sufficient networking resources are ensured, the request passes successfully the network check and proceeds to the MEC resources availability check. In the case where there are sufficient resources on the MEC side, the request is accepted and the required resources are allocated. However, if the resources are not sufficient, the proposed policy differentiates between RTA and NRTA incoming requests. In the first case, for the allocation of an RTA, we investigate the possibility of migrating other application VNFs from the MEC server to the Application Cloud, in order to release MEC resources and be able to accommodate the new RTA request. In the second case, the NRTA is executed directly to the Application Cloud, taking into account that the latency tolerant nature of such applications.

IV. Testbed Architecture

In order to demonstrate the potential of the described architecture, we introduce a real implementation of a MECenabled LTE testbed, described in Figure 3. The

hardware of the testbed consists of two physical servers where the functionalities of the core tier, i.e., the Application Cloud, the EPC and the NFVO, and the edge tier, i.e., the MEC and the eNB, are deployed, as well as another physical server that enables the management, in terms of infrastructure virtualization, of the other two. In terms of computing resources, the physical server at the edge site has

Figure 2 – On-the-fly allocation and migration flow chart

significantly lower computational power compared to the core server. Furthermore, we employ LimeSDR, a Software Defined Radio (SDR) platform that acts as the LTE Antenna, while the UE consists of a simple Raspberry Pi[1] and an LTE dongle.

In terms of networking, the three physical servers are connected to a router through 1 Gbps Ethernet interfaces, while the edge server is connected to the LimeSDR with a USB 3.0 cable. In the radio access part, ideally, there should be a wireless connection between the eNB antenna and the UE. However, given that transmissions over the licensed LTE frequency bands are not allowed, we employ a coaxial cable with an attenuator, in order to emulate the transmission losses, as well as a duplexer, in order to isolate the transmission (Tx) from the reception (Rx) channel.

With respect to the software installation, Openstack[2], the open-source Infrastructure-as-a-Service (IaaS) platform, is employed to provide and control the VMs that will host the application and network VNFs. The Openstack Controller Node, deployed on one physical server as shown in Figure 3, hosts the compute and network management components for the infrastructure orchestration, while the Compute Nodes, deployed on the other two physical servers, provide a pool of physical resources, where the VMs are executed. Openstack is based on services, each of which has a specific role to play. For instance, the Nova service, part of the Openstack
Compute Services, that resides on both Compute Nodes is

responsible for spawning, scheduling and decommissioning the VMs on demand, while the Neutron service, which resides on all three nodes, is responsible for enabling the networking connectivity. All nodes need two network interfaces, namely, the management, for the communication among Openstack services, and the provider network for the communication among the VMs. The Openstack services are deployed on LXD[3] powered containers in order to ensure their isolation from the VMs that run on the same physical server. Finally, it is worth noting that Openstack supports two important features, namely the horizontal scaling, i.e., the expansion of the physical resources, simply by adding new physical servers where the Compute Node services are deployed, and the live migration of the VMs. The migration is classified as live due to the fact that after a VM migration is complete, the VM status resumes exactly from the same state it was before the migration.

After successfully deploying the Openstack services, the system is ready to host the VNFs. We use OpenAirInterface (OAI) [4], an opensource SDR-based software/hardware platform that provides the LTE protocol stack as services that can run on simple PCs,

1

2

[3] LXD Container, https://linuxcontainers.org/lxd/

[4] OpenAirInterface, http://www.openairinterface.org/

thus enabling the implementation of the LTE EPC, the LTE eNB and the UE. OAI gives the ability to deploy the network VFNs, i.e., the eNB and EPC functions, either on the same or on separate environments. Hence, we deploy the eNB functions at the edge, while the EPC functions are deployed at the core side, with the ability though for the latter to move between the core and the edge, on demand. The UE software, that initiates the application requests, is executed on the Raspberry Pi.

