Flow-based Network Slicing: Mapping the Future Mobile Radio Access Networks

Estefanía Coronado and Roberto Riggio
Wireless and Networked Systems (WiN)
FBK CREATE-NET, Trento, Italy
Email: {e.coronado, rriggio}@fbk.eu

***Abstract*—Nowadays mobile networks are asked to support different applications and services characterized by very specific Quality of Service (QoS) requirements. With this aim in mind, deploying network slices with particular resource allocation policies on a per-service basis becomes extremely relevant. In this regard, we introduce a solution able to dynamically partition the underlying physical infrastructure of a mobile radio access network into multiple logical slices with distinctive service-level agreements. We leverage Software-Defined Networking principles to provide fine-grained flow identification and sophisticated QoS management policies on a generic architecture supporting 4G and 5G networks with the objective of mapping the path towards the future mobile networks. The experimental evaluation of the deployed prototype on a real-world testbed has demonstrated the slicing capabilities of the system while ensuring full performance and functional isolation. We release the entire implementation under a permissive APACHE 2.0 license for academic use.**

***Index Terms*—mobile networks, LTE, 4G, 5G, network slicing, SDN, network virtualization, programmable architectures.**

I. Introduction

While until now mobile networks have been asked to support conventional Mobile Broadband (MBB) services, in the near-term outlook a single physical infrastructure must attend the demands of diverse users and applications. The telecommunications industry is pushing towards scalable and vertical-tailored systems, which calls for programmable architectures capable of dynamically instantiating, configuring and modifying specific network disciplines on a per-service basis. Software-Defined Networking (SDN) has been positioned as one of the game-changing solutions for transforming the current and future networking landscape. In fact, its ability to abstract the underneath physical resources, and to define customized services and network functions, has made it a natural fit for the forthcoming mobile ecosystem.

The Next Generation Mobile Networks (NGMN) Alliance defines a network slice as "*a set of network functions and resources, forming a complete instantiated logical network to meet certain network characteristics required by the service instance*" [1]. This statement can be translated into the ability of a network to support slice-specific requirements in terms of latency, throughput and availability for delivering a particular service [2]. This service-based architecture, strengthened by the programmability and virtualization principles that characterize future mobile networks such as 5G, is reflected in the recent 3GPP releases and technical reports [3]–[5].

So far mobile network technologies have been designed to provide specific functionalities mainly oriented to telephony. Nevertheless, fueled by the digital transformation, 5G and future mobile networks are expected to serve a variety of services, starting from the already established LTE networks. To cope with these new use cases, network slicing enables the creation of logical networks customized with precise network resources and isolation properties, optimized to fulfill specific requirements and to operate independently over a common infrastructure. As a clear sign of the growing importance of this view, slicing in the Radio Access Network (RAN) is envisioned as a cornerstone of the 5G New Radio (NR) in the last 3GPP specifications [6], where it is described as the definition and association of a set of configuration rules able to satisfy the requirements of the services allocated in a specific network portion.

In this context, this paper proposes a novel solution in the area of the flow-based slicing in mobile radio access networks, building on a generic architecture able support 4G and 5G networks. The slicing solution leverages SDN principles and network programmability to fill the gap from the current LTE technology toward 5G systems, and pursues the four objectives specified by 3GPP as design requirements in 5G RAN slicing: (i) *Resource Management*, radio resources must be shared between slices according to a certain scheduling policy; (ii) *Isolation*, functional and performance isolation must be guaranteed; (iii) *Quality of Service (QoS)*, service differentiation must be ensured; and (iv) *Transparency*, the resource configuration must be performed in a transparent manner for the users [5].

Taking into account the previous objectives, the contribution of this work is three-fold:

1) Dynamic partition of the underlying infrastructure in mobile radio access networks (in both 4G and 5G) into multiple vertical slices by means of the *Slice Policy* abstraction, which is able to define distinctive performance characteristics for each slice.
2) Flow-based slicing enabled by the *Traffic Rule* abstraction, which facilities the definition of customized radio resource management policies for a precise portion of the flowspace.

Research leading to these results has been supported by the European Union by the EU funded H2020 5G-PPP project under Grant Agreement 761592 (5G-ESSENCE Project).

978-1-7281-1856-7/19/$31.00 ©2019 IEEE

3) Prototype implementation of a flexible hypervisor capable of ensuring performance and functional isolation. Although our solution is 4G and 5G compliant, the prototype shown in the experimental evaluation is limited to 4G networks since no open-source 5G stacks are currently available.

To the best of our knowledge, this is the first work enabling programmatic definition of radio resource management policies on specific portions of the flowspace supporting both 4G and 5G radio access networks. The implementation is released under a permissive APACHE 2.0 license for academic use[1].

The remainder of this paper is organized as follows. Section II overviews the related work. In Sec. III we present the flow-based slicing solution in mobile networks. Section IV outlines the main design principles and implementation details. The evaluation methodology is described in Sec. V, while the experimental results are discussed in Sec. VI. Finally we draw our conclusions and present the future research in Sec. VII.

II. Related Work

Network slicing emerges in the NGMN's vision as a key architectural approach where the resources of a specific slice are optimized for a particular service from the core network to the RAN. This topic has become a critical issue in mobile networks and has raised significant interest in 5G-PPP projects [7] and research works [2], [8], [9]. However, it should be noted that network slicing is a concept not natively offered by 4G, and that belongs to 5G specifications and next generation networks. For this reason, a network slice model addressing both the 4G and 5G challenges acquires greater importance.

In the core network, slicing has experienced significant progress due to the efforts of 3GPP to reshape it towards a modular architecture. These contributions are reflected in DECOR (3GPP Release 13 [10]) and in its extension, evolved DECOR (eDECOR) included in Release 14 [11], two frameworks that enable service-dedicated core networks to address services and users with different characteristics. In the core network, the Evolved Packet Core (EPC) provides QoS policies control. However, its scalability may be limited due to the continuous synchronization between the components. This problem is analysed in PEPC [12] by refactoring the state and the access mode to the EPC. Higher scalability is also pursued by SCALE [13], which virtualizes the Mobility Management Entity (MME) element in the EPC to replicate the state of the devices across several Virtual Machines (VMs). Finally, higher level of customization and flexibility has also attracted substantial interest in slicing in the 5G core network [14]–[17].

Focusing on the RAN, network slicing is expected to vertically span the physical resources, and abstract them in a way that the network management procedures can operate in a technology-agnostic manner. Furthermore, the architecture must ensure inter-slice isolation so that the Service Level Agreement (SLA) of the operators is guaranteed. In this regard, both dedicated and shared models can be found in the literature. In the former, slices are isolated in terms of Control Plane (CP), User Plane (UP) and MAC scheduler. Nonetheless, physical resources are completely dedicated to a specific slice, which may involve resource waste. By contrast, in the latter models, network elements are shared between slices. To this end, a common scheduler is responsible for assigning and redistributing the physical resources according to priority levels, thus ensuring greater scalability and elasticity.

RAN slicing principles come in large part from the static concept of RAN sharing [18], [19]. Most of the works in this topic recall the broad outlines defined by 3GPP regarding resource sharing levels: Multi-Operator Core Networks (MOCN) and Multi-Operator RAN (MORAN) [20]. In MOCN the spectrum is shared among operators, while MORAN assigns dedicated frequency band to each of them. However, unlike its predecessor, network slicing goes one step further considering performance and functional isolation, as well as service differentiation. Nevertheless, some of these challenges are still a pending matter in the 5G vision, especially those related to flow-oriented service differentiation [8], [21], [22].

Software-Defined RAN (SD-RAN) can be an interesting approach to tackle the issues in RAN slicing since it facilitates flexible RAN management. In this regard, SoftRAN [23] proposes an architecture in which control functions are distributed, whereas the latency-sensitive ones are handled in the Base Station (BS). This work is extended in RadioVisor [24] to enable resource allocation based on the traffic demands of each slice. Nevertheless, slice isolation is not ensured. Conversely, FlexRAN [25] implements an SD-RAN platform and a south-bound API to assign the radio resources to the slices according to the requirements of the users.

Besides resource partitioning, RAN slicing must guarantee performance and functional isolation. In this respect, the system described in [26] introduces a solution that allows the definition of specific resource and management policies for each slice, thus providing the required isolation level. Likewise, in [27] a two-level MAC scheduler is presented to abstract and isolate Physical Resource Blocks (PRBs) between slices. However, resource customization is not considered. Conversely, Orion [28] enables the mapping of PRBs into virtual RBs to ensure control and slice isolation. Finally, in [29] the importance of both the isolation capabilities and the adaptation of the slice descriptors according to the network state is examined in LTE networks via simulation.

Despite the improvements made, the approaches presented above are not sufficient to fulfill the requirements of RAN slicing in future mobile networks. In fact, although the relevance of flow and service differentiation has been started to be investigated [29]–[32], there is not current research offering such a practical support in customized RAN slicing solutions.

