Experience Building a Prototype 5G Testbed

Citation for published version:
Foukas, X, Sardis, F, Foster, F, Marina, MK, Lema, MA & Dohler, M 2018, Experience Building a Prototype 5G Testbed. in *Proceedings of the First International Workshop on Experimentation and Measurements in 5G (EM-5G) 2018.* ACM, Heraklion, Crete, Greece, pp. 13-18, First International Workshop on Experimentation and Measurements in 5G, Heraklion, Greece, 4/12/18. https://doi.org/10.1145/3286680.3286683

Digital Object Identifier (DOI):
10.1145/3286680.3286683

Published In:
Proceedings of the First International Workshop on Experimentation and Measurements in 5G (EM-5G) 2018

Experience Building a Prototype 5G Testbed

Xenofon Foukas
The University of Edinburgh
x.foukas@ed.ac.uk

Fragkiskos Sardis
King's College London
fragkiskos.sardis@kcl.ac.uk

Fox Foster
The University of Edinburgh
fox@tardis.ed.ac.uk

Mahesh K. Marina
The University of Edinburgh
mahesh@ed.ac.uk

Maria A. Lema
King's College London
maria.lema_rosas@kcl.ac.uk

Mischa Dohler
King's College London
mischa.dohler@kcl.ac.uk

ABSTRACT

While experimental work in the context of 5G has gained significant traction over the past few years, the focus has mainly been on testing the features and capabilities of novel designs and architectures using very simple testbed setups. However, with the emergence of network slicing as a key feature of 5G, creating larger scale infrastructures capable of supporting virtualized end-to-end mobile network services is of paramount importance for experimentation. In this work, we describe our experience in building such a prototype cross-domain testbed targeting 5G use cases, by enabling multi-tenancy through the virtualization of the underlying infrastructure. The capabilities of the testbed are demonstrated through the use case of neutral-host indoor small-cell deployments, followed by a discussion on the challenges we faced while building the testbed, which open up new research opportunities in this space.

CCS CONCEPTS

• **Networks** → **Wireless access points, base stations and infrastructure**; **Network experimentation**; **Mobile networks**;

KEYWORDS

5G networks, prototype 5G testbed, 5G experimentation

ACM Reference Format:
Xenofon Foukas, Fragkiskos Sardis, Fox Foster, Mahesh K. Marina, Maria A. Lema, and Mischa Dohler. 2018. Experience Building a Prototype 5G Testbed. In *Workshop on Experimentation and Measurements in 5G (EM-5G'18), December 4, 2018, Heraklion, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3286680.3286683

1 INTRODUCTION

The research and standardization efforts on the fifth generation of mobile networks (5G) have been in full swing during the last few years. In contrast to the previous mobile network generations, a significant difference in the way that 5G research is being conducted is the increasing use of prototype system implementations as a means for experimenting and evaluating research ideas, rather than relying strictly on simulations. This change stems from a number of factors, including the appearance and widespread adoption of programmable Software-Defined Radios (SDRs) and the softwarization of the mobile network functions through various open source projects like OpenAirInterface (OAI) [12] and srsLTE [8]. This has made the low-cost deployment of mobile networks over commodity hardware a reality, allowing interested parties outside the telecommunications industry, like academics, to enter into this research space and to experiment with novel ideas, significantly accelerating innovation.

Until now, most research works in the 5G space that rely on prototype system implementations have focused on individual parts of the mobile network architecture (e.g. the RAN [5, 6] or the mobile core [11, 21]). Such systems are usually evaluated using simple small scale deployments comprised of a handful of commodity PCs. However, more recently there has been an increasing research interest towards the realization of more complex mobile network deployments that can support end-to-end multi-tenancy or *network slicing* in 5G parlance to study scenarios with multiple diverse services.

The key concept behind network slicing is the capability of virtualizing the underlying infrastructure and of creating logical networks by deploying and appropriately chaining Virtual Network Functions (VNFs), following Software-Defined Networking (SDN) and Network Function Vitualization (NFV) principles. However, implementing prototype systems that focus on the concept of end-to-end network slicing can be a challenging task, since any testbed realization needs to take into consideration a number of factors, including the deployment and configuration of the underlying infrastructure so that it can support virtualization and provide certain performance guarantees, the appropriate placement of the relevant VNFs so that they can meet certain service requirements etc. Therefore, this added layer of complexity can be a hindrance for those interested in realizing prototype systems focusing on end-to-end network slicing despite the availability of cheap hardware and open source software solutions.

With the above in mind and to enable our own research in the domain of 5G and network slicing, in this paper we provide a detailed documentation of our experience from building a prototype 5G testbed (Section 3). The testbed provides cross-domain capabilities, spanning the domains of the University of Edinburgh (UoE) and of King's College London (KCL), and enables the deployment of end-to-end network slices using commodity hardware and virtualized mobile network functions. The testbed capability for realizing 5G use cases is demonstrated through the emerging paradigm of neutral-host indoor small-cell deployments [1] (Section 4). This paper is meant to be not just a cookbook for those interested in realizing similar

EM-5G'18, December 4, 2018, Heraklion, Greece

ACM ISBN 978-1-4503-6083-8/18/12…$15.00
https://doi.org/10.1145/3286680.3286683

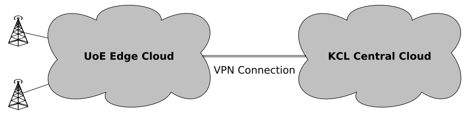


Figure 1: High-level overview of 5G testbed deployment

deployments, but also to highlight the challenges we identified during the testbed building process (Section 5), which create further opportunities for research in this space.

2 RELATED WORK

As mentioned in Section 1, the key enablers of prototyping and experimentation based research in the context of mobile networks, especially for the academic community, have been projects providing concrete implementations of mobile network functions following the 3GPP standards, like OAI [12] and srsLTE [8]. A number of 5G-oriented works have appeared over the past few years using those basic implementations as building blocks for realizing novel system designs, focusing on various aspects of the mobile network on the RAN (e.g. [5, 6]) or the mobile core (e.g. [11, 21]). More recently, this focus has expanded on creating frameworks that enable the deployment and management of end-to-end network slices (e.g. [7, 13]). However, in these works the main focus has been on how to introduce new functionality to the relevant mobile network functions or on the way that these functions can be managed and not providing a detailed blueprint on the experience, requirements and challenges of building a larger scale testbed. Our work reported here has a somewhat complementary focus, putting the spotlight on these aspects instead.

It should be noted that a number of projects focusing on testbeds for wireless experimentation already exist, with some noteable examples being PhantomNet [3], NITOS [9] and R2lab[17]. Such testbeds have helped accelerate mobile research, by allowing researchers to test their implemented systems in realistic settings and to share their work with others. Even more recently, there has been an increasing interest in going a step beyond and creating city-scale testbeds for 5G experimentation, with some characteristic examples being the POWDER [20] and COSMOS [19] testbeds developed as part of the PAWR project [14] in the US. However, despite the benefits that such projects offer, they do not allow researchers to access the low level details of the underlying infrastructure, something that can be important in order to create more holistic multi-layered solutions or to gain useful insights about the observed results of a system deployed over the infrastructure. In such cases, relying on custom solutions, like a private cloud deployment can be a very appealing approach, which is one of the main motivations behind this work.

3 PROTOTYPE 5G TESTBED

Here we describe the design of our prototype 5G testbed. As illustrated in Fig. 1, the testbed is cross-domain and is composed of two cloud deployments; one located at the Informatics Forum building of UoE in Edinburgh and one located at the Strand Campus of KCL in London. The UoE cloud was developed from scratch and is meant to act as an edge cloud, providing radio access and mobile edge computing capabilities. On the other hand, the KCL cloud was developed as part of the 5GUK project [23] and embodies the centralized location where part or all of the mobile core functions of a tenant could reside.

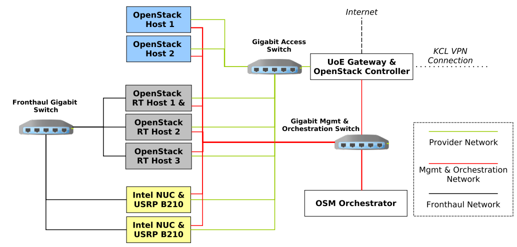


Figure 2: UoE edge cloud layout

The cross-domain communication of the two cloud deployments is enabled through a VPN connection over a JANET[1] network link.

