Prototyping NFV-based Multi-access Edge Computing in 5G ready Networks with Open Baton

Giuseppe A. Carella, Michael Pauls, Thomas Magedanz
Technische Universität Berlin,
Berlin, Germany
email:{giuseppe.a.carella, michael.pauls, thomas.magedanz}
@tu-berlin.de

Marco Cilloni, Paolo Bellavista, Luca Foschini
University of Bologna,
Bologna,Italy
email:{luca.foschini, paolo.bellavista}
@unibo.it

***Abstract*—With the increasing acceptance of Network Function Virtualization (NFV) and Software Defined Networking (SDN) technologies, a radical transformation is currently occurring inside network providers infrastructures. The trend of Software-based networks foreseen with the 5th Generation of Mobile Network (5G) is drastically changing requirements in terms of how networks are deployed and managed. One of the major changes requires the transaction towards a distributed infrastructure, in which nodes are built with standard commodity hardware. This rapid deployment of datacenters is paving the way towards a different type of environment in which the computational resources are deployed up to the edge of the network, referred to as Multi-access Edge Computing (MEC) nodes. However, MEC nodes do not usually provide enough resources for executing standard virtualization technologies typically used in large datacenters. For this reason, software containerization represents a lightweight and viable virtualization alternative for such scenarios. This paper presents an architecture based on the Open Baton Management and Orchestration (MANO) framework combining different infrastructural technologies supporting the deployment of container-based network services even at the edge of the network.**

***Index Terms*—Network Function Virtualization; Software Defined Networks; Management and Orchestration; 5G; Containers; Multi Edge Computing; Multi-access Edge Computing**

I. Introduction

With the always increasing traffic demand, network providers are forced to upgrade their network infrastructures in order to cope with the novel set of use cases developed for the upcoming 5th Generation of Mobile Network (5G). However, with the current approaches, increase of capacity requires huge investments in vendors equipments, which in some cases does not have any return on investments. Network Function Virtualization (NFV) and Software Defined Network (SDN) are two innovative approaches that simplify the way network resources are deployed and controlled across distributed locations by also imposing a rethinking and redesign of the whole (traditional) network infrastructure. From the one hand, NFV supports decoupling through virtualization, network resources and services from the physical devices and appliances they run on. Network Functions (NFs) are transformed into Virtual Network Functions (VNFs), logical blocks that abstractly represent the several services and components provided by the infrastructure NFs. On the the other hand, SDN simplifies the control plane of network elements by extracting it from the physical devices, and providing abstracted Application Programming Interfaces (APIs) for on demand control of the network.

In order to adopt NFV and SDN technologies, network operators are forced to migrate towards standard de-facto cloud-based distributed infrastructures. VNFs can be moved into datacenters closer to the current users of the service, depending on the actual needs of the users of that particular services provided by those VNFs. This has been unfeasible so far, mainly due to the high costs of relocating staff and appliances across multiple locations. More recently, the rapid deployment of datacenters is paving the way towards a different type of environment in which the computational resources are deployed up to the edge of the network[1]. Along that direction, ETSI Multi-Access Edge Computing (MEC) envisions the idea of offering on-demand infrastructure resources to service providers and application developers at the edge of the network.

ETSI NFV and MEC are providing guidelines and architectures for supporting management and orchestration of applications on top of such distributed infrastructure. With the advent of the 5G-ready Networks, network services will be mainly implemented as software components executing on standard operating systems. Regardless of whether the final architecture is MEC or NFV, there is no doubt that basic compute, storage and networking resources need to be provided as-a-service to the Management and Orchestration (MANO) layer, orchestrating services across those virtualized infrastructures. Indeed, although the main objective is the same, virtualization technologies and standards employed in those domains are usually different. ETSI NFV is based on standard de-facto Virtualization Infrastructure Managers (VIM) like OpenStack, while MEC is targeting more lightweight technologies for basic infrastructure resources due to the limited computational capabilities usually available at the edge. Furthermore, also the way network services are configured and managed on top of heterogeneous infrastructure technologies differs. For the sake of simplicity, we will use the ETSI NFV terminology when referring to MEC scenarios. Therefore, the combination of multiple VNFs in a network service, as defined by ETSI NFV, corresponds to MEC applications executing on top of the MEC nodes.

Although there are gaps with regards to the information models for describing services in those different domains, our current approach focuses on providing an integrated orchestration solution, based on the Open Baton[2] MANO framework, combining the NFV and MEC use cases within a single orchestration environment. Open Baton is an open source project launched by the Technical University of Berlin and the Fraunhofer FOKUS institute, providing an extensible NFV MANO framework for orchestrating network services across multiple Point of Presences

[1]http://sdn.ieee.org/newsletter/march-2016/mec-enablement-by-means-of-an-open-source-etsi-mano-orchestrator

[2]http://openbaton.github.io/

978-1-5090-6008-5/17/$31.00

(PoPs). In this work, realized together with the University of Bologna, we significantly extended Open Baton along two significant directions: i) to support auto-registration of MEC nodes as available Point of Presence; ii) to support management and configuration of network services on top of the most common container orchestration platform, Docker. This paper is structured as follows. Section II provides some background information about the different scientific and industrial activities in the context of NFV, MEC and Containerization. Section III and Section IV present, respectively, our architecture proposal and prototype implementation. Section V shows some preliminary validation results and Section VI concludes the paper.

II. Background

This section provides the needed background material about all main technologies to enable the deployment of MEC-based services in the context of an NFV framework.

A. ETSI Network Function Virtualisation

Network Function Virtualization (NFV) is an initiative started in 2012 by the ETSI standardization organization. It aims to simplify the Management and Orchestration of complex network services by decoupling the software from the hardware on commodity infrastructures and proposes an architectural framework based on a set of well defined functional requirements [1]. ETSI announced the public availability of the initial set of specification documents, including the Management and Orchestration (MANO) domain [2], by the end of 2014 (phase 1). As commonly agreed and well described in this White Paper about perspectives on NFV priorities for 5G contributed by many big network operators around the globe [3], NFV is going to be one of the main enabler for 5G.

B. ETSI Multi-access Edge Computing

Mobile Edge Computing (MEC), recently renamed Multi-access Edge Computing (MEC), emerged in 2014 as the key enabler for offering "a service environment with ultra-low latency and high-bandwidth" [4]. Similarly to NFV, this new approach, driven by ETSI, assumes that mobile edge applications are running as software-only entities on top of a virtualization infrastructure. Going further along this direction, the idea of MEC is to host the applications on the network edge supposing that this location is potentially the closest location to the current location of the user who is consuming the service. According to this model, ETSI specified all main requirements of the MEC framework [5]. Moreover, the ETSI MEC specification group proposed a reference architecture that includes several functional blocks, such as, the mobile edge host, platform, orchestrator, platform manager and applications [6]. As NFV, MEC can also be seen as a key technology for enabling 5G as figured out in this White Paper [7].

C. Container-based Virtualization

The advent of software containerization has dramatically changed developers approach to how software is deployed and managed. A "container" (also known as a jail) is an isolated user space instance of an operating system. The isolation and versatility provided by containerization solutions like Docker, make them obvious choices when choosing which platforms are more suited to host and be deployment targets for Network Services and VNFs.

Some of the most relevant container implementations are FreeBSD Jails, Linux Containers (LXC), OpenVZ, rkt and Docker.

Container ecosystems have the potential to address some of the challenges that virtualisation poses when used together with NFV [8], such as the aforementioned performance and efficiency costs [9], the potential deployment slowdown issues caused by very large software images, and networking I/O overhead. As previously explained, applications in containers run on the host OS without any hardware indirection; thus, they can run more efficiently than their VM-based counterparts in many cases [10].

The aforementioned characteristics of containers make them a very appropriate choice for MEC: a MEC application can be represented by a container image, by nature easily deployable and scalable through its ability to be spawned several times in a really short time, with lower hardware requirements and complexity than those of a Virtual Machine.

D. Related Work

Since the first emergence of the term MEC, it has been gained a lot of attraction and became a very hot topic in the scientific area over the last year.

The work of Yu [11] gives first insights into the architectural overview of the MEC platform with respect to the key functionalities which has to be exposed from such a framework. Besides, this paper also surveys the state of the art by identifying main open research challenges of MEC.

A deeper view on use cases related to the Internet of Things and the 5G world, in particular the benefits and challenges, is provided in the work of Sabella et al. [12].

Another work from Wang et al. [13] focuses on the convergence of computing, caching and communication by introducing the key technologies that are enabling the concept of mobile edge networks.

Mach and Becvar [14] published a thorough survey on architecture and computation offloading including reference scenarios to point out how existing concepts which might be employed to integrate MEC functionalities into existing mobile networks.

Novel approaches towards a unified MEC framework has already been proposed by following the ETSI MEC architectures and principles in order to satisfy the set of MEC-related requirements. A general approach is the one proposed by Rimal et al. [15] empowering existing integrated fiber-wireless access networks in order to offer MEC capabilities. A double-tier MEC-NFV architecture is designed and proposed in the work of Sciancalepore et al. [16]. Let us note that this last approach is well-aligned with the ETSI NFV MANO specification that introduces a new management subsystem in order to provide application-oriented orchestration capabilities which are offered to the mobile edge applications and their management.

III. Architecture

This section describes the proposed high-level architecture of the MANO system able to orchestrate network services across NFV and MEC infrastructures. As already mentioned in the introduction, in order to orchestrate Software-based network services on top of different domains one of the major requirements is to be able to allocate infrastructural resources provided by heterogeneous technologies.

On the one hand, the NFV MANO specification defines several functional blocks, each one with a well-defined set of responsibilities and certain management and orchestration operations on well-defined entities, leveraging the services offered by the so-called NFV Infrastructure. On the other hand, the MEC architecture shows strong similarities with the NFV one in terms of functional components. Few extensions are needed in order to manage Mobile Edge Application, strongly relying on radio network and location information. In our architectural design process, we have identified the ETSI NFV MANO architecture as the most suitable one for supporting different domains due to its flexible interfaces and commonalities across different use cases.

For achieving this cross-domain orchestration our architecture makes use of the Virtual Network Function Manager (VNFM) and the Virtualized Infrastructure Manager (VIM) functional elements introduced within the ETSI NFV architecture. The VIM and the VNFM can be a customized entity dealing with the particular requirements of a certain domain, in this case MEC. As mentioned in the introduction, our assumption is that lightweight virtualization technologies, like containers, will be employed for providing compute, network and storage capabilities to services running on the edge nodes.

Most of the current work in the NFV domain strongly relies upon OpenStack as standard de-facto Virtualized Infrastructure Manager (VIM) [3]. Although OpenStack may be able to provide containers as a service, it consumes quite a lot of hardware resources. Therefore, the most suitable solution is to make directly use of containers deployment tools in order to reduce resource consumption on the edge. The Figure 1 provides a very high level overview of a possible integration between the MEC and the NFV framework.

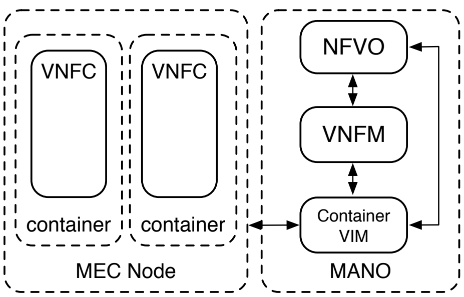

Fig. 1: MEC proposed architecture

In our current approach, the VIM becomes a very small functional component providing a Nf-Vi-H reference point to the Orchestrator and the VNF Manager mainly for three fundamental purposes:

- To provide an interface for deploying containers.
- To allow the VIM to monitor the MEC node through an event system or polling.
- To allow the VIM to send the necessary commands, configurations, alerts, policies, responses and updates to the MEC node, of which it is the sole controller.

In addition to a VIM capable of deploying containers, it is required to have a specific VNFM capable of configuring network services on top of containers. This VNFM receives the VNF descriptors from the NFVO and uses the information for configuring containers accordingly to the descriptor specification.

IV. Prototype implementation based on Open Baton

An implementation of the proposed high level architecture has been realized using the Open Baton NFV MANO framework[4]. A minimal set of extensions have been realized to the core Open Baton components for supporting MEC use cases, while most of the work has been devoted to the implementation of the additional components: a VIM driver and a VNFM able to manage and orchestrate Docker containers maintaining compatibility with the ETSI NFV information model. The Figure 2 exposes the Open Baton architecture.