III. Flow-Based Slicing System

Next-generation mobile networks seek to integrate network services with diverse requirements (i.e., in terms of throughput, reliability, security, etc.) within a common physical infrastructure. This entails a major re-engineering of the 3GPP net-

[1] Online resources available at: http://5g-empower.io/

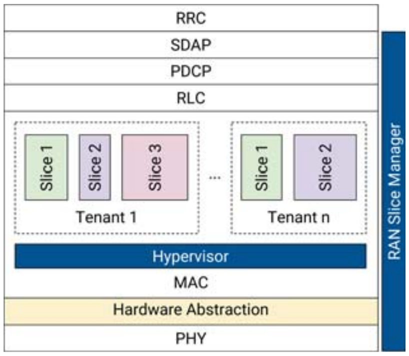


Fig. 1. High-level architecture of the flow-based RAN slicing system.

work architecture and of the associated protocols. Until now, network slices have been defined as logical isolated network portions based on predefined policies. Nevertheless, it should be possible for network operators to have a programmatic interface for network slicing. To offer finer granularity and to make slice management fully customizable, we focus on the RAN network to introduce the concept of flow-based RAN slicing. By *flow* we intend any traffic matching certain characteristics (the meaning of the term *"flow"* will be clarified in Sec. III-C). Examples of such flows can be very diverse, ranging from TCP or HTTP traffic, to traffic targeted to a specific IP address.

A. System Architecture

In SDN the network intelligence is shifted from the RAN to a logically-centralized software-defined controller. This enables: (i) flexible traffic differentiation; (ii) network slice composition and customization in a high-level fashion; and (iii) implementation of sophisticated management policies. In this work we rely on SDN to enable elastic slice instantiation.

The high-level architecture of the slicing system at the radio node is depicted in Fig. 1. As can be seen, the solution is based on a programmable hypervisor introduced at the MAC level of the mobile network stack. The RAN Slice Manager is in charge of allocating the physical radio resources, and of abstracting and exposing them to the software-defined controller. Conversely, the Hypervisor is responsible for managing the network slices and applying the policies provided by the controller. It is worth noticing that at the MAC layer, radio resources in the downlink are scheduled over both frequency and time domains while in the uplink traffic is scheduled over time. Given that our solution aims to support both domains, this work targets only downlink traffic, leaving uplink slicing as a future work, since different allocation strategies and hypervisor policies are required.

With our approach, multiple tenants can be hosted on the same physical infrastructure. Conversely, each tenant can instantiate a variable number of slices so that different SLAs can be fulfilled. Finally, the same slice can be shared between various User Equipments (UEs), and the same UE can make use of several slices simultaneously. Notice how details about pricing although important are out of the scope of this paper.

B. The Slice Policy Abstraction

The *Slice Policy* abstraction defines for each radio node the treatment to be given to the *flows* belonging to such a slice. The parameters defining a *Slice Policy* are the following:

- *Slice ID.* It is the unique slice identifier within a tenant.
- *Resources.* Indicates the radio resources assigned to the slice. It can be expressed (as a percentage or as a numerical value) in terms of Resource Block Groups (RBGs) per Transmission Time Interval (TTI).
- *UEs.* Sets the maximum number of UEs that can be connected to this slice.
- *UE scheduler.* Indicates how the UEs within the same slice shall be treated. By default, resources are scheduled between the UEs in a Round Robin fashion.

Each slice is identified across the entire network through the tuple `(PLMN ID, Slice ID)`, where the Public Land Mobile Network Identifier (PLMN ID) distinguishes the tenant in which the slice is deployed and the Slice ID identifies the slice within a given tenant. As it will be further explained throughout this section, the Slice ID can be any 16-bits numeric value of the extended OpenFlow header [33]. For simplicity, in this work we have chosen the Differentiated Services Code Point (DSCP) field as Slice ID. In this sense, the DSCP of the packets is modified depending on the slice in which the particular traffic flows must be allocated. In this manner, it is ensured that services with the same requirements are provided with the same network resources. The details about the *flow* tagging procedure are described in Sec. IV-D.

Notice that when a tenant is defined in the system, a default *Slice Policy* is created and associated to such a tenant. Therefore, an slice is always present in each tenant, which ensures that the traffic not matching the condition of any slice is properly handled.

C. The Traffic Rule Abstraction

OpenFlow switches are configured through OpenFlow rules, which allow network programmers to define how packets must be forwarded. These rules are composed of a *match*, which identifies a *flow*, and an *action*, which defines the operation to be executed on that specific *flow*. The *match* builds on the so-called OpenFlow header, which comprises a set of packet fields in the link, network, and transport layers [33]. Conversely, *actions* can be diverse, e.g., modify a field in a packet, forward to a specific port, etc. Openflow represents a good solution to deploy more flexible networks with fine-grained control on the packet forwarding.

In this work we define a *flow* as anything that can be described using an OpenFlow rule. Based on this, we then extend the OpenFlow rule concept to provide customized slicing in current and future mobile RANs. For this, we introduce a network abstraction called *Traffic Rule*. As shown in Fig. 2, similarly to the OpenFlow rule, a *Traffic Rule* allows network programmers to *tag* a precise portion of the flowspace with a particular DSCP code. Given that the DSCP of a *Slice Policy* identifies a slice within a tenant, any *Traffic Rule* defined

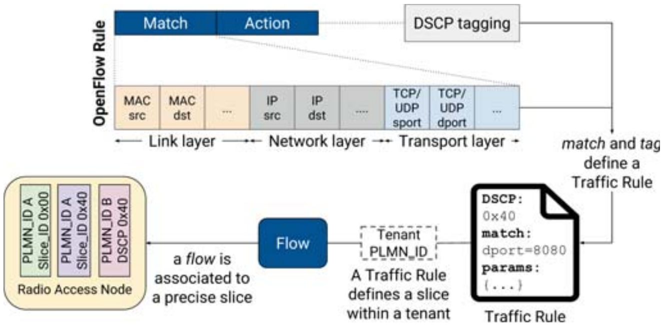


Fig. 2. Relationship between the *Slice Policy* and the *Traffic Rule* abstractions.

with the same *tag* is associated to such a slice. We remind the reader that the same design principles would apply for any other value selected as Slice ID.

It should be noted that multiple *Traffic Rules* can have the same *tag*. Therefore, the flow-oriented vision makes possible to map traffic with diverse characteristics (i.e., matching different *Traffic Rules*) to the same network slice, and hence, to be treated in the same manner. Moreover, Fig. 2 also shows how at the radio access nodes resources are scheduled just at slice level (and not at tenant level) since the tuple (PLMN-id, Slice ID), which unequivocally identifies a slice within a radio node, includes also the PLMN ID of the tenant. This design decision allows the system to behave in the same way when managing single or multiple slices, and to provide them with the required isolation level. As a consequence, the proposed solution can be applied to both 4G and 5G networks maintaining the compatibility with the 3GPP specifications.

IV. Prototype Implementation details

To showcase the flow-based slicing solution, we have deployed an experimental prototype based on a disaggregated RAN implementation comprising an SD-RAN controller and a radio access node. The functionality of the SD-RAN controller is provided by the 5G-EmPOWER platform [34], a Mobile Network Operating System (OS) supporting different Radio Access Technologies (RATs) such as Wi-Fi and LTE. In this work we extend the capabilities of 5G-EmPOWER with the abstractions introduced in Sec. III.

A. Control and Data Plane Implementation

The high-level view of the prototype is sketched in Fig. 3. The prototype is based on a open-source SD-RAN implementation encompassing the SD-RAN 5G-EmPOWER controller and an LTE eNB building on srsLTE [35]. It must be stressed that the design presented in this work can be applied to 4G and 5G networks. However, the prototype validation is limited to 4G since no open-source 5G stacks are currently available. Furthermore, the prototype integrates an open-source EPC (built on NextEPC [36]) connected to the radio access node through a Mobile Edge Host embedding an OpenFlow switch with a general purpose computing platform (the reason for this node will be clarified in the rest of this section). Finally, a backhaul controller (Ryu [37]) is also incorporated.

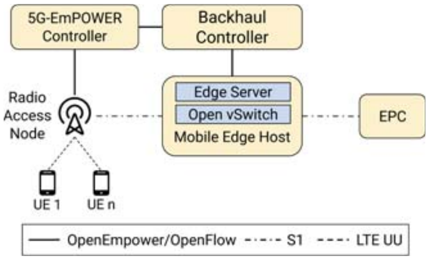


Fig. 3. Network topology used for the deployment.

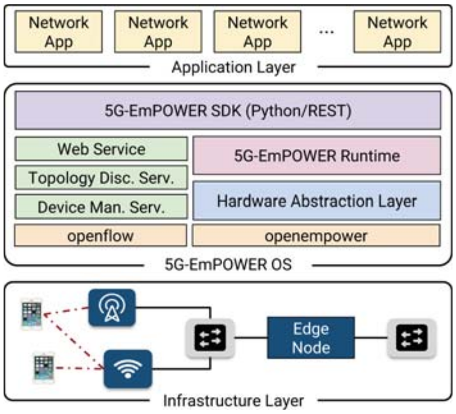


Fig. 4. The 5G-EmPOWER MEC-OS System Architecture.

The 5G-EmPOWER architecture is divided into infrastructure, control and application layers as shown in Fig. 4. The core of the system resides in the control layer enabled by the 5G-EmPOWER Operating System (OS). The OS comprises several modules that can be loaded/stopped at runtime in the form of plugins, and that can be combined for implementing complex network management applications. The main components of the 5G-EmPOWER OS are described below:

- *5G-EmPOWER Runtime*, which implements the network intelligence. Leveraging the hardware abstraction layer, it provides the instructions to the network elements in the infrastructure layer. This communication takes places through the southbound interface implemented by the *OpenEmpower* protocol using a persistent TCP connection. Further information about this protocol can be found online [38]. Conversely, the communication with the backhaul controller is performed through the openflow intent-based interface presented in [39].
- *Device Manager Service*, which tracks the managed radio access nodes including information such as the IP address and the offered capabilities (e.g., RAN slicing).
- *Topology Discovery Service*, which gathers data from UEs and radio access nodes to build a global network view exposed to the application layer to implement advanced management policies in the form of network Apps.
- *Web Service*, which provides the users with the interface to interact with the 5G-EmPOWER OS. The functionality is split into the SDK REST server and the front-end Graphical User Interface (GUI).

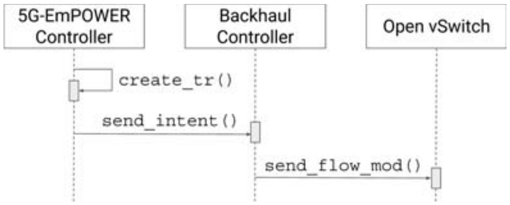


Fig. 5. Procedure to define a new *Traffic Rule*.

In the infrastructure layer, the 5G-EmPOWER Agent is embedded into the radio nodes to interact with the 5G-EmPOWER OS. This component comprises two parts: the Agent itself and the Wrapper. The Agent is responsible for serializing and transmitting the *OpenEmpower* messages to/from the OS, while the Wrapper defines a set of mandatory platform dependent operations specific to a network stack (e.g., 4G/5G) needed to be part of the 5G-EmPOWER managed network. These operations are invoked by the OS. Examples include radio access stratum reconfiguration, and UE reports [38].

B. Slice Definition

5G-EmPOWER provides a rich set of programming primitives through the Python-based SDK. Following this, the *Slice Policy* abstraction is exposed to the application layer through an object mapping properties to operations. This allows manipulating the slices in a tenant by accessing the `slices` property of a `Tenant` object. For example, defining a *Slice Policy* with Slice ID *0x40* is as simple as shown below.

```
>>> tenant.slices["0x40"] = \
>>>    Slice(dscp="0x40", descriptor=descriptor)
```

The descriptor defines the slice properties to be applied in the radio nodes managed by the 5G-EmPOWER Agent. However, in some cases different properties may be needed on certain nodes. For instance, the following descriptor assigns 5 RBGs to the slice in all the nodes, and schedules the UEs in a Round Robin fashion, while the node with address `"aa:bb:cc:dd:ee:ff"` is configured with 2 RBGs.

```
{
"properties": {"rbgs": 5, "sched_id": RR},
  "vbses": {
    "aa:bb:cc:dd:ee:ff": {
      "properties": {"rbgs": 2, "sched_id": RR},
    }
  }
}
```

This instruction triggers a message towards each node in that tenant which, in turn, instantiates a new slice with the properties specified in the descriptor.

C. Traffic Rule Definition

Following the previous principles, a *Traffic Rule* can be created in a `Tenant` object accessing the `traffic_rules` property to specify the treatment to be given to a certain *flow*.

```
>>> tenant.traffic_rules["dport=8080"] = \
>>>    TrafficRule(dscp="0x40", match="dport=8080")
```

Notice that this design decision allows the creation of a *Traffic Rule* to be independent from the *Slice Policy* one. In other words, a *Traffic Rule* with a certain *tag* can be created without slices and vice versa. Moreover, multