A 5G Facility for Trialing and Testing Vertical Services and Applications

Sagar Arora, Karim Boutiba, Mohamed Mekki, Adlen Ksentini

A 5G facility for trialing and testing vertical services and applications

Sagar Arora, Karim Boutiba, Mohamed Mekki, and Adlen Ksentini, *Senior member, IEEE*

***Abstract*—With its low latency and high speeds, 5G aims to support vertical services like automotive, industry, agriculture, and manufacturing. Unlike the precedent mobile generations, which just provided voice and data to domestic and business customers, 5G aims to create a common infrastructure to support diverse requirements of the vertical industry's needs in terms of communication and networking capabilities, such as high data rate and low latency. But before a commercial deployment, a trial phase is needed to validate that the network can support these requirements. In this context, several 5G facilities have been established to run 5G trials, mainly built to be used by networking experts. However, vertical service owners have small to no knowledge of the technical details of the 5G infrastructure. In this paper, we introduce EURECOM 5G facility, which was specifically designed to run vertical use cases by abstracting and simplifying as much as possible the trial deployment and Key Performance Indicator (KPI) collection. EURECOM 5G facility provides a rich number of 5G components to test, including 5G New Radio, Network Slicing, Edge Computing, and KPI visualization, allowing verticals to have a real 5G environment for testing their applications and services. Finally, the facility relies mainly on open-source components.**

***Index Terms*—Vertical industry, trials, tests, 5G**

I. Introduction

Unlike the precedent generation of mobile networks, the 5G architecture has been specifically designed to support novel network services, including vertical industry use cases, which have different requirements than classical broadband mobile applications and services. Vertical industry services require, for instance, low latency, high reliability, high mobility support, and in some cases, high bandwidth with Uplink-dominated traffic. Examples of such services are: industry 4.0, autonomous driving, Unmanned Aerial Vehicles (UAV), Augmented and Virtual Reality (AR/VR), eHealth, smart cities, etc. Before the advent of 5G, the vertical industry employed different proprietary networks or networks that did not completely fulfill the needed requirements in terms of performance. In this vein, the 5G architecture is designed to use a common infrastructure to sustain different types of network services, including vertical industry use cases. Thanks to network slicing, a novel concept introduced in 5G, virtual instances of the network are created and tailored to sustain applications' needs using a shared infrastructure. In 5G, applications and network services are classified according to their requirements via three types of: (i) enhanced Mobile BroadBand (eMBB); (ii) ultra Reliable Low Latency Communications (uRLLC); (iii) massive Machine Type Communications (mMTC). For each class, a set of networking requirements has been established, which need to be fulfilled by the network. In order to support these requirements, different technological improvements have been achieved by 5G compared to 4G. These enhancements concern all the system components, i.e., all Radio Access Network (RAN) layers, including 5G New Radio (NR) that allows reaching gigabits per second and the cloud-native 5G Core Network (CN). Besides, the 5G architecture heavily relies on cloud and edge computing to add flexibility and agility as well as to guarantee low latency (thanks to edge computing).

The first commercial deployment of 5G, known as Non-Standalone (NSA), started in the summer of 2019. It deploys only 5G NR while using the 4G CN, aiming to improve broadband connectivity that provides a high data rate for the end users. More recently, the second deployment phase has started, expecting a full deployment of the 5G system, including the new 5G CN featuring network slicing to unleash the 5G capacity to support vertical industry use cases. However, before any commercial deployment of a new service, a trial phase is expected, where the application or the network service is tested (or trialed) and validated using an operational network. Indeed, the trial step will allow the vertical owner to ensure that its service can safely run on top of the new network by checking that needed Key Performance Indicators (KPI) are satisfied. For example, it is important in the case of flying drones to ensure that low latency is supported to ensure reliability and safety. The trial step will also allow for testing different configurations of the 5G infrastructure to understand the best one that runs the vertical service optimally.

In this paper, we will present EURECOM's 5G trial facility, which permits to trial and validate vertical services on top of a 5G SA infrastructure composed mainly of Open-source tools and enables the testing of far-edge computing. The facility, by design, is unique compared to the existing ones. It was specially devised to abstract the details of the low-level components to simplify as much as possible running trials and collecting KPIs without being experts in 5G. The facility features are: (1) test vertical applications developed as a monolithic application or composed of micro-services, including both the client and server-side; (2) automate the configuration and deployment of a trial as well as results collection via a high-level abstracted interface; (3) explore different 5G radio configurations by testing different network slice types; (4) test the deployment at the edge and validate Multi-access Edge Computing (MEC) service API: (5) test the end-user part of the vertical service using cots 5G User Equipment (UE). All these features make the facility unique for testing advanced vertical scenarios and use cases. Moreover, all the components

Authors are with communication systems department of EURECOM, France. contact: name.surname@eurecom.fr

are open-source, which allows a constant improvement of the facility components (i.e., continuous integration).