3.1 Infrastructure configuration

UoE Edge Cloud – The UoE edge cloud design is illustrated in Fig. 2. It is based on OpenStack and is composed of 5 compute nodes. Two of the nodes (24-core Intel Xeon Silver 4116 @ 2.10GHz, 32GB RAM) are used for standard VNF hosting without strict execution or latency requirements (e.g. for mobile core functions) and three nodes (10-core Intel Xeon Silver 4114 @ 2.20GHz, 16GB RAM) target VNFs with real-time (RT) constraints (e.g. RAN-related VNFs). A number of optimizations have been applied to the RT-hosts to enable real-time performance, including disabled CPU C-states/frequency-scaling, the use of a low-latency Linux kernel, isolated CPUs for the operation of the hosts' OS, and for OpenStack and the configuration of OpenStack to enforce pinning of the deployed VNFs in dedicated rather than shared CPU cores. The grouping of the compute nodes to RT and non-RT is enabled through the host-aggregates mechanism of OpenStack, by assigning an RT or non-RT meta-data tag to each compute node. Through the tag mechanism, each VNF can indicate whether it has RT requirements, so that the OpenStack scheduler can assign it to the appropriate compute node.

For the radio front-end, small-factor Intel NUC PCs (4-core Intel i7 @ 3.2GHz, 8GB RAM) are connected to B210 USRP SDRs (two units). These nodes are not part of the OpenStack cloud, but instead are dedicated machines that allow the deployment of Physical Network Functions (PNFs), which can be shared among all tenants of the infrastructure in different configurations (e.g. a monolithic eNB or the lower layers of a C-RAN functional split) as elaborated later.

All of the nodes in the UoE deployment communicate via two separate dedicated networks using Gigabit Ethernet switches; one network is used for the remote management of the hosts (ssh, OpenStack API, VNF orchestration, etc) and the other acts as a provider network that allows the intercommunication of the PNFs and the VNFs (control and data plane traffic) as well as their communication with the Internet and the KCL testbed via a gateway node, also acting as the OpenStack controller. An additional node is also connected to the management network, providing orchestration capabilities across both the UoE and KCL domains, using ETSI OSM [4] and some additional custom orchestration scripts, as will be described later. Furthermore, a third dedicated fronthaul network is provided for the communication of the RT-hosts and the small-factor PCs using a Gigabit Ethernet connection. This fronthaul network is reserved for any type of network communication that requires low-latency

[1] https://www.jisc.ac.uk/janet

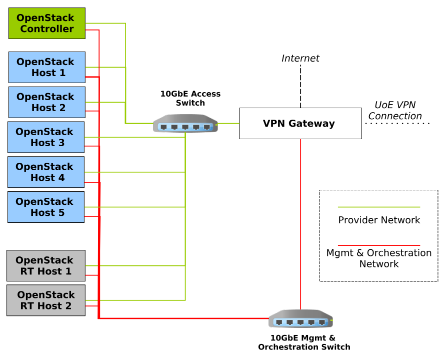


Figure 3: KCL cloud layout

guarantees that should not be mixed with any other type of traffic generated by the nodes. Examples of traffic with such constraints include the traffic exchanged between the higher and lower layers of the RAN protocols in configurations with C-RAN functional splits or the traffic exchanged in SD-RAN settings, where time-critical RAN control functions like MAC scheduling are decoupled from the data plane and are running in separate physical nodes.

The UoE cloud also features two Nvidia 1080 Ti GPUs to support edge-related machine learning tasks [18] (e.g. smart monitoring, interference management, base station coordination, etc). The first GPU is on one of the RT-hosts, so that it can be directly accessed by VNFs via PCI passthrough. Similarly to real-time VNFs, a VNF requiring a GPU for a machine learning task can simply be tagged appropriately, so that the OpenStack scheduler can deploy it to the aforementioned compute node (assuming the GPU availability). The second GPU is mounted on the orchestration node, so that it can be used for tasks related to the management and orchestration of the infrastructure.

KCL Central Cloud – The design of the KCL cloud illustrated in Fig. 3 is similar to that of the UoE site. It is also built on OpenStack and it is composed of 7 hosts used as compute nodes and 1 node used as the OpenStack controller. As in the case of the UoE cloud, the host-aggregates feature of OpenStack is also used in KCL, with five of the compute nodes (dual socket, 22-core Intel Xeon E5-2699A v4 @ 2.4GHz, 128GB RAM) used for VNF hosting without strict execution and latency requirements, while the other two hosts are optimized to offer real-time performance for hosted VNFs.

In terms of networking, the KCL site uses 10GbE links, with dedicated links for functions such as storage, API and communication of VMs through provider and overlay networks. This design isolates Openstack operations from VNF operations on the network, thus giving more consistent results during testing. It is also possible to perform live VNF migrations between the OpenStack compute hosts for maintenance purposes or to enable edge-caching and VNF localisation based on network congestion or other parameters. Moreover, the two RT-hosts feature dedicated network interfaces for SRIOV, making it possible to attach VNFs directly to a network interface, bypassing OpenStack networking. This is used for VNFs that require ultra-low latency or very high-bandwidth that might adversely affect other VNFs if they would share the same network interface. The access of all the nodes to the Internet is provided through a separate gateway, which is also used as a VPN server, enabling the communication between the KCL and the UoE site.

Table 1: VNF/PNF capabilities and deployment options

Name	Type	Location	Capabilities
Orion Controller	VNF	UoE RT-hosts	Slice MAC scheduling, monitoring etc.
Orion Hypervisor /BBU	PNF, VNF	UoE RT-hosts, UoE Intel NUCs	Virtualization of radio resources and data plane state, monolithic eNodeB, higher layers of functional split
Orion RRU	PNF	UoE Intel NUCs	Lower layers of functional split
srsLTE	PNF	UoE Intel NUCs	Monolithic conventional eNB
MME	VNF	UoE non-RT hosts, KCL non-RT hosts	Control plane operations of LTE
SP-GW	VNF	UoE non-RT hosts, KCL non-RT hosts	Mobile core user-plane traffic
HSS	VNF	UoE non-RT hosts, KCL non-RT hosts	Database for user authentication

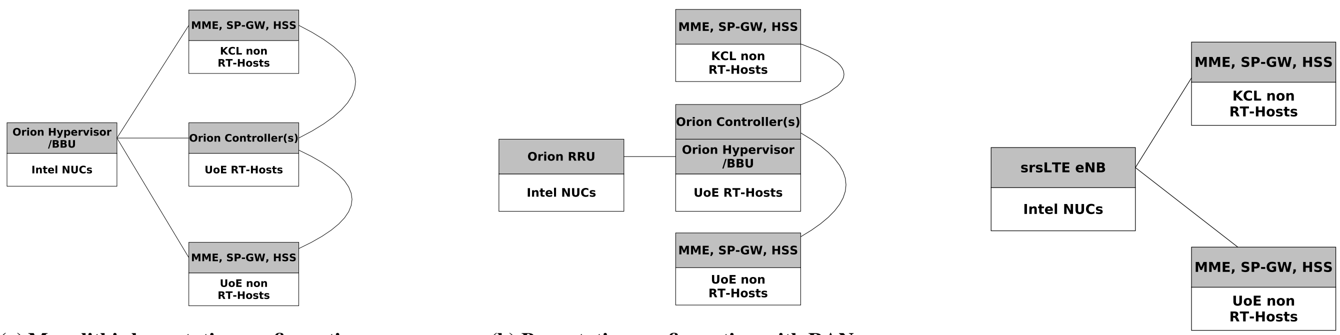
3.2 Supported Network Functions

To allow the sharing of the RAN among tenants (both hardware and spectrum), we leveraged the Orion RAN slicing system [6]. Orion provides functionally isolated virtual control planes (RAN controllers) for network slices and reveals virtualized radio resources to them through a Hypervisor component, ensuring both functional and performance isolation. Based on Orion's design, each tenant can either take full control of its slice by being assigned its own RAN controller, which can be fully configured in terms of the RAN control operations from the MAC layer and above (e.g. MAC scheduling) or multiple tenants can share the same RAN controller (e.g. the neutral-host's controller) in a RAN sharing approach, with the neutral-host being responsible of managing and sharing the radio resources among the tenants. These two approaches can co-exist over the same deployment, i.e. some tenants having their own independent RAN controllers for their slice, while others sharing a common controller. Orion is in turn built on top of the open source OpenAirInterface (OAI) LTE platform [12], which apart from operating as a monolithic eNodeB has built-in C-RAN support, offering three functional splits: lower-PHY, higher-PHY and MAC. Although in principle any of these functional splits can be employed, the Orion implementation is only compatible with the first two. The testbed also provides support for monolithic eNBs without slicing capabilities through the use of srsLTE.

For the mobile core we employed openair-cn [2], which is the most complete open source EPC implementation available, allowing the deployment of the HSS, MME and SP-GW functions as separate processes over the same or over different physical or virtual machines. Another open source EPC alternative that is also supported by our testbed is srsEPC [22]. However, the core functions of srsEPC are bundled in a single binary and therefore there is no flexibility in the deployment options for the core, making this a less appealing solution.

Based on the aforementioned components, we created a number of PNFs and VNFs that can be deployed over the UoE and KCL infrastructure to enable various 5G use cases that rely on end-to-end multi-tenancy. The details regarding the capabilities and the deployment locations of the aforementioned functions are listed in Table 1 and are illustrated in Fig. 4.

Starting from the EPC, we created three separate non-real-time VNFs for the MME, SP-GW and HSS, respectively. The EPC of a tenant can be deployed using a set of those three VNF instances, which can be flexibly placed in the non real-time hosts of the infrastructure in different combinations (e.g. all EPC functions in the UoE or KCL infrastructure, MME and HSS VNFs in KCL and SP-GW VNF in UoE etc.). This flexibility in the placement options, allows



(a) Monolithic base station configuration with RAN slicing.

(b) Base station configuration with RAN slicing and functional split.

(c) Monolithic eNB without RAN slicing.

Figure 4: VNF/PNF placement options

us to emulate different use cases and types of deployments, as will be further explained through an example in Section 4, regarding the use case of a neutral-host indoor small-cell deployment.

On the RAN side, different types of deployments can be used, based on the base station configuration. In the simplest case, when a legacy end-to-end 4G network without RAN slicing needs to be deployed, the srsLTE eNB can be used as a PNF over the Intel NUCs. In the Orion case and when base stations are used as monolithic components, the Hypervisor of Orion and the eNB are deployed over the Intel NUCs of the UoE infrastructure as a PNF running directly on the physical machine. However, in case of a functional split, the lower part of the split (e.g. low-PHY) is deployed as a PNF over the Intel NUCs, while the higher-part of the split (e.g. higher-PHY etc.) plus the Orion Hypervisor are deployed as a VNF over the RT OpenStack compute nodes of the UoE cloud. In both cases, the Orion Hypervisor allows the sharing of the base station among tenants, with the Orion controllers of the slices being deployed as separate VNFs over the RT compute nodes of UoE. It should also be noted that both srsLTE and the lower part of the split of Orion could also be deployed as VNFs rather than PNFs over the RT compute nodes, but we did not opt for this solution, because we would be very limited in terms of the placement of the radio front-end of the cells (restricted to only the location of the RT-hosts). Instead, the small-factor PCs currently hosting the radio front-end PNFs are very flexible and can be easily moved into different locations depending on our project and the layout of the RAN it requires (e.g. placement of cells in different locations across the UoE Informatics Forum building).

3.3 Network Orchestration

For the NFV orchestration, as already mentioned, the testbed provides cross-domain orchestration capabilities, by using a combination of ETSI OSM and our own custom scripts (both hosted at the UoE site). In terms of the supported network functions, we created VNF descriptors for the subset of those listed in Table 1, which are actually expected to be used as VNFs (i.e. mobile core functions, Orion Hypervisor/BBU and Orion Controller). For the configuration of the available VNFs, we employed the Juju charms framework, which already provides (partial) integration with OSM. Moreover, we created a number of network service descriptors for the various deployments described in Fig. 4. For each slice that we want to create, its network service descriptor contains all the slice-specific VNFs (e.g. the mobile core functions plus an Orion controller). In the case of the Orion Hypervisor (when used as a VNF), given that it is a function that is shared among multiple slices, we created an independent network service descriptor to allow it to be managed separately from the slices that use it. This is very important, since the life-cycle of the Orion Hypervisor is very different from that of the slices (many slices can be created and destroyed during the life of the Hypervisor).

It should be noted that OSM does not currently allow the deployment of a single network service over multiple domains in an automated manner (e.g. a single service descriptor allowing the placement of the mobile core at KCL and the RAN functions at UoE). To overcome this issue and to support the placement of functions over different domains, we also created smaller network service descriptors providing only a subset of the service. This makes it possible for us to combine those smaller services and manually choose the domain in which to deploy each one of them (e.g. use OSM to deploy a mobile core network service at KCL and then a RAN network service at UoE and chain them together through custom scripts). In terms of the RAN slicing and the chaining of the VNFs to the PNFs, we created our own custom solution (a set of Python and bash scripts), given that OSM currently provides no inherent support for this.

Based on this configuration, the idea for creating a slice over the testbed is that OSM is used to deploy all the functions that are slice specific (e.g. the Orion controller and mobile core). The functions are then placed in the appropriate compute nodes (RT and non-RT) by the OpenStack schedulers of the corresponding domain and are automatically configured through the Juju charms. Once the functions are ready, our custom scripts are executed and instruct how to chain the functions to the PNFs and start the service (e.g. BBU, slice's Orion controller and mobile core or srsLTE eNB to mobile core).

4 NEUTRAL-HOST USE CASE

To highlight the capabilities of our testbed, we consider the use case of a *"neutral-host"* [1], which enables the cost-efficient and simplified deployment of indoor small-cell networks. The key idea is that a third party entity (the neutral-host) takes the responsibility of deploying and managing the small-cell infrastructure, which is shared by multiple operators for a fee. The term operator could refer to traditional operators, who wish to extend their coverage to the indoor space, non-traditional operators, who may come with innovative

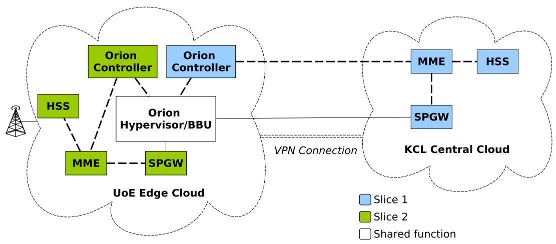


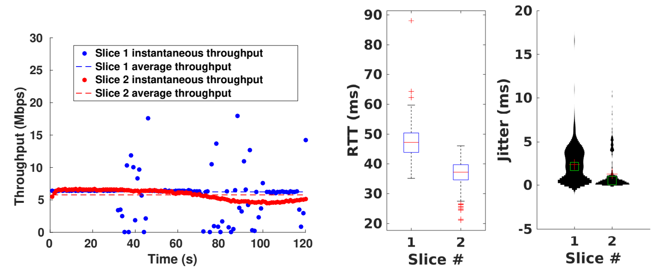
Figure 5: Neutral-host use case setup with 2 tenants

revenue models (e.g., free network access monetized by advertising and analytics) or even "local" operators that provide a private network for the inhabitants of the indoor space (e.g. in an enterprise setting). In all cases, the neutral-host becomes the only entity that needs to liaise with the site owner and address issues such as power and backhaul, relieving the operators of dealing with associated issues. In terms of the spectrum used by the neutral-host, different approaches have been proposed, including pooling licensed spectrum, using unlicensed spectrum or relying on shared spectrum (e.g. CBRS [10]).

As virtualization is a natural means for sharing the small-cell infrastructure, the neutral-host concept aligns well with the 5G vision of supporting a diverse array of services across different mobile network operators and verticals. From this perspective, the neutral-host paradigm can be seen as a use case of network slicing, where each operator is assigned its own indoor space virtual RAN. This RAN becomes part of the operator's end-to-end network solution, which could involve an existing core network in the form of specialized hardware or a cloud realization of the core (e.g., [11]) that can either be deployed locally at the indoor space or at a central location.

Here, we demonstrate how our prototype 5G testbed could be used to realize a neutral-host setting by having the UoE edge cloud act as the neutral-host deployment and the KCL central cloud act as a remote location that could be part of an operator's network backhaul. We consider a scenario involving two operators, each requiring a different configuration and chaining of network functions and we realize this setting through the creation of two slices over the testbed, as illustrated in Fig. 5. The first slice targets a traditional operator that wants to extend its coverage in the indoor space of UoE and uses its own mobile network core, while the second slice targets a private network deployment over the indoor space (e.g. at an enterprise setting). For the realization of the small-cell virtual RAN of the tenants we employed the monolithic eNB configuration presented in Section 3 and assigned individual Orion controllers for each of the two slices. Regarding the available spectrum, we used band 7 with 5MHz of bandwidth (25 resource blocks per subframe) that was equally and statically divided by the Orion Hypervisor between the two tenants (12 resource blocks per subframe for each slice). In terms of the mobile core, the RAN of the first slice was connected to EPC VNFs deployed over the KCL cloud, emulating the central mobile core deployment of the traditional operator, while the second slice was assigned mobile core VNFs that were co-located with the RAN functions at the UoE edge cloud (but in non RT-hosts).

Based on this setup and to validate our deployment, we connected two commercial UEs (one per slice) and measured the slices' performance in terms of throughput, latency and jitter as illustrated in Fig. 6. In the case of throughput (Fig. 6a), both slices achieve the same results on average, something that is expected, since both have been allocated the same amount of radio resources. However, when observing the instantaneous throughput, we can see that slice 1 presents fluctuations, while slice 2 is stable. Similarly, in terms of the round-trip time (RTT) and jitter (Fig. 6b), slice 1 performs worse than slice 2, having a higher latency and greater packet delay variations. The higher latency observed in slice 1 is mainly caused due to the placement of its mobile core over the KCL cloud, which means that traffic has to go through more hops to reach its destination. On the other hand, the throughput fluctuations and the increased jitter are mainly caused by the VPN link that connects the UoE and KCL clouds, leading to an unpredictable behavior.



(a) (b)

Figure 6: Throughput, latency and jitter measurement results for the deployed slices

Based on this example and the accompanying results, it becomes obvious that the configuration of the tenants' slices in the neutral-host context (and in any other slicing context for that matter) should take into consideration both the capabilities of the tenants (e.g. use of existing mobile core) and the effects of their slice deployment choices to the performance of their offered services. Testing such scenarios through a realistic larger scale testbed like the one presented in this work can provide very useful insights towards this direction.

5 CHALLENGES AND OPEN RESEARCH OPPORTUNITIES

This section briefly discusses the challenges that we faced, the lessons we learned and the open issues that create interesting opportunities for further research in this space.

Monitoring virtualized mobile networks. Given that one of the most important premises of 5G is the capability to support heterogeneous services with very diverse performance requirements, ensuring the QoS of the network slices is of paramount importance. However, one challenging aspect that became apparent during the process of building the testbed is the difficulty in monitoring the deployed services and identifying the root causes of a potential performance degradation (e.g. increased latency, throughput fluctuations etc).
The main reason for these challenges is the multi-layered architecture of next-generation mobile networks that includes the virtualized infrastructure (VNFs, compute nodes etc), the networking layer (switches, wired links, etc), the network protocols of the deployed services (e.g. 3GPP-based protocols) and the radio interfaces. Degraded performance could potentially originate from issues in any of the

aforementioned layers or even from their combination. The identification of such issues requires the capturing and analysis of data from multiple sources, which can be heterogeneous in terms of their format, their time granularity, etc. While emerging solutions in the context of NFV orchestration provide monitoring frameworks that allow the gathering of monitoring data from various sources (e.g. the Data Collection, Analytics and Events (DCAE) subsystem of ONAP [15]), there is still no mechanism that can intelligently combine the collected data to effectively identify the problems of the deployed services with a minimum overhead for the infrastructure. Therefore, creating advanced monitoring mechanisms for the effective and efficient monitoring of 5G deployments is of paramount importance.

Cross-domain challenges. An interesting aspect highlighted through the cross-domain setup of our testbed and the use case of Section 4 is the effect of the link connecting the remote domains to the performance of the deployed services in terms of jitter, packet loss etc. This can become a very important issue, considering that many 5G applications like VoIP calling and video streaming have low latency requirements and that cross-domain deployments are expected to become the norm for all large scale operators. Therefore, finding ways to improve performance in cross-domain settings is very significant. A promising approach to achieve this is Software-Defined Wide Area Networking (SD-WAN). In this, following the SDN philosophy, traditional branch routers are replaced with virtualization devices, in which application-aware flow policies can be applied through a centralized controller, improving the performance of low latency services. While existing orchestration and management solutions like ONAP and OSM provide mechanisms for SDN-based control of traffic (e.g. SDN-assist in OSM), further research is required on how these mechanisms can be applied in the WAN context.

Another important issue regarding the cross-domain orchestration of network services, which we realized while building the testbed, has to do with ensuring the QoS of the deployed functions across domains. The problem is that while currently the primitives for certain QoS operations can be the same across domains, their semantics and implementation can be very different. For example, when deploying a VNF with real-time requirements, the orchestrator has no visibility of the available compute nodes provided by each infrastructure and it is left up to the infrastructure provider to decide where exactly to place the real-time VNF and how to ensure its performance. Similarly, when deploying VNFs over infrastructures that provide legacy switches, traffic priorities indicated through services like DiffServ can be interpreted differently across domains. The result in both cases is that the performance of two identical services deployed across different domains could vary significantly. While there already exist some ways to partially resolve these problems, like the use of SDN instead of legacy switches and of mechanisms like Enhanced Platform Awareness (EPA) for interfacing the orchestration layer with the infrastructure, more work is still required towards this direction.

Domain-specific and intelligent orchestration. As already mentioned in Section 3, when building our infrastructure, even though we used ETSI OSM for the orchestration of our services, we also had to rely on our own custom orchestration solutions to provide a complete service. The main reason for this is that existing orchestration solutions like OSM provide generic frameworks for the management and orchestration of network functions, but are not yet capable of supporting the life-cycle management of domain-specific network functions, especially in the context of the 5G RAN (e.g. the Orion Hypervisor), which present a number of idiosyncrasies in terms of the way they are deployed and of their performance requirements [16].

