Network Store: Exploring Slicing in Future 5G Networks

Navid Nikaein*, Eryk Schiller§, Romain Favraud*†, Kostas Katsalis♮,
Donatos Stavropoulos♮, Islam Alyafawi§, Zhongliang Zhao§, Torsten Braun§, and
Thanasis Korakis♮

*EURECOM, †DCNS
firstname.name@eurecom.fr

§University of Bern
name@iam.unibe.ch

♮University of Thessay
{kkatsalis,dostavro,korakis}@uth.gr

ABSTRACT

In this paper, we present a revolutionary vision of 5G networks, in which SDN programs wireless network functions, and where Mobile Network Operators (MNO), Enterprises, and Over-The-Top (OTT) third parties are provided with NFV-ready *Network Store*. The proposed *Network Store* serves as a digital distribution platform of programmable Virtualized Network Functions (VNFs) that enable 5G application use-cases. Currently existing application stores, such as Apple's App Store for iOS applications, Google's Play Store for Android, or Ubuntu's Software Center, deliver applications to user specific software platforms. Our vision is to provide a digital marketplace, gathering 5G enabling *Network Applications and Network Functions*, written to run on top of commodity cloud infrastructures, connected to remote radio heads (RRH). The 5G Network Store will be the same to the cloud as the application store is currently to a software platform.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design - Wireless Communication System.

Keywords

5G, Network Slicing, Network Store, NFV, SDN, Cloud, Network as a Service.

1. INTRODUCTION

Although there is no formal specification or description of what a 5G system will be, there is no denying that towards the new 5G ecosystem, we are not really focusing on just enhancing a technology like LTE. As all the research community suggests, 5G technologies aim to provide a holistic end-to-end infrastructure that will include all aspects of the network, while provisioning processing power on-demand. In this holistic approach, network abstraction and virtualization will serve as key enabling technologies for delivering consistent and enhanced end-user Quality of Experience in a highly heterogeneous environment.

Furthermore, traditional cellular networks architectures, simultaneously targeting multiple use cases, need to be revised to support many business application scenarios and different types of users. The reason is that the current structure of wireless networks does not fit the cloud concept in terms of scalability and reliability, due to the strong coupling between control and data planes, state-full design, and dependency to dedicated hardware. The exploitation of cloud technologies, Software-Define Networking (SDN) and Network-Function Virtualization (NFV), can provide the necessary tools to break-down the vertical system organization, into a set of horizontal micro-service network architectures. These can be used to combine relevant network functions together, assign the target performance parameters, and map them onto infrastructure resources.

In this paper, we present the design of a 5G-ready architecture and a NFV-based *Network Store* that can serve as a digital distribution platform for 5G application use-cases. The goal of the *Network Store* is to provide programmable pieces of code that on-the-fly reserve required resources, deploy and run the necessary software components, configure and program network elements according to the SDN and NFV paradigms [1], and provide the end-user with a 5G slice that perfectly matches the demands. The *Network Store* is a necessity as the 5G networking opens a multitude of applications. The Next Generation Mobile Network (NGMN) association's white paper [2] alone envisions 28 use-cases combined with multi-Radio Access Technologies (RATs) and various performance expectations, as also suggested in other proposals (e.g., Ericsson's 5G white-paper [3]).

We believe that our initial design with Network Functions can encourage 3rd party vendors to implement Network Applications in the same way as application stores motivate software development for mobile platforms, such as iOS and Android. Network Functions and Network Applications will allow the cloud to be re-programmed, built, and scaled as a function of particular business application scenarios, and give rise to the concept of network slices for 5G systems. From the industry perspective, we foresee that the *Network Store* business model will be a valuable asset for the Telecom industry in the near future.

This paper studies, designs, and prototypes a *Network Store* along with a network slicing architecture for 5G systems. A slice can be defined as a composition of adequately configured network functions, network applications, and underlying cloud infrastructures that are bundled together to meet the requirement of a specific use case or business model. The concept of network slices is not new and has been recently introduced by NGMN and adopted by the main Telecom manufacturers, however, its technology readiness level (TRL) remains at TRL2 that is only the technology concept is formulated. To this end, the proposed *Network Store*

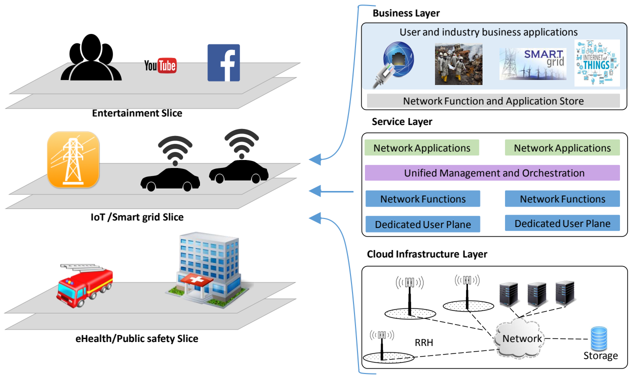

Figure 1: 5G Slice Approach

framework aims to achieve two complementary goals by providing:

1. **Service-oriented network architecture** that is programmable and extensible in terms of infrastructure, network services, and applications;
2. **Network slicing** to operate virtual networks on top of physical infrastructures, with virtual resource isolation and virtual network performance guarantees.

To meet these demands, a 5G system requires migrating from slow-moving proprietary and expensive hardware/software platforms (vertical approach) towards open software-defined functions on top of logical resources, leveraging the commodity hardware (horizontal approach). The proposed approach will enable the delivery of the network as a service and provide the flexibility needed to provision network resources on-demand and to tailor network slices to particular use cases. To achieve this goal, we leverage the Infrastructure as a Service (IaaS) cloud, which runs various atomic or composed services that are instantiated, chained, and bundle together. The network services operates over virtual and/or physical cloud resources under the control of the service manager and service orchestrator.

Note that, however, despite the multitude of proposed architectures, the number of real implementations capable of running 5G concepts is very limited due to a large number of unsolved scientific and technological challenges. For example, modern network architectures, such as CONCERT [4], FP7 MCN [5], only provide a certain level of mobile network virtualization that in most cases target the Mobile Network Operator's (MNO) Total Cost of Ownership problem.

The remainder of this paper is organized as follows. Section 2 describes the proposed system architecture. In Section 3, we discuss implementation details per layer. A proof-of-concept implementation of the LTEaaS slice and the experimental results on the requirements of the LTEaaS eNB application are presented in Section 4. Finally, Section 5 provides concluding remarks and possible future directions.

2. SLICING-ENABLED 5G ARCHITECTURE

A holistic top-down programmable network architecture is proposed. It allows us to dynamically combine and program (virtual) network functions together as the function of a particular application domain. In addition, it supports connectivity as a service, and delivers the resulting bundle of network services. The virtual network functions and their connectivity are selected by the unified control plane based on knowledge of target services, use-case requirements, and the supported technology features. Also various cloud techniques, which allow for the network abstraction and separation from the underlying physical infrastructure layer, are used and will be described in the following sections.

In the proposed architecture, every 5G deployment will be provided as a slice involving various software and hardware components (a similar layered architecture has been proposed by NGMN [2]). The adopted SDN/NFV-based virtualization scheme is presented in Figure 1 and investigated over the following multi-layered architecture:

- **The business layer** supports use-cases that can be provision thanks to the marketplace of virtual network functions (VNF) and applications (VNA). It creates the slice Manifest that encodes all the details required by the service orchestrator, service manger, and infrastructure manger logics to deploy the service bundle.
- **The service layer** supports the creation, and configuration of the service bundle according to the business use-case needs described in the slice manifest file. It has direct access to real-time network information required by the VNA, and provides network life-cycle service management.
- **The infrastructure layer** supports the real-time reconfigurable cloud ecosystem and virtualization for fast and ultrafast services (placement, deployment, provisioning).

A key innovation and technology enabler of the proposed system is the *Network Store*. The *Network Store* will offer hundreds of services that are ad-hoc available to every slice owner. Essentially, using the *Network Store*, one can deliver customized network slice templates tailored to particular use-cases. For example, an LTE network slice can result from several templates, which dynamically install, program, and configure all the LTE network-specific elements that correspond to specific business use-cases, for instance public-safety or low-latency mobile networks. Selecting broadband services from the *Network Store* triggers the lower layers as follows: the unified control plane, which resides on top of the infrastructure layer, triggers the creation of required LTE NFV elements (such as eNB, EPC, and Home Subscriber Service - HSS) and supports the proper configuration for each one with possibly advanced network features either through VNF (e.g., CoMP) or VNA (e.g., crowd distribution). Moreover, the unified control plane requests adequate computing, storage, and network resources from the infrastructure layer to maintain the target performance of network functions and applications. To ensure the performance isolation among the network functions and applications, each slice has a dedicated user plane.

3. DESIGN ELEMENT OF NETWORK SLICES

This section details the main design elements of the proposed cloud-native network slicing architecture as shown in Figure 2. It mainly focuses on the infrastructure and the service layers and their implementation details. For the business layer, with the devised approach in the infrastructure and service layers, a large set of applications, use cases, and business models identified in the NGMN white paper can be supported by the proposed solution.

3.1 The Infrastructure Layer

The infrastructure layer spans multiple sub-layers of traditional cloud computing architectures. It includes the bare metal layer and the virtualization layer that is responsible for the physical resource abstraction to facilitate infrastructure management. The following subsections provides further details about this layer.

3.1.1 *Programmable Computing, Network and Storage Hardware*

The lowest sub-layer of the infrastructure layer includes processing, storage, wired/wireless network elements, as well as, RF front-ends. The virtual resources (i.e., cloud regions) that are controlled by the infrastructure manager following ETSI NFV and/or SDN methodologies and mechanisms [1, 6, 7] reside on the upper second sub-layer. The virtualization layer, be it type 1—directly deployed on top of the hardware infrastructure or type 2—implemented on top of the Operating System, is a key enabler for IaaS clouds. There are a few virtualization techniques such as Operating System Level (e.g., Linux LXC) or Hardware Assisted Virtualization provided by Intel VT-x (e.g., XEN, Linux KVM) allowing for various execution requirements. Every slice's virtual infrastructure must be isolated from other slices to operate efficiently without violating the specific performance requirements (performance isolation). Consistency Availability and Partition tolerance (CAP) conjecture [8] states that it is impossible for any distributed system to provide consistency, availability, and partition tolerance at the same time. Thus, a trade-off must be made at the design time for different slice regions to control the partitioning and fulfil two out of the three guaranties. In order to maximize the infrastructure utilization, it is required to dynamically and freely relocate hardware resources depending on current and local needs, under the control of cloud operators (be it full or partial).

Virtual resources, virtual services, service bundle, and service chains, cloud as well as NFV and SDN technologies must be leveraged towards rapid building. NFV enables for extreme flexibility when it comes to the micro-service architecture, service chaining, and service life cycle management, hiding all the configuration complexities of underlay resources (physical and virtual) [9]. As an example, different network functions and/application components have special requirements in terms of processing time, delay, and jitter. They require to be adequately provisioned on appropriate platform configurations that can accommodate for the intended application requirements (e.g., on dynamically managed cloud regions with different virtualization schemes, i.e., LXC, KVM, XEN). NFV is therefore a glue between the infrastructure layer of the architecture and the service layer, where the latter provides NFV services to the former.

Regarding the network segment for every slice, the application layer of the architecture has to remain agnostic of the real network topology that is used by the virtual network provider, while the network slice operator must be able to determine the exact status of the network resources. This requires identifying (a) how network state and resources are exposed (and represented) to enable network application and services, (b) how the (virtual) network functions are aggregated and composed to construct the network, and (c) how the network is configured and instantiated for a particular application domain. Software-defined networking (SDN) is emerging as a natural solution for next generation cellular networks, as it enables further network function virtualization opportunities and network programmability. We plan to leverage SDN technologies and use SDN controllers together with programmable data plane technologies in order to support advanced programmability of the mobile networks (switching) over radio data-plane abstractions.

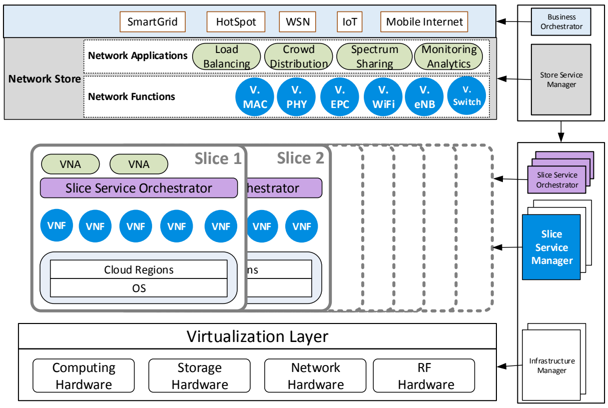

Figure 2: Vertical service bundle and chaining creation of network slices using network store.

One example of such a programmability is Canonical's JUJU Charm framework that allows us to model, build and scale each service bundle on any cloud. The JUJU services together with Metal As a Service (MAAS) are able to program physical severs on-demand allowing for programmability of cloud resources (computing, storage, and networking) and RF hardware in the remote radio head. Therefore, JUJU and MAAS can be seen as key enablers behind the so called programmable cloud, which adapts the infrastructure according to the requirements of slices instantiated (e.g., by deploying an appropriate number of adapted cloud servers with the LXC virtualization).

3.1.2 *Programmable RF Hardware*

Remote radio heads (RRH) have to be programmable to a large extent supporting various radio access networks (RAN). To allow for RAN programmability, the RRH can be equipped with the Field Programmable Radio Frequency (FPRF) technology (e.g., FPGA-based Myriad-RF) supporting various mobile broadband standards, such as 3GPP 4G, future 5G, and IEEE 802.11x, in different frequency bands, namely below 6GHz, from 6-30GHz (cmwave), and from 30-100GHz (mmwave). In addition, device drivers to the RF interface, as well as mechanisms for on-the-fly program desired RF technologies at the RRH, have to be developed. These will later enable NFV to determine an appropriate signal processing split between BBU and RRH [10,11]. By signal preprocessing, the RRH can offload baseband and protocol workers and optimize workload across local, edge, and centralized cloud.

3.1.3 *Radio Fronthaul Architecture*

Currently, the interface that interconnects BBUs with RRHs is based on a point-to-point synchronous link that can support multiplexing through daisy-chaining, e.g., Common Public Radio Interface (CPRI) or Open Base Station Architecture Initiative (OBSAI). This interface requires a high speed transport medium (e.g., fiber, passive WDM, cable, microwave) to carry the I/Q samples. It is called a fronthaul and its bandwidth requirements depend on the nature of the data transported (i.e., analog/digital, layer 1/2). Support of such a proprietary HW/SW interface is not a sustainable solution as a typical cloud uses a fast Ethernet-based infrastructure that consists of Ethernet network adapters, cabled/fiber-based links, switches, and routers. The commercial CPRI/OBSAI interface might only be maintained between the RRH and edge switches of the cloud. Transport of I/Q samples between RRHs and cloud servers (running VNFs of the radio network) for a software based signal processing, should be secured through efficient packet-based switching in support of statistical multiplexing for various fronthaul topologies (e.g., mesh). Also, a new functional split between BBU and RRH is emerging to reduce the required fronthaul capacity [12]. In addition, IEEE 1904.3 Standard specifies encapsulation and mapping of I/Q data over Ethernet (Radio Over Ethernet). While high-precision timing protocols over Ethernet exist, e.g., through IEEE 1588v2, other timing distribution approaches have to be also considered such as GPS, PHY layer clock, and SyncEth.

3.2 Service and Business Layer

Network abstraction and virtualization that occur in the infrastructure layer, reduce the information needed for network-wide control, since the control is made on a slice basis. The service layer supports network slice life-cycle management in terms of slice design, deployment, resource provisioning, runtime management, and disposal instructed by the business layer and agreed between providers. In particular, it bundles and chains the VNF under the control of slice service orchestrator, which provides direct access to real-time VNF information required by the VNA. It also implements slice services, which provide authentication, authorization, and slice control functions (instantiation, management, and disassembling of slices).

Business layer provides functionalities for both the infrastructure and business layers through a *Network Store*. The *Network Store* is comprised by two components: the *Network Function Store* and the *Network Application Store*. It maintains the network function store for VNF programming (such as virtual eNB, EPC, HSS of the LTE network) and dynamically provides slices with appropriate building blocks. Moreover, the business layer contains the network application store for VNA deployment that will be exposed to end users, such as the service providers, network operators and enterprises, for value added services. The interface to the end-to-end management and orchestration entity allows us to build dedicated network slices for an application, or to map the application onto the existing network slices.

3.2.1 The Network Function Store

This layer provides a *Function Store* for every slice. The marketplace serves as a high-availability storage and version control for the packaged network functions as well as a database containing the necessary meta-data describing each of the published function is provided. Such meta-data describes the required resources, reliability, and placement that a network function needs once it is instantiated. Furthermore, the marketplace offers a software development kit (SDK) and a store service manager allowing 3rd party network function developers to implement their network functions and to allow service providers to manage network functions. The network function store provides innovative and cutting-edge network functions and services for OTT players and enterprise business segments, based on the existing network infrastructure and capabilities.

3.2.2 Network Application Store

Network applications will make use of (quasi) real-time network information available at the service layer based on the set of SDN-based APIs to provide a feature-rich service. As an example, two network applications are presented [13]. Localization services, which rely on network-based measurements of active user devices providing solution for crowd distribution, mobile advertising, and tracking, in areas where GPS does not naturally work. Caching application, which relies on content popularity and prediction to minimize the time-to-start and round-trip time and enhance the performance for user traffic. The above applications might have multiple variants, e.g., different implementations of the same functionality, which have different performance or characteristics. Depending on the timing requirements, they can be placed either close to (particular) VNFs at the edge cloud or close to other VNA at the centralized cloud.

4. PROOF OF CONCEPT DEMONSTRATIONS

Following the reference architecture model presented in Section 2, a proof-of-concept prototype of the LTE as a Service (LTEaaS) slice is described. A service manager is providing supported services to the UI and request their creation from the OpenStack HEAT service orchestrator. The orchestrator is in charge of the end-to-end life-cycle management of the LTEaaS instance in a particular domain.

4.1 System Specification

We built a cloud setup able to run a LTE as a Service (LTEaaS) slice, which includes signal processing in the cloud (c.f. Figure 3). Based on our previous studies [11,14], we developed a portable OpenStack-based data center, with a specific configuration that satisfies the needs of deadline critical applications, such as RAN. The considered cloud server consists of commodity hardware, for typical processing, such as Intel processors, mother board, memory, network adapters (Intel i7 at 3.2 GHz, 8 GB RAM, 250 GB SSD). In this setup, we use NI/ETTUS USRP B210 as the RF frontend connected directly to the cloud server without the fronthaul network [15]. The cloud server is under Linux Ubuntu distribution with the low-latency Linux kernel version 3.17 that contains Linux deadline scheduler. This provides statistical guaranties on the LTE eNB to successfully meet its real-time requirements. On top of Linux, OpenStack is deployed to manage large pools of computing, storage, and networking resources. The OpenStack installation includes Heat whose mission is to create a human- and machine-accessible service, for managing the entire life-cycle of a virtual infrastructure and applications, within OpenStack. We describe the whole virtual infrastructure of the LTEaaS slice using

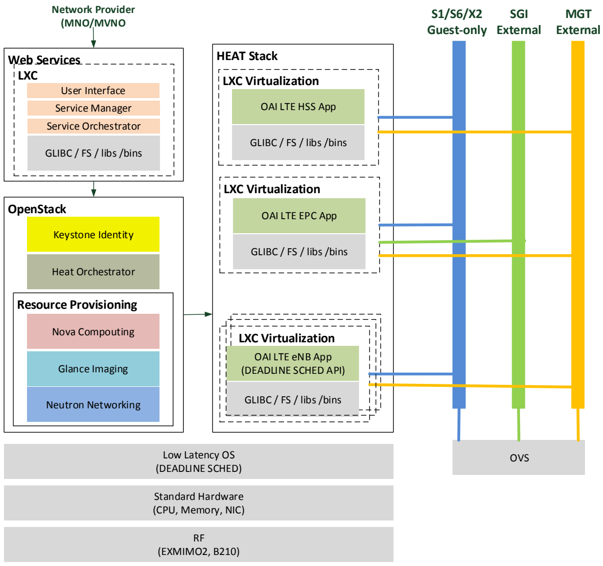

Figure 3: LTE as a Service Slice

a single Heat orchestration Template (HoT) that specifies the LTE network elements, their initial parameters and the required networking for this particular slice. Here, Heat manages the instantiation process of the VNFs, which are provided by the OAI, and their initialization according to the parameters provided by meta-data from the HoT. Currently, OAI implements the eNB, EPC, and HSS of the LTE standard. Due to the fact that the virtual eNB of OAI is a deadline critical application, we decided to configure OpenStack to use LXC as its virtualization technology to achieve a near bare metal performance. We later demonstrate that the maintenance of an appropriate cloud configuration is a key for baseband processing applications.

4.2 LTE eNB Application Requirements

In LTE-FDD, the total RX (UL) + TX (DL) processing should take less than 3 ms to comply with HARQ deadlines, which limits the number of concurrent threads/cores per eNB sub-frame to 3. By analyzing processing time for a 1 ms LTE sub-frame, 2 cores at 3 GHz are needed to handle the total BBU processing of an eNB (20 MHz channel bandwidth, SISO) [11, 14]. One processor core for the receiver, assuming 16-QAM on the uplink, and approximately 1 core for the transmitter processing with 64-QAM on the downlink, are required to meet the HARQ deadlines for a fully loaded system. Therefore, eNB is the most critical software component of the LTEaaS slice. Processing load is mainly dominated by uplink and increases with growing PRBs and MCSs. Furthermore, the ratio and variation of downlink processing load to that of uplink also grows with the increase of PRB and MCS. Moreover, average processing times are very close for all tested execution environments, but container approach (e.g., LXC and Docker) to virtualization have slightly lower variations than that of hypervisor approach (e.g., KVM), especially when PRB and MCS increase. These results provide insight on how computing resources have to be provisioned to meet the real-time requirements.

4.3 Appropriate provisioning of LTE eNB

In this section, we study performance of the time-critical eNB application within the LTEaaS slice architecture. Several experiments have been conducted based on the OAI platform, particularly relying on the implemented standard compliant LTE UE and eNB protocol stacks [15]. The LTEaaS shown in Figure 3 is deployed on the cloud center as described above. The real-time eNB application is operating in FDD band 7 SISO mode with 10 MHz channel bandwidth (50 PRBs). Downlink and uplink MCS are fixed to 26 and 16 in order to produce high processing load. Only 4 uplink sub-frames (SFs) are granted by the eNB, namely SF #0, 1, 2 and 3, allowing UL transmission to occur in SF # 4, 5, 6, 7.

To analyze the feasibility and performance of LTEaaS, two Linux OS schedulers are used, namely SCHED_FIFO or SCHED_DEADLINE (low-latency policy) in LXC. Furthermore, the container CPU resources are controlled by Linux cgroups and cpusets policies. To measure the achievable UL goodput of each case, a commercial LTE UE dongle, Bandrich C500, is connected to the instantiated network through the legacy over-the-air attachment procedure. Then traffic is generated from the UE to a local server connected to the EPC for a duration of 120 seconds using iperf.

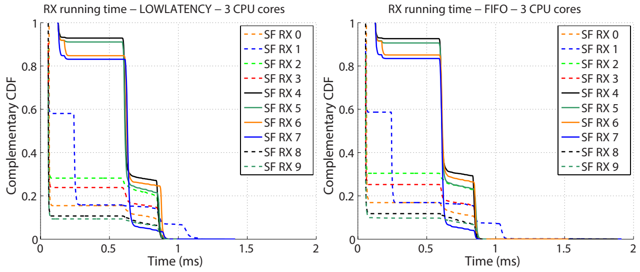

Figure 4: OAI LTE softmodem running on 3 CPU cores

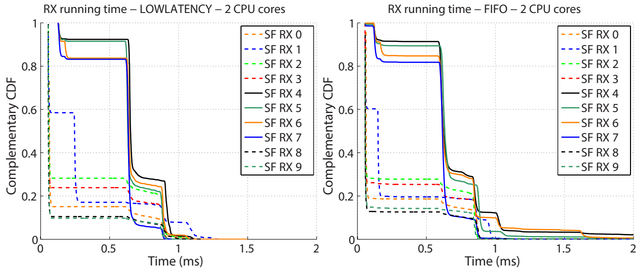

Figure 5: OAI LTE softmodem running on 2 CPU cores

It can be seen from Figure 4 that the behavior is very similar for both schedulers when 3 CPU cores are available and the OS is not loaded by other processes. They provide enough cores and CPU power to directly execute the required threads separately. We have no missed SF in either case, meaning that all threads were able to finish before their deadline requirements (RX 2 ms and TX 1 ms). Figure 5 shows the same results, but with two available CPU cores. We note that the results are the same as before when using low-latency scheduler. However, when using FIFO scheduler, the SF 2 ms processing time requirement is not always met (tail of the curves). This causes missed SFs during the transfer. While with the low-latency scheduler, the number of missed sub-frames remains 0, with FIFO scheduler it grows up to 708 during the 120 seconds transfer. This represents 0.6% of lost SFs caused by the scheduler not meeting the deadlines. The value might seem small, but as shown in Figure 6, it reduces the average goodput by more than 6%. These results confirm the importance of adequate hardware resources provisioning (programmable cloud concept) and scheduling to maintain the achievable performances. A similar behavior is to be expected on TX if the DL channel becomes loaded.

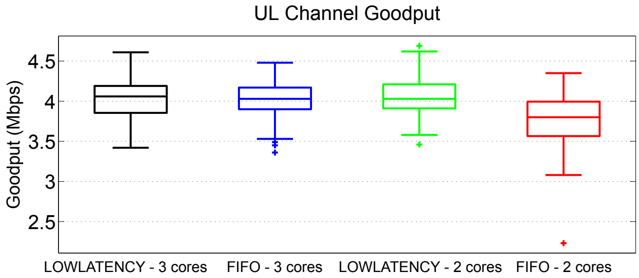

Figure 6: Impact of the execution environment on the LTE softmodem UL performance

5. CONCLUSIONS

In this work, we presented a slice-based 5G architecture that efficiently manages network slices. We combined many existing paradigms such as NFV, SDN, and cloud-computing, but the main contribution of this paper resides in the concept of "Network Store in a programmable cloud" allowing for dynamic network slicing. We show that programmable cloud is necessary to dynamically respond to the needs of various VNFs as appropriate component provisioning has a deep impact on the service performance. From the physical network infrastructure point of view, we also indicate that the Ethernet-based fronthaul network will significantly simplify and lower the price of the cloud network infrastructure for telecommunications. Our future plans include the extension of the *Network Store* with new virtual functions related to the wireless network and coupling with advanced SDN control. Additional activities regarding the separation of the cloud servers and the remote radio heads are also envisioned for the near future.

Acknowledgements

Research and development leading to these results has received funding from the European Framework Programme under FP7 grant agreement 318109 for the MCN project and H2020 grant agreement 671639 for the COHERENT project. The work from the University of Thessaly was supported by the ERC08-RECITAL project, co-financed by Greece and the European Union (European Social Fund) through the "Education and Lifelong Learning" Operational Program – NSRF 2007-2013.

6. REFERENCES

[1] ETSI. Network Functions Virtualisation (NFV), White paper. Tech. rep., 2014.

[2] A deliverable by the NGMN Alliance, NGMN 5G White Paper. https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf, Feb. 2015. Accessed: 2015-05-29.

[3] ERICSSON WHITE PAPER, 5G SYSTEMS. http://www.ericsson.com/res/docs/whitepapers/what-is-a-5g-system.pdf, Jan. 2015. Accessed: 2015-05-29.

[4] Liu, J., Zhao, T., Zhou, S., Cheng, Y., and Niu, Z. CONCERT: a cloud-based architecture for next-generation cellular systems. *IEEE Wireless Communications 21*, 6 (Dec. 2014), 14–22.

[5] Studer Ferreira, L., Pichon, D., Hatefi, A., Gomes, A., Dimitrova, D., Braun, T., Karagiannis, G., Karimzadeh, M., Branco, M., and Correia, L. An architecture to offer cloud-based radio access network as a service. In *Proceedings of European Conference on Networks and Communications (EuCNC)* (June 2014), pp. 1–5.

[6] EU FP7 – MCN Project, D2.5. Final Overall Architecture Definition. Tech. rep., 2015.

[7] Katsalis, K., Korakis, T., Landi, G., Bernini, G., Rofoee, B., Peng, S., Anastasopoulos, M., Tzanakaki, A., Christofi, D., Georgiades, M., Larsen, R., Ferrer Riera, J., Escalona, E., and Garcia-Espin, J. Content Project: Considerations Towards a Cloud-Based Internetworking Paradigm. In *Proceedings of Future Networks and Services, IEEE SDN4FNS* (Nov. 2013), pp. 1–7.

[8] Panda, A., Scott, C., Ghodsi, A., Koponen, T., and Shenker, S. CAP for Networks. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking* (2013), HotSDN.

[9] Wilder, B. *Cloud Architecture Patterns*. O'Reilly, 2012.

[10] Wubben, D., Rost, P., Bartelt, J., Lalam, M., Savin, V., Gorgoglione, M., Dekorsy, A., and Fettweis, G. Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible centralization through cloud-RAN. *Signal Processing Magazine, IEEE* (2014).

[11] Nikaein, N. Processing Radio Access Network Functions in the Cloud: Critical Issues and Modeling. In *Proceedings of ACM MOBICOM Workshop on Mobile Cloud Computing & Services* (2015).

[12] Rost, P., Bernardos, C., Domenico, A., Girolamo, M., Lalam, M., Maeder, A., Sabella, D., and Wubben, D. Cloud technologies for flexible 5G radio access networks. *Communications Magazine, IEEE* (2014).

[13] Patel, M., Joubert, J., Ramos, J. R., Sprecher, N., Abeta, S., and Neal, A. Mobile-Edge Computing. Tech. rep., ETSI, white paper, 2014.

[14] Alyafawi, I., Schiller, E., Braun, T., Dimitrova, D., Gomes, A., and Nikaein, N. Critical Issues of Centralized and Cloudified LTE-FDD Radio Access Networks. In *Proceedings of International Conference on Communications, IEEE ICC* (June 2015).

[15] Nikaein, N., Knopp, R., Gauthier, L., Schiller, E., Braun, T., Pichon, D., Bonnet, C., Kaltenberger, F., and Nussbaum, D. Demo – Closer to Cloud-RAN: RAN as a Service. *Proceedings of ACM MOBICOM Demonstrations* (2015).