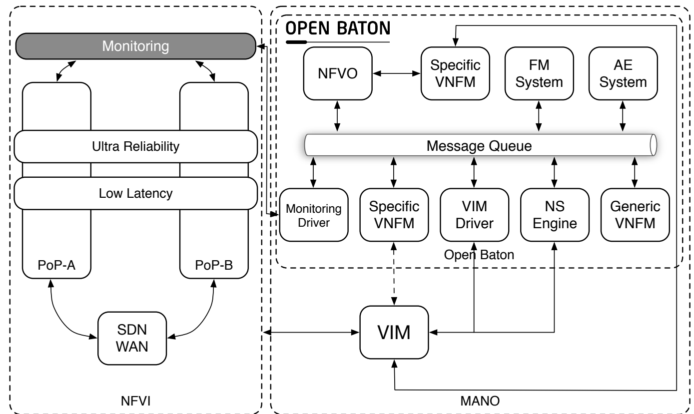

Fig. 2: Open Baton framework

Considering Docker as the container deployment solution it is clear that the first step is to build a VIM driver for Open Baton in order to allow deployment of Docker containers and networks. The NFVO shall only know and use abstractions of the Docker entities exported by the VIM, to guarantee maximum extensibility and reusability of the MANO with other cloud resource managers. This service and its clients will therefore only expose data structures as defined in the Open Baton information model (following the ETSI NFV one). The VIM provides also some authentication functionalities fully integrating with the Open Baton model (based on user roles) reducing at a minimum the external exposure to the Docker daemon through the implementation of only those functions that will be deemed necessary to the client.

Open Baton provides two different mechanisms for allocating resources on the infrastructure. This can be achieved either directly by the NFVO talking to the VIM (using the driver as intermediate), or by the VNFM talking to the VIM. In this specific work, the model chosen is the first one, putting the responsibility of resource allocation to the NFVO itself, instead of relying on the VNFM to carry those duties, to increase the genericness of the solution.

In addition to the Docker VIM driver, it has been necessary to develop a particular VNFM, called Container VNFM, in order to deal with the lifecycle of containerized VNFs. This VNFM receives lifecycle events from the NFVO, and transforms those requests in actions towards the container VIM. Instantiation, modification and start of the containers is obtained composing the information received from the NFVO.

The Figure 3 exposes the final architecture of the proposed solution. The Container VIM has been implemented in Go, using the remote APIs for interacting with Docker. This module exposes an API to the VNFM and the VIM driver via a ”mgmt” protocol. The VNFM and the VIM driver interact with the NFVO via the RabbitMQ message bus. The VIM driver and the VNFM have been also implemented in Go, using the Open Baton Software Development Kits (SDK)[5].

[3]https://www.opnfv.org/

[4]https://github.com/mcilloni/openbaton-docker

[5]https://github.com/openbaton/go-openbaton

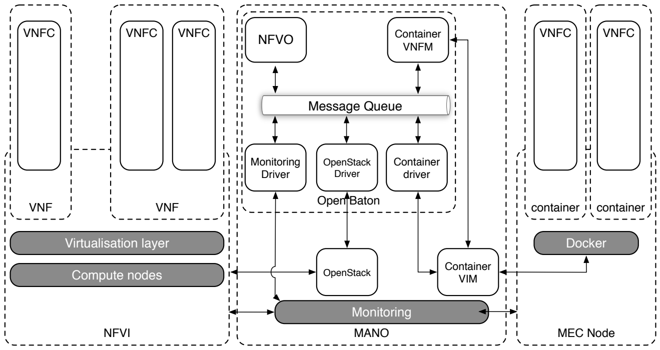

Fig. 3: Integration between docker and Open Baton

V. Early validation results

A beta version of this work has been implemented and validated with a very simple scenario. The idea is to deploy a simple client/server network service in order to validate that the implemented frameworks is capable of instantiating those VNFs on top of containers, and configure them accordingly. Figure 4 shows the architectural model of the deployed network service.

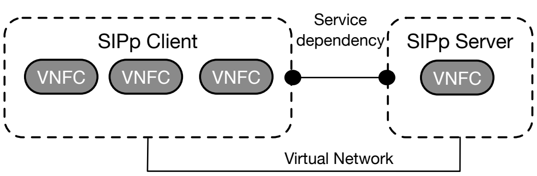

Fig. 4: SIPp Client-Server scenario

The network service deployed is implemented with the SIPp[6] traffic generator. In particular, one VNF acting as client and one as server. Once instantiated the client starts sending SIP traffic to the server, as per their service dependency. This scenario has been validated on the SoftFIRE[7] federated infrastructure, providing a centralized Open Baton instance controlling distributed testbeds across Europe. On top of one of those testbeds has been instantiated a cluster of Docker instances which were registered to the NFVO as an additional PoP. The deployment was validated with the instantiation of multiple containers, and results obtained are in order of seconds. In addition, manual scaling operations were executed and validated.

VI. Conclusion

The work presented in this paper focuses on providing a solution for extending the ETSI NFV MANO framework to support deployment of Software-based networks at the edge of network. The limited resources available at the edge of the network requires the usage of lightweight technologies, like containers, for deploying network functions.

The extensions to the Open Baton framework provide a viable solution for achieving this objective. The initial results obtained with this implementation have been published as open source contributions to the Open Baton community. The solution designed will be further extended to support additional requirements which may be identified while prototyping different use cases.

[6]http://sipp.sourceforge.net/

[7]http://softfire.eu/

Acknowledgment

This work has been partially funded by EU FP7 Integrated Project SoftFIRE. The project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement no. 687860.

References

[1] ETSI Industry Specification Group (ISG), "Network Functions Virtualisation (NFV); Management and Orchestration; Functional requirements specification," ETSI - European Telecommunications Standards Institute, Tech. Rep. ETSI GS NFV-IFA 010 V2.2.1, September 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/010/02.02.01_60/gs_NFV-IFA010v020201p.pdf

[2] ——, "Network Functions Virtualisation (NFV); Management and Orchestration," ETSI - European Telecommunications Standards Institute, Tech. Rep. ETSI GS NFV-MAN 001, December 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf

[3] "White Paper on NFV priorities for 5G," , February 2017. [Online]. Available: https://portal.etsi.org/NFV/NFV_White_Paper_5G.pdf

[4] "Mobile-edge computing introductory technical white paper," ETSI - European Telecommunications Standards Institute, , September 2014. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf

[5] ETSI Industry Specification Group (ISG), "Mobile Edge Computing (MEC); Technical Requirements," ETSI - European Telecommunications Standards Institute, Tech. Rep. ETSI GS MEC 002 V1.1.1, March 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf

[6] ——, "Mobile Edge Computing (MEC); Framework and Reference Architecture," ETSI - European Telecommunications Standards Institute, Tech. Rep. ETSI GS MEC 003 V1.1.1, March 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf

[7] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "ETSI White Paper No. 11: Mobile Edge Computing - A key technology towards 5G," ETSI - European Telecommunications Standards Institute, , September 2015. [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf

[8] J. Anderson, H. Hu, U. Agarwal, C. Lowery, H. Li, and A. Apon, "Performance considerations of network functions virtualization using containers," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2016, pp. 1–7.

[9] K. Ye, X. Jiang, S. Chen, D. Huang, and B. Wang, "Analyzing and modeling the performance in xen-based virtual cluster environment," in *2010 IEEE 12th International Conference on High Performance Computing and Communications (HPCC)*, Sept 2010, pp. 273–280.

[10] R. Behrends, L. K. Dillon, S. D. Fleming, and R. E. K. Stirewalt, "White paper: Programming according to the fences and gates model for developing assured, secure software systems," Department of Computer Science, Michigan State University, 06921 Sophia Antipolis CEDEX, France, Tech. Rep. MSU-CSE-06-2, January 2006.

[11] Y. Yu, "Mobile edge computing towards 5g: Vision, recent progress, and open challenges," *China Communications*, vol. 13, no. Supplement2, pp. 89–99, N 2016.

[12] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-edge computing architecture: The role of mec in the internet of things," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 84–91, Oct 2016.

[13] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2017.

[14] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2017.

[15] B. P. Rimal, D. P. Van, and M. Maier, "Mobile edge computing empowered fiber-wireless access networks in the 5g era," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 192–200, February 2017.

[16] V. Sciancalepore, F. Giust, K. Samdanis, and Z. Yousaf, "A double-tier mec-nfv architecture: Design and optimisation," in *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, Oct 2016, pp. 1–6.