6 CONCLUSIONS

In this paper, we have presented our experience building a prototype 5G testbed that enables multi-tenancy for the deployment of end-to-end network slices. A detailed testbed design was provided and its capabilities were demonstrated through the use case of neutral-host indoor small-cell deployments. We have also reported a number of challenges that we encountered during the testbed building process and outlined some interesting open research problems to address in the 5G architecture domain.

REFERENCES

[1] 5G Americas. 2016. Multi-operator and neutral host small cells: Drivers, architecture, planning and regulation. (Dec 2016).

[2] OpenAirInterface Alliance. 2018. Openair-CN. https://gitlab.eurecom.fr/oai/openair-cn. (2018). Accessed: 2018-09-11.

[3] Arijit Banerjee et al. 2015. Phantomnet: Research infrastructure for mobile networking, cloud computing and software-defined networking. *GetMobile: Mobile Computing and Communications* 19, 2 (2015), 28–33.

[4] OSM ETSI. 2016. Open Source MANO. *OSM home page* (2016).

[5] Xenofon Foukas et al. 2016. FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks. In *Proceedings of the 12th ACM CoNEXT*. ACM, 427–441.

[6] Xenofon Foukas et al. 2017. Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture. In *Proceedings of the 23rd ACM MobiCom*. ACM, 127–140.

[7] Ginés García et al. 2018. POSENS: a practical open-source solution for end-to-end network slicing. *IEEE Wireless Communications Magazine (Special Issue: 5G Testing and Field Trials)* (2018).

[8] Ismael Gomez-Miguelez et al. 2016. srsLTE: an open-source platform for LTE evolution and experimentation. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*. ACM, 25–32.

[9] Nikos Makris et al. 2015. Enabling open access to LTE network components; the NITOS testbed paradigm. In *In IEEE NetSoft*. IEEE, 1–6.

[10] Mobile Experts. 2017. CBRS: New Shared Spectrum Enables Flexible Indoor and Outdoor Mobile Solutions and New Business Models. (March 2017).

[11] Binh Nguyen et al. 2018. ECHO: A reliable distributed cellular core network for hyper-scale public clouds . In *Proceedings of the 24th ACM MobiCom*. ACM.

[12] Navid Nikaein et al. 2014. OpenAirInterface: A flexible platform for 5G research. *ACM SIGCOMM CCR* 44, 5 (2014), 33–38.

[13] Navid Nikaein et al. 2018. Mosaic5G: Agile and Flexible Service Platforms for 5G Research. *ACM SIGCOMM CCR* 48, 3 (2018).

[14] NSF. 2018. Platforms for Advanced Wireless Research (PAWR). https://www.advancedwireless.org/. (2018). Accessed: 2018-10-18.

[15] ONAP. 2018. Data Collection, Analytics and Events (DCAE). https://onap.readthedocs.io/en/beijing/submodules/dcaegen2.git/docs/sections/architecture.html. (2018). Accessed: 2018-09-07.

[16] ONAP. 2018. ONAP Support for 5G Wireless Network. https://wiki.onap.org/pages/viewpage.action?pageId=15992173. (2018). Accessed: 2018-09-07.

[17] Thierry Parmentelat et al. 2018. nepi-ng: an efficient experiment control tool in R2lab. In *ACM WiNTECH 2018-12th ACM International Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization*. 1–8.

[18] Michele Polese et al. 2018. Machine Learning at the Edge: A Data-Driven Architecture with Applications to 5G Cellular Networks. *arXiv preprint arXiv:1808.07647* (2018).

[19] Cosmos Project. 2018. COSMOS. http://cosmos-lab.org/. (2018). Accessed: 2018-10-18.

[20] POWDER Project. 2018. Platform for Open Wireless Data-driven Experimental Research (POWDER). https://powderwireless.net/. (2018). Accessed: 2018-10-18.

[21] Zafar Ayyub Qazi et al. 2017. A high performance packet core for next generation cellular networks. In *ACM SIGCOMM*. ACM, 348–361.

[22] srsEPC. 2018. srsEPC. https://github.com/srsLTE/srsLTE/tree/master/srsepc. (2018). Accessed: 2018-09-11.

[23] 5G UK. 2018. 5GUK Test Network. https://www.gov.uk/government/case-studies/5guk-test-network. (2018). Accessed: 2018-09-11.