

Comparison of Real-Time Communications Service KPI in Edge and Cloud Domains Through Multi-Domain Transport Networks

Pol Alemany*, Juan L. de la Cruz*, Ricard Vilalta*, Ramon Casellas*, Ricardo Martínez*, Raul Muñoz*, Ana Pol†, Anton Roman†

*CTTC, Castelldefels, Spain, Email:palemany@cttc.es

†Quobis, O Porriño, Spain, Email: [ana.pol,anton.roman]@quobis.com

Abstract—This paper presents the deployment of a Network Slice with a Real-Time Communications Network Service being deployed in two different sites (edge and cloud) of a network comparing the service performance when the user connects to the edge or to the cloud through an optical transport network. This idea has been validated by carrying out tests over a real network testbed.

I. INTRODUCTION

Virtualization is one of the pillars in which networking has evolved to properly work on energy efficiency or on time delays reduction in order to give better service quality to the final user. Having the power to deploy a virtual image of a network function when and where is necessary has given a massive network configuration variability to the network operators never seen before. This is possible because a service can be ready in some minutes or can be removed in few seconds once it is no longer necessary, allowing to use the resources first for one service and then for a new one. All this is done by using Network Function Virtualization (NFV) technologies managing Virtual Network Functions (VNFs) independently or chaining them through a Network Service (NS). One of the latest evolutions within the NFV field to improve the NSs management is the concept of Network Slicing. A Network Slice (NetSlice) is a group of one or more interconnected NSs which aim to give the final user a more complete functionality of the requested service.

Up until some years ago, the idea was to deploy the services inside "the cloud" allowing the user to upload all its information and use any service included there. The cloud is actually any data-center (DC) placed in the core of any network domain. Now this idea is changing due to the huge amount of data being transmitted around the world. This means that the small-resources DCs are being placed on the edge of the network and the low requirements requests are being processed in there. By deploying services closer to the user, the data process is done right on the edge reducing time delays and the DCs in the core are much less congested and focused towards those high requirements services [1].

Following the work done in [2] and [3], this paper describes the deployment of a NetSlice composed by a Real-Time Communications (RTC) NS in both edge and cloud of a

network interconnected through an optical transport network and, finally, compares the performance of the RTC NS in the two different sites. This paper is divided in three sections. The first section presents the NFV elements and the environment architecture used to carry out the experimental tests, a second section describes the obtained results and the third section contains the conclusions.

II. EDGE/CLOUD NFV NETWORK SLICE DEPLOYMENT

This section introduces some concepts regarding the NFV and Network Slicing fields together with a basic structure of each used element. Moreover, it presents the architecture and its structure as well as how the deployments were done in each site.

A. Real-Time Communications Network Service

The most basic element in NFV is a Virtual Network Function (VNF) [4] consisting on a descriptor to create a Virtual Machine (VM) with specific characteristics in order to offer a certain network functionality (i.e. proxy). Having this in mind, an NS is a set of interconnected VNFs (if there is more than one) to properly have a service ready to be used. The life-cycle management of both elements is carried out by a NFV orchestrator (NFV-O).

The NS selected is a Real-Time Communications (RTC) application able to create voice calls or video-conferences sessions with two or more users. Thanks to its internal architecture (Fig. 1), it allowed to create a complex NS composed by five Virtual Network Functions (VNFs):

- Reverse Proxy (VNF-RP)
- WebRTC Application Controller (VNF-WAC)
- Backend Services (VNF-BS)
- Dispatcher (VNF-DS)
- Media Server (VNF-MS)

Furthermore, the NS needs two more components which allow the configuration of the VNFs and the NS:

- Function Specific Manager (FSM)
- Service Specific Manager (SSM)

More detailed information can be found in [5] and [6].

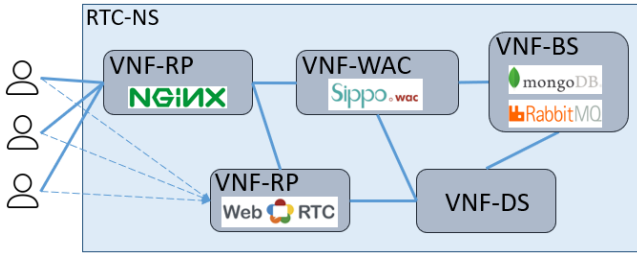


Fig. 1: RTC-NS internal architecture.

B. Real-Time Communications Network Slice

Network Slicing follows the same idea NSs do with VNFs. By linking multiple NSs, the user should have a better user experience and the network owner should be able to manage its network resources more efficiently. The elements that a Netslice Manager (Netslicer) must deal with are two: a Network Slice Template (NST) and a Network Slice Instance (NSI). The first one describes a set of NSs to be deployed and how are they related to each other as well as other possible options. The second element is the record of each instantiation based on the previous NST with the VMs and virtual networks information created in the DC where the NSI is deployed.

Using the Netslicer within the selected NFV-O allowed to specify two aspects:

- In which of the possible (edge and cloud) Virtual Infrastructure Managers (VIM) should the instantiation be done.
- To define the connection points (called ingress) from where users placed in external networks can access the network where the NSIs are deployed.

C. Distributed Architecture for Edge/Cloud NFV Network Services

Fig. 2 shows the architecture generated when both concepts work together. From the data plane point of view, the network is divided in the three common divisions:

- Edge Network: where the Multiple-Access Edge Computing (MEC) idea shows up by having low-resources DCs together with network elements such as base stations for mobile communications or fixed connection access sites. If a user request reaches this point of the network and the station/DC has enough resources, it should answer the request. Otherwise, the request is forwarded to the network.
- Transport Network: interconnects the Edge with either the Cloud or other Transport and Cloud network domains. The architecture used has a transport network with two parallel paths: one is packet-based while the other is an optical network. This second path is composed by four electrical-optical switches running OpenVswitch (OVS) and the four optical switches are managed with a private optical controller.
- Cloud Network: the deepest part of any network where DCs with a massive amount of resources are available to

process those service requests with high requirements. If so, congestion in the transport network is reduced together with delays affecting the user experience.

As the data plane, the control plane has 3 levels:

- On the top there is a NFV-O managing the services life-cycle; deploying them in VIMs, monitoring the NSs behaviour and their SLAs, etc. Currently, there are multiple available NFV-O software such as Open Source MANO (OSM) [7], Open Network Automation Platform (ONAP) [8] or SONATA-NFV [9] (the selected one to use in the evaluation phase).
- Under the NFV-O there are two different elements: VIMs and WAN Infrastructure Managers (WIMs). A VIM is a software in charge of managing the NFV Infrastructure (NFVI) compute, storage and network resources by creating Virtual Machines (VMs) or similar elements such as Containers with the requested service. In the used architecture there are two different VIMs with OpenStack (VM manager): one in the Edge Network composed by a single OpenStack controller/compute node and another in the Cloud Network with a multi-node OpenStack with one controller/compute node and two more compute nodes. A WIM is a Software-Defined Network (SDN) Controllers manager and is in charge (when requested by the NFV-O) to manage virtual interconnections among the transport network elements (e.g. to make the E2E slice interconnections, to configure the connection point where the users connect to get a service).
- At the bottom, there is the network with its elements awaiting to receive commands from VIMs/WIM to accomplish what the NFV-O requested.

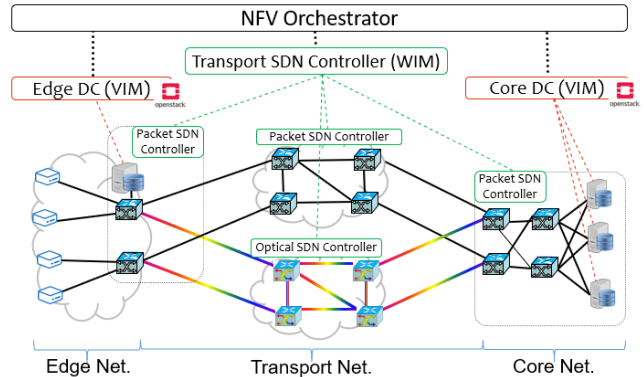


Fig. 2: Edge/Cloud Network Architecture.

III. EXPERIMENTAL EVALUATION

This section presents how the deployments were done, which parameters were used and how were they acquired while testing to be properly evaluated and presented in the last subsection.

A. Edge vs. Core Use Case Deployments

When comparing the performance of a service deployment in two different sites of a network, there are mainly two time

values to consider: the deployment time necessary to have the service ready and the time delays to transmit data between transmitter and receiver.

Regarding the first time and with all the previous elements presented, two different deployments were done. The first one selecting the creation of the NetSlice in the VIM placed on the edge (left side in Fig. 2), while the second one placed in the cloud VIM (right side in Fig. 2). In both cases the deployment time was similar as it mainly depends on how big (MB or GB) the software to install into the new VM is. For this reason, this value was not taken into consideration for the comparison.

The second time depends mainly on how much availability has the network. As different co-existing flows are being processed by network elements, the delay of each individual service data flow is affected and this is the reason why MEC was designed; to reduce as much as possible the traffic within the transport and cloud networks parts.

B. Tests Development

To evaluate the performance in both sites, the RTT and Jitter parameters were used in order to know how long a packet needs to reach the receiver (Jitter) and to come back (RTT).

To properly acquire both parameters, different calls of 10 minutes each were done using the NS deployed in both VIMs. As previously described in subsection II-A, one of the VNFs composing the NS contains a WebRTC [10] which collects data samples in real-time. Then, the samples were processed to create the results presented in Fig. 3 and Fig. 4.

One last aspect to take into consideration before the results are presented is the fact that all the samples were taken in a scenario in which the flows had the best conditions without the concurrence of other flows.

C. Results

An RTC service means two data flows: audio and video. The following subsections present the Jitter results for the audio and the RTT results for the video flow. In order not to repeat descriptions, the parameter not presented has a similar behaviour.

1) *Audio*: Fig. 3 contains the temporal evolution of the Jitter. Comparing the mean values of each case, it is possible to validate the difference between them and confirm that the value on the edge VIM (0.0015 ms) is lower than the mean value in the core VIM (0.041 ms). Regarding the maximum values, on the edge case, there's a peak value of over 0.004 ms and on the core case, its peak value is over 0.11 ms.

2) *Video*: Similar to the figure related to the audio section, Fig. 4 presents how the RTT evolves in time for each data packet with video information. Overall, the RTT mean value for the edge case is 0.39 ms, whereas for the core case is 0.42 ms. Here again, the theory in which the delays from the edge should be lower than those in the core, is also confirmed.

In fact here what is important to remark is that the length of the optical fiber used while the experiments were carried out, was of 30Km which is not a considerable distance. But what matters is that just by having that distance, there is an

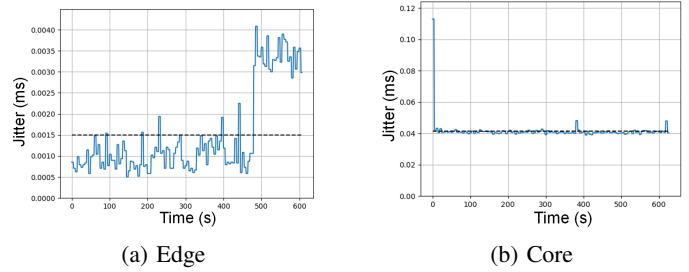


Fig. 3: Audio Jitter over time.

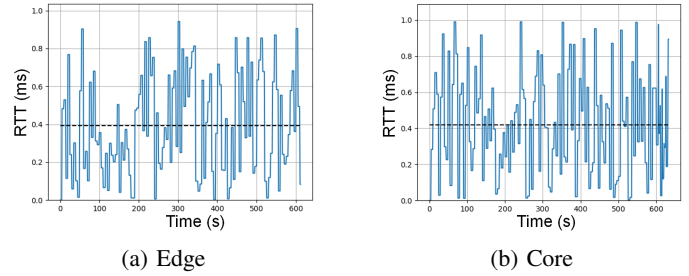


Fig. 4: Video RTT over time.

increment of 7,3% of the RTT. This means that with transport networks of higher distances, the RTT should increase over 10% easily, and this is a considerable delay to take into account for low-delay services like the RTC service presented.

IV. CONCLUSIONS

This paper proved that by placing the services deployments closer to the user, data transmissions are more stable in reception and the transmitter must wait less time in order to ensure the transmitted data has been well-received.

ACKNOWLEDGMENT

Work Funded by the EC through 5GPPP 5GTANGO (761493) and Spanish AURORAS (RTI2018-099178).

REFERENCES

- [1] R.Vilalta, et al. *SDN/NFV orchestration of multi-technology and multi-domain networks in cloud/fog architectures for 5g services*, IEEE 21st OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS), 2016.
- [2] C.Parada, et al. *5GTANGO: A beyond-MANO service platform*, IEEE European Conference on Networks and Communications (EuCNC), 2018.
- [3] P.Aleman, et al. *Experimental Validation of Network Slicing Management for Vertical Applications on Multimedia Real-Time Communications over a Packet/Optical Network*, IEEE 21st International Conference on Transparent Optical Networks (ICTON), 2019.
- [4] ETSI GS NFV-MAN 001 v1.1.1 *Network Functions Virtualisation (NFV);Management and Orchestration*, ETSI 2014.
- [5] G.Xilouris, et al. *D7.1 Evaluation strategy and testbed setup report*, EUC 5GTANGO Project 2017.
- [6] D.Behnke, et al. *D7.2 Implementation of pilots and first evaluation*, EUC 5GTANGO Project 2019.
- [7] *OSM Home page*, OSM official site. <https://osm.etsi.org/>
- [8] *ONAP Home page*, ONAP official site. <https://www.onap.org/>
- [9] *SONATA-NFV Home page*, SONATA-NFV official site. <https://www.sonata-nfv.eu/>
- [10] *WebRTC Home page*, WebRTC official site. <https://webrtc.org/>