The rest of this paper is organized as follows. In Section II, we present a set of existing facilities that allow running tests and trials. Section III details the facility's architecture, components, and functioning. In Section IV, we introduce some performance results regarding the infrastructure as well as application performances. Finally, section V concludes the paper.

II. Related work

Several facilities and test platforms exist. We will discuss some of them in this section.

Powder (the Platform for Open Wireless Data-driven Experimental Research) is a facility for experimenting the future of wireless networking scenarios in a city-scale "living laboratory" [1]. Its objective is to foster experimental research for a range of heterogeneous wireless technologies [2]. POWDER-RENEW is equipped with cutting-edge computing, storage, and cloud resources, as well as state-of-the-art Software Defined Radios (SDR). These include open-source RAN software, e.g., OpenAirInterface(OAI), srsLTE, and the OpenRAN Real-time intelligent Controller (RIC). Moreover, the POWDER-RENEW platform has been used to demonstrate automated optimization of 5G networks in [3]. COSMOS Cloud Enhanced Open Software Defined Mobile Wireless Testbed for City-Scale Deployment project is aimed at the design, development, and deployment of a city-scale advanced wireless testbed in order to support real-world experimentation on next-generation wireless technologies and applications. Researchers will be able to run experiments remotely on the COSMOS testbed by logging into a web-based portal which will provide various facilities for experiment execution, measurements, and data collection [4]. Colosseum is the world's most powerful wireless network emulator [2]. It is housed at Northeastern University Innovation Campus in Burlington. It is a wireless emulator with 256 programmable software radios. It enables academic, government, and industry researchers to perform scalable and repeatable experimentation in wireless systems in a large-scale emulation environment [5]. Arena 5G is an indoor testbed that allows researchers to experimentally evaluate wireless protocols and solutions for indoor 5G deployments in an office-like environment. For instance, Arena 5G can be used to evaluate the performance of standard-compliant cellular networks through the OAI and srsLTE protocol stacks. Arena 5G has been used to demonstrate future cellular network capabilities, 5G RAN optimization and RAN slicing [2]. The 5TONIC co-creation laboratory [6] was established to provide an open environment where members from business, industry, and academia could collaborate with the telecoms community on specific 5G mobile research and innovation projects. The aim is to support innovation and help organizations work together to develop and deliver market-ready 5G solutions, technology, applications, and business ventures. The 5TONIC laboratory includes a solid baseline of facilities, infrastructure, and equipment to support advanced experimentation in the 5G virtual network function and wireless systems areas. The 5GENESIS facility [7] provides a flexible and open experimentation suite with network slicing in order to support and facilitate validation of vertical industry KPIs over the 5G infrastructure. However, their infrastructure relies on closed-source software, which does not allow testing new algorithms at the network level (RAN and CN). Besides, the considered KPIs are limited to the throughput, latency, and radio coverage statistics which may not be sufficient to validate the use cases. The 5G-VINNI [8] (5G Verticals Innovation Infrastructure) project provides a set of interconnected facilities in UK, Norway, Spain, and Greece. Each facility covers fixed/multi-radio access, backhaul, core network, service technologies, and architectures targeted for 5G, including end-to-end virtualization and slicing as key components to support vertical use cases. However, their infrastructure relies on closed-source software, which does not allow testing new algorithms or adapting NFs to enable new use cases for 6G. Besides, they don't provide a methodology to validate the use cases, such as KPIs validation.

It should be noted that most of the mentioned facilities require a strong knowledge of low-level technologies to run the trial. This knowledge is not easy to have from actors outside the network community, such as the vertical industry actors. The latter are generally experts in their field and have small to no knowledge of the low-level technologies constituting the 5G infrastructure and system. EURECOM 5G facility has been, by design, devised to abstract as much as the complexity of the infrastructure, and without or little knowledge of 5G in order to allow actors to run trials and tests on top of the 5G facility.

III. EURECOM 5G Facility

As stated earlier, testing and validating vertical services' KPI is critical before a commercial deployment on top of 5G networks. EURECOM 5G facility has been designed specifically to provide the vertical with a high-level system to run a trial and collect KPI, without taking care of the system's complexity and low-level information. Figure 1 shows a high-level architecture of the facility. Three layers are distinguished: the vertical and user space, orchestration and management, and infrastructure. The user layer is where the vertical and the trial owner interact with the facility to define, run and monitor a trial. It is mainly composed of the webportal. The orchestration and management layer comprises all the entities managing the life-cycle of the trial, i.e., configuring, instantiating, running the trial as a network slice, and monitoring the KPI. Finally, the infrastructure layer is composed of elements that run the 5G components, such as RAN, CN, MEC applications, Virtual Network Functions (VNF), and MEC Platform (MEP). Due to space limitation, we will not provide technical details on the infrastructure layer, which mainly relies on OpenAirInterface (OAI) and Kubernetes for edge and far-edge computation platform.

A. The user layer: webportal

The webportal is the facility's key element, as it is the interface with the vertical and trial owner. The webportal

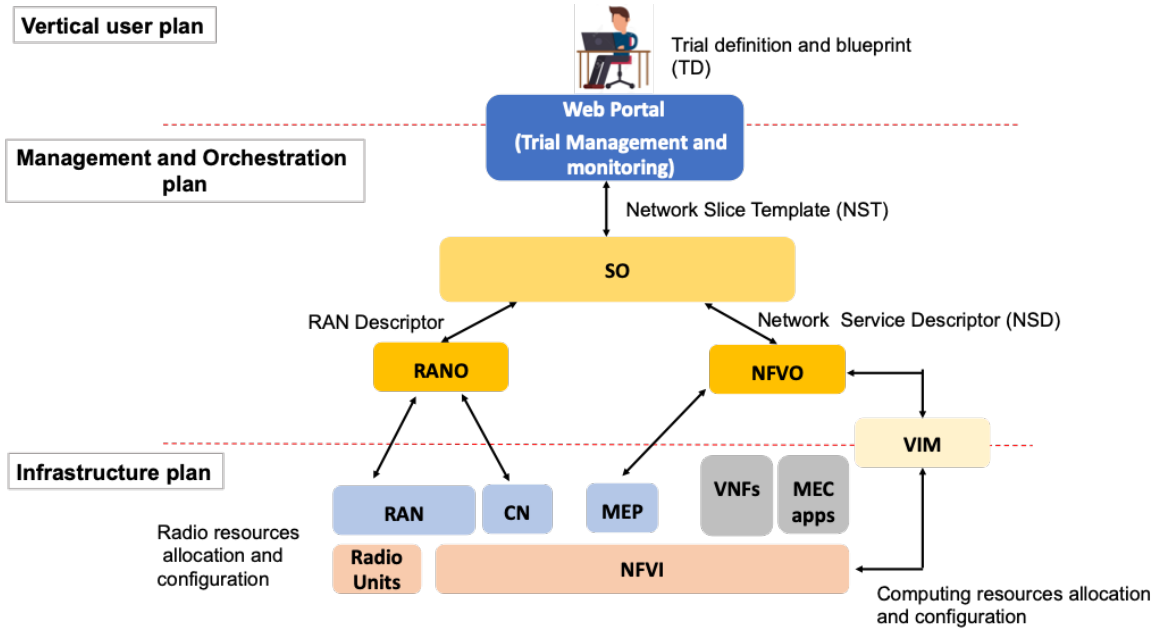

Fig. 1. The facility architecture

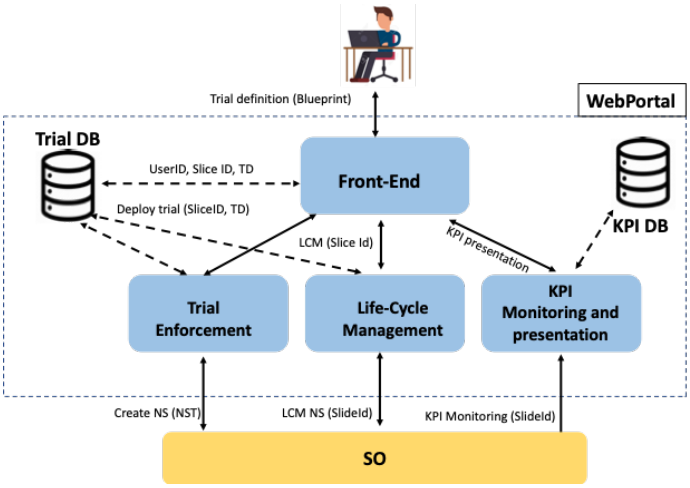

Fig. 2. The webportal components

aims to abstract the 5G components by providing a high-level view of the trial management to the vertical, i.e., to deploy and monitor a trial. Figure 2 illustrates the webportal architecture, which comprises a front-end, trial enforcement, life-cycle management, and KPI monitoring and presentation, as well as two databases (DB). All the components collaborate to ensure the trial's life-cycle, consisting of the definition and preparation, configuration and instantiation, run-time management, and deletion.