The NVFO, which is responsible for the network resource allocation and management, is deployed as an

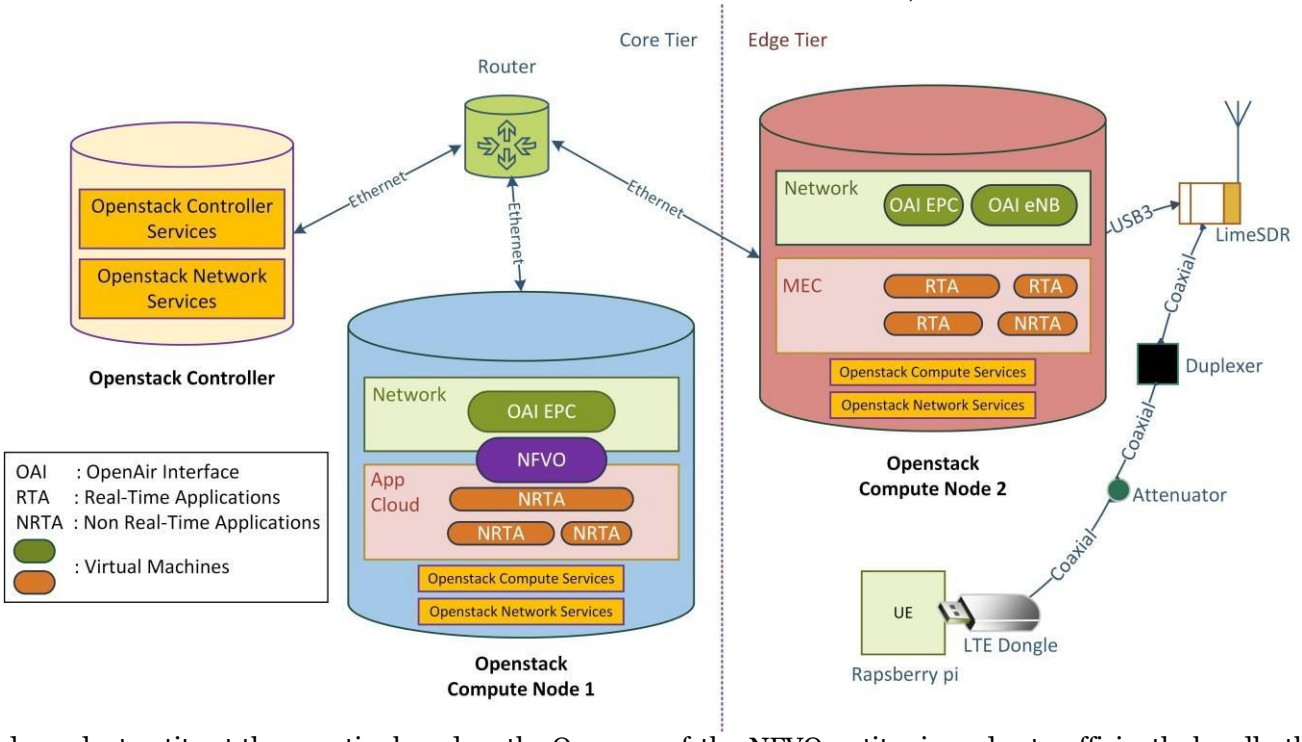


Figure 3 - Testbed Architecture

independent entity at the core tier based on the Open Source Mano (OSM)[5]. The NFVO executes all the necessary control functions to handle incoming application requests, by managing the admission control process and the allocation of the most suitable edge and core resources to the relevant VNFs. Furthermore, by controlling the Openstack's native live migration feature, the NFVO can reallocate the VNFs, when application or network VNFs rearrangement is needed. On the one hand, depending on the computing resources utilization level, the NFVO may take the decision to migrate an NRTA from the edge to the core, in order to release the needed resources for another possible incoming request that may be otherwise rejected. On the other hand, in case of increasing RTA requests, the NFVO may decide to move EPC functions from the core tier to the edge, in order to accelerate the system's response and reduce latency. This VNF migration is done without compromising the system's stability, as the VMs' isolation is guaranteed by Openstack's live migration feature, ensuring the uninterrupted functionality of the system.

V. Conclusion

In this paper, we presented a MEC-enabled 5G architecture able to exploit the recent advances in virtualization techniques and cloud computing, in order to flexibly and jointly manage the cloud and edge resources. Moreover, we described the desired features of the NFVO entity, in order to efficiently handle the allocation and reassignment (migration) of application and network VNFs across the core and edge tiers, based on the specific application requirements and available resources. Finally, we presented a MEC-enabled LTE testbed, based on open-source software and generic purpose hardware, able to verify the

[5] Open Source Mano, https://osm.etsi.org