1) Trial definition and preparation: The trial definition and preparation are done by the vertical using the front-end Graphical User Interface (GUI), which corresponds to a webportal. This step consists of filling out a form that describes the trial scenario, the network resources, and the KPI to measure. Besides meta-data information on the trial, such as the start and end time of the trial, filled data concerns all the components of the network slice needed to run the trial. EURECOM facility allows the vertical to specify the information on the RAN and the vertical services that should run at the cloud or MEC. For the RAN, the needed information is the type of the requested slice (i.e., uRLLC or eMBB or mMTC), maximum latency or minimum bandwidth, and the UEs identifiers (SUPI) that are allowed to connect to the network slice. The RAN Orchestrator (RANO) will later use the RAN information to estimate the necessary radio resources to satisfy these requirements. Regarding the network service to be deployed, the facility allows the deployment of monolithic or micro-service-based applications. Furthermore, the network service can be composed of one or more applications, but also applications that can run on the client side, i.e., on the user device. The trial owner can deploy not only services at the edge but also on the device side to be able to test the server side of the application from a 5G device.

To define a network service (i.e., a set of applications connected together to provide a service), a tenant uses the GUI to create a Network Service Descriptor (NSD), which will contain one or more applications defined using the Application Descriptor (AppD) model of MEC ETSI [11]. We extended the AppD with one field that indicates the type of deployment: edge or far edge (i.e., on the 5G device). Again, the tenant can use the GUI to fill the AppD field, simplifying the configuration process. Consequently, the vertical does not need to know about the NSD and AppDs formats. Indeed, the vertical has to fill the form, and automatically the NSD with all AppDs is generated. At this step, the vertical needs to provide information, such as the location of the application image(s) to deploy both on the edge and far edge, the amount of CPU as well as Memory to assign to the application. Finally, the vertical can select, from a list, the KPI to monitor.

As output, the front-end produces a Trial Descriptor (TD) that contains all the information entered by the trial owner, i.e., meta-data, RAN information, NSD with the list of AppD, and the KPI to monitor. New information is added to the meta-data part, which is the trial Identifier (ID) generated by the front end. The trial ID is used to identify the trial, as a vertical may run several trials of the same scenario but with different network configurations. The TD is stored, along with other information (like the vertical ID), in the Trial DB.

2) Configuration and instantiation: This phase starts after the generation of the TD by the front-end module. Once the TD is stored in the Trial DB, the Trial enforcement is called. The Trial enforcement translates the TD to a Network Slice Template (NST) and uses the NBI of the facility Slice Orchestrator (SO), to first request the configuration of the Network Slice and check the resource availability. Once the request is accepted by the SO (a Slice ID is created and sent back), the Trial enforcement requests the instantiation of the Network Slice using the returned Slice ID. When the SO confirms the instantiation of the network slice, meaning that the trial can start, the Trial enforcement updates the status of the trial to "running" in the DB and informs the front end that the trial can start. The front end displays this information on the GUI (i.e., a web page), showing the trial's status, and allowing the vertical to start the monitoring process. Once done, the front end forwards the request, including the Slice ID and KPI list (obtained from the Trial DB), to the KPI monitoring and presentation module. The latter sends the request to the SO along with the Slice ID and KPI list. The SO replies with two URLs; the first is to access the dashboard to visualize KPI in real-time, and the second is to subscribe to a broker to access the data stream representing the KPI in raw data form. The latter will allow the vertical to store data on the trials for future usages, such as training Machine Learning (ML) models. Then, the KPI monitoring and presentation module creates an entry in the KPI DB, where the Slice ID and the corresponding URLs are stored. Then, it forwards the URLs to the webportal, which displays them to the trial owner. The latter can decide to use only the dashboard, subscribe to the broker's URL to obtain the raw data, or use both.

3) Run-time management: The front end allows the vertical to update the assigned computing and network resources to a running trial. Again, this can be done through the webportal, where the vertical selects the resource type. Two possibilities are given to the vertical, update the RAN resource by requesting more or fewer radio resources; and update the computing resources of a running application. For each running application at the edge, the vertical can specify new values for CPU and memory. The webportal redirects the request, with the Slice ID, to the Life-cycle management module. The latter uses the NBI of the SO to update the network slice resource. The SO confirms or rejects the update if there are not enough resources, and hence the vertical is informed about the status of the request.

The vertical can also resume a trial and restart it. The resume step consists in sending a request to the SO through the life-cycle module to stop the slice without deleting its associated computing and network resources. The SO also postpones the collection of KPI. The restart procedure consists in instantiating the network slice again. When resumed, the trial status in the Trial DB is updated accordingly.

4) Deletion: The vertical, when deemed appropriate, can manually stop and delete the trial before the end via the webportal. Then, the request is sent to the SO via the life-cycle module. Unlike the resume case, the SO will stop the slice and remove all the resources dedicated to it. The virtual image of the applications is off-boarded from the computing infrastructure (NFVI). The trial DB is updated by removing the Slice ID corresponding to a trial. The front end proposes to the vertical if the TD should be stored for future use, for instance, as a Blueprint. If the vertical accepts, the TD is not removed from the DB. It will be proposed as a Blueprint to create another trial.

5) Monitoring: Monitoring the performances of the different components is a critical process when testing a network service. Indeed, the vertical needs to extract useful information regarding the behavior of its applications from the infrastructure point of view. While the vertical can easily extract service level KPI, it can be very pertinent to combine them with infrastructure KPI to build root cause analysis and improve the performance of its applications. We grouped the collected KPI into three groups: one on the RAN (such as latency, uplink, and downlink data rate), one on the edge cloud (such as CPU and memory usage as well as data rate), and finally, one on the network slice level (such as the time needed to deploy and decommission a network slice). Readers can refer to [9] for more details on the monitoring mechanism used in the facility.

It should be noted that when writing these lines, we do not provide monitoring information on applications running at the end device or far edge.

B. Orchestration and Management plan

As stated earlier, a trial or a test in EURECOM's facility is run as a Network Slice, which needs to be orchestrated and managed to run properly on top of the 5G infrastructure of the facility. The orchestration and management plan is composed of the Slice Orchestrator (SO), which is in charge of the Life Cycle Management (LCM) of the Network Slice (NS), the RANO that manages the LCM of the RAN part of a NS, and the Network Function Virtualization Orchestrator (NFVO) that manages the LCM of the applications described using AppD. Following the 3GPP management, model [10], the facility SO corresponds to the Network Slice Management Function (NSMF), while NFVO and RANO to Network Sub Slice Management Function (NSSMF).

1) SO: It is the entry point of the management and orchestration layer. It exposes a Northbound Interface (NBI) to the webportal for the NS LCM and monitoring management. It uses the NBI exposed by the NFVO and RANO to deploy a Network Slice on each domain and starts the monitoring process. The SO of the EURECOM 5G facility implements the NS LCM as specified by 3GPP [10], which covers the functions related to commissioning (allocate and activate a network slice), operations (read and modify network

slice configuration), and decommissioning (de-allocate and de-activate a network slice). In the implementation of the SO, we decided to group the allocate and activate in one API call (create) exposed to the webportal in order to avoid ambiguity for the vertical, which can use only one action "create". The SO will take care of both steps, allocate and activate. Indeed, once receiving a request to create a Network Slice from the webportal, the SO extracts from the NST, the NSD, and the RAN template, which are communicated to the NFVO and RANO, respectively. This step corresponds to allocating the necessary resources to the network slice at the NFVO and RANO. The latter validate the request if enough resources are available to satisfy the resources specified in the NSD and RAN template. They send back a NSD Identifier (ID) and RANID, which correspond to the sub-slice identifier at the NFVO and RANO. If either the NFVO or the RANO rejects the request, the SO rejects the request and notifies the webportal immediately, indicating that not enough resources are available to run the network slice (i.e., the trial). Otherwise, SO starts the activation step by sending a message to the NFVO and RANO to instantiate the sub-slices (NSDID and RANID) and create the monitoring agents that collect the KPI metrics specified by the vertical. Both NFVO and RANO acknowledge the success of the operation by sending back information on the sub-slices (using a confirmation message), which will be sent by SO to the webportal along with the URLs to observe the monitored KPI and subscribe to the broker to collect KPI raw data. The message also includes the KPI regarding the time needed to create the network slice calculated by SO. Finally, SO changes the status of the network slice to running in DB.

Moreover, SO supports the run-time update of the network slice configurations; specifically it allows updating the radio resources dedicated to the radio sub-slice and the computing resources for deployed applications of the vertical. Finally, the deactivate step consists in stopping the slice and the corresponding sub-slices by keeping the resources allocated to the slice, which allows the vertical to resume the slice. In contrast, the deallocate step removes all the resources dedicated to the slice and deletes the entry in the DB.

2) NFVO: The NFVO role is to deploy vertical applications, which are defined in the NSD part of a Network Slice over the virtualization platform. The NFVO takes as inputs the NSD, including the list of AppD. At the network slice allocation step, it first checks if enough computing resources are available to run the applications described in the NSD. If yes, it onboards the virtual images on the VIM. This is generally the most time-consuming action. It consists in downloading the software images provided via URL in the AppD. At the network slice activation step, the NFVO requests the instantiation of the applications that have been onboarded into the VIM. This action will deploy each application as a container to run on top of Kubernetes or Openshift for edge applications. In contrast, a lightweight container management technology is considered for the far edge. Indeed, we used K3S, a version of Kubernetes, which is a very lightweight cloud-native management system. It needs a reduced footprint in terms of the needed computing resources.

It should be noted that deploying edge applications implies the need of traffic redirection. Therefore, NFVO interacts with the MEC Edge Platform (MEP) to guarantee that the different instances will receive the traffic coming from UEs. Finally, the NFVO also creates a monitoring agent that collects data on the CPU, memory usage, and networking information of the applications belonging to a specified slice.

The NFVO allows updating the computing resources dedicated to each application. The solution we adopted is to duplicate the container running the application, increase the computing resources of the new container, and stop the old container. When the NFVO receives the deactivate request, it stops the application instance as well as the monitoring agent. The application is stopped by destroying the container(s) running the application but keeping the image(s) onboarded; while when it receives the deallocation request, NFVO off-boards the software image(s) from the VIM (including the far-edge nodes) and deletes the monitoring agent dedicated to the edge sub-slice.

3) VIM: The role of the VIM in the facility is to handle the life-cycle of containers (Onboard/Offboard, Instantiation, Run-time, Update, Terminate) deployed on the edge infrastructure or the end device. The developed VIM [11], particularly for the edge, supports micro-service deployment using a different combination of PoDs and containers. To recall PoDs are the smallest schedulable entity in Kubernetes. They provide an ecosystem for multiple containers to interact. More details on these patterns are available in [12].

We have developed three types of VIM that can deploy containers on top of Kubernetes, Openshift, and K3S. The NFVO selects the appropriate VIM according to the supported type of container infrastructure. The VIM is similar in terms of functionalities, whatever the container infrastructure to use. The only difference is in the Southbound API that communicates with the infrastructure manager, i.e., Kubernetes, Openshift, or K3S. Our objective is to have a modular VIM that can be updated if a new container infrastructure needs to be used by just adding a new southbound API plug-in. The devised VIM uses an image registry database to handle the container images (i.e., application images). When NFVO receives a request to onboard an image, the image is first downloaded into the image registry from which the image is built and pushed towards the local repository of the container infrastructure. The VIM supports Download/Build/Pull and push images in the local infrastructure repository in the following formats: Compressed (Tar format), Git repository, Public container repository, and Internal repository.

4) RANO: The RANO aims to manage the RAN resources dedicated to a Network Slice. The RANO is equivalent to the Non-real time controller defined by the O-RAN alliance [**?**]. It also uses Real-time Intelligent Controller (RIC) to monitor and control gNB. At the allocation step, RANO checks the availability of the radio resources as expressed in the RAN template. To do that, the RANO relies on the resource estimation algorithm introduced in [14] to translate the requirement in terms of latency or throughput into the needed Physical Resource Blocks (PRB) and check if the latter are available or not. The same process is conducted

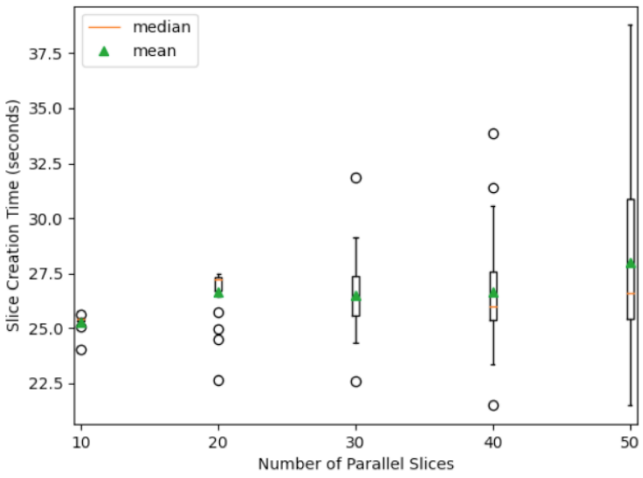

Fig. 3. The needed time to deploy a network service vs number of parallel network services (run as a edge slice)

when the vertical wants to increase radio resources on runtime. At the activation phase, the RANO relies on the two-level architecture introduced in [15] to isolate and enforce the RAN slice by creating the appropriate L2 scheduler. Per the SO request, the RANO creates a monitoring agent to collect RAN KPI regarding a specific slice. The monitoring agent is run as a xApp (i.e., applications running at RIC) that collects KPIs from the gNB. At the deactivate step, the monitoring agent (xApp) is stopped, while at the deallocation phase, the xApp and the L2 scheduler dedicated to the radio sub-slice are deleted.

IV. Performance Evaluation

In this section, we present some results showing the facility's performances from the vertical point of view when deploying a test (for example, the time needed to deploy and start a test). We tested the deployment of different configurations of a network service, at the edge cloud, in terms of the number of composed vertical applications. We also tested the time needed to deploy over the far edge. It should be noted that SO and NFVO have been implemented from scratch using Python3.

We divided the tests into three parts. The first part is about the NSD deployment at the edge, the second part concerns the deployment on the far edge, while the last part shows KPI of a Mission Critical Service (MCS) application deployed on the facility [1]. For the first part, we focused on the time to deploy a NSD (part to be deployed at the edge), as the time to accommodate a RAN sub-slice is negligible. Each NSD contains a different number of container images and AppDs. It is worth recalling that the case of having several software images (i.e., container) indicated in only one AppDs corresponds to a micro-service deployment [12]. The container images of Network Service (N) 1, N2, and N3 were already present in the cluster image repository, meaning that the time to onboard is lower than the case where the image is present

[1]https://www.youtube.com/watch?v=8NvExnbb33M

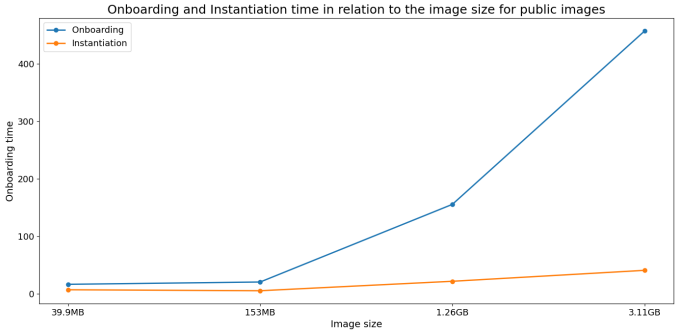

Fig. 4. Far edge performance - deployment time

in a public or a private directory (available on the Internet). For N4, one of the container images was present in a public repository, while the others were present in the local cluster. Table I summarizes the time taken to create and delete each NSD. For each network service, the tables include the number of AppD and the total number of container images to deploy (i.e., onboard and instantiate). We remark that the time to deploy a network service is higher than the time to delete, mainly due to the onboarding processing and particularly the time to download the image from the local or public repository. Further, we observe that deploying a network service with higher number of images takes more time, particularly for N4, which has one image in a public directory. This is explained by the fact that VIM has to pull or build software images, schedule the container/pods, attach a network interface and assign an IP address. To assess the facility's performance when parallel slices are deployed, i.e., parallel experiments are run. We started by creating 10 slices, and after 30 seconds of runtime, we deleted them. This pattern was continued for handling 10, 20, and up to 50 network services (or edge sub-slices). All the slices used the same N1, and the container image was already present in the cluster image repository. We performed the same experiment 100 times for each data point using Monte Carlo simulation. Figure 3 illustrates the creation time of the network service. Like the precedent result, the creation time is the most time-consuming process. The mean and median time needed to create a network service varies between 25 and 28 seconds.

The second part of the tests is dedicated to the far edge deployment. To this aim, we deployed different applications with different sizes at the COTS UE running a K3S instance and measured the time needed to onboard and instantiate an image. The results are shown in Figure 4. Similarly to the first tests, we remark that the time of onboarding images takes more time than the instantiation. Moreover, the time of instantiation and onboarding is proportional to the image size. We remark that more than 400 seconds is needed to onboard an image (available in a public repository) exceeding 3 Gbps. We can conclude from the results of the two first parts that the time to deploy a complete trial is not very large, even with the large image size of the applications.

In this last part of the results, we show the performance experienced by an MCS application deployed on the facility.

TABLE I
NEEDED TIME FOR DEPLOYING 4 NS

Network Service	Creation Time (s)	Deletion Time (s)	AppD	SwImage
N1	20.42	10.32	1	6
N2	15.26	10.32	1	1
N3	30.52	15.29	2	6,1
N4	50.43	20.4	3	1,1,1

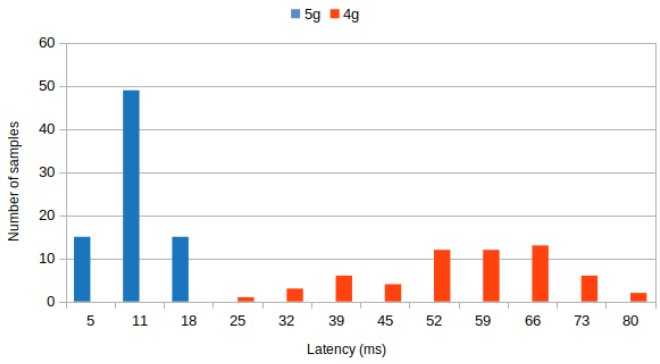

Fig. 5. Latency observed by a MCS application

Figure 5 shows the latency (in ms) observed by the application for different video packets using 5G and 4G networks. The application runs at the edge. We focused on the latency KPI as it is critical for MCS services; communication latency should be near-real between the first responders. We remark from the figure that the end-to-end latency in 5G/Edge deployment is less than 20ms, 11ms on average. In contrast, running the application in a 4G/edge deployment increases the average latency to more than 50ms.

V. CONCLUSIONS AND THOUGHTS

In this paper, we presented the EURECOM 5G facility to run vertical service trials. The facility provides a rich number of 5G components, including the support of network slicing, new 5G Radio, and deployment at the edge as well as far edge. The facility has been devised to abstract the complexity of the lower layer of the 5G infrastructure, aiming at simplifying to non-expert persons the deployment of trials and KPI monitoring. Indeed, 5G involves many components that make the definition of a trial very difficult, particularly when a service involves a lot of applications, considering that the creation of NSD is very difficult and complex if we follow the NFV standards. Therefore, the solution proposed through the webportal permitted abstracting this complexity and helped define a very complex scenario such as the one trialing MCS.

The facility has been used by several European projects, such as 5G!Drones and 5GVictori; two projects dedicated to vertical industry trials. From these projects, we obtained useful feedback from vertical players, which allowed us to update the design of the webportal to improve the facility's usability to run the trials.

Finally, the facility is continually improved with new components for B5G and 6G. Indeed, in the near future, the facility will support 5G millimeter wave and Re-configurable Intelligent Surface (RIS), which will allow for trialing 6G vertical use-case scenarios.

ACKNOWLEDGMENT

This work was partially supported by the European Union's Horizon 2020 Research and Innovation Program under 5G!Drones project (Grant No. 857031) and SLICE-SC project (Grant No. 101008468).

REFERENCES

[1] POWDER: https://powderwireless.net/
[2] Leonardo Bonati, Michele Polese, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia, "Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead," Computer Networks (COMNET), vol. 182, December 2020.
[3] L. Bonati, S. D'Oro, L. Bertizzolo, E. Demirors, Z. Guan, S. Basagni, and T. Melodia,"CellOS: Zero-touch Softwarized Open Cellular Networks," Computer Networks, vol. 180, pp. 1–13, June 2020.
[4] Cloud Enhanced Open Software Defined Mobile Wireless Testbed for City-Scale Deployment (COSMOS). http://www.cosmos-lab.org.
[5] Colosseum. https://www.colosseum.net.
[6] 5G Tonic: https://www.5tonic.org/platform.
[7] M. Christopoulou et al. "5G Experimentation: The Experience of the Athens 5GENESIS Facility", IFIP/IEEE International Symposium on Integrated Network Management (IM), Bordeaux, France, 2021.
[8] M. Ghassemian, P. Muschamp, D. Warren, "Experience Building a 5G Testbed Platform", 2020 IEEE 3rd 5G World Forum (5GWF).
[9] M. Mekki, S. Arora, A. Ksentini, "A Scalable Monitoring Framework for Network Slicing in 5G and Beyond Mobile Networks", IEEE Transactions Network Service Management 19(1): 413-423 (2022)
[10] 3GPP specifications, Management and orchestration; Concepts, use cases and requirements, TS 28.530.
[11] S. Arora, A. Ksentini, C. Bonnet, "Lightweight edge slice orchestration framework", ICC 2022, 16-20 May 2022, Seoul, South Korea.
[12] S. Arora, A. Ksentini, C. Bonnet. "Availability and latency aware deployment of Cloud native edge slices", in IEEE Global Communications Conference, Globecom 2022;8 December 2022, Rio de Janeiro, Brazil
[13] B. Brik, K. Boutiba, A. Ksentini. " Deep Learning for B5G Open Radio Access Network: Evolution, Survey, Case Studies, and Challenges", IEEE Open Journal of Communications Society, 228-250 (2022)
[14] S. Bakri, P. A. Frangoudis, A. Ksentini, M. Bouaziz. "Data-Driven RAN Slicing Mechanisms for 5G and Beyond", IEEE Transactions Network Service Management 18(4): 4654-4668 (2021)
[15] A. Ksentini, N. Nikaein. "Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction", IEEE Communications Magazine 55(6): 102-108 (2017)

BIOGRAPHIES

Sagar Arora is in the final year of his doctoral studies at Eurecom, Sophia Antipolis. His doctoral thesis is on designing a network slice orchestration framework for cloud-native container based 5G network functions and MEC applications. Currently he is employed at OpenAirInterface Software Alliance

Karim Boutiba is pursuing a Ph.D. in the Communication Systems department at EURECOM, France, under the supervision of Pr. Adlen Ksentini. He is working towards enforcing

Network Slicing in 5G networks and beyond. His research interests include Next-Generation Networking, 5G New Radio, Network Slicing, Open RAN and Reinforcement Learning for 5G networks and beyond.

Mohamed Mekki is currently a doctoral student at EURECOM, Sophia Antipolis. His doctoral thesis is on Cloud Edge Continuum (CEC) to support emerging network services that require low latency and high bandwidth usage.

Adlen Ksentini is an IEEE COMSOC distinguished lecturer. He obtained his Ph.D. degree in computer science from the University of Cergy-Pontoise in 2005. Since March 2016, he is a professor in the Communication Systems Department of EURECOM. He has been working on several EU projects on 5G, Network Slicing, and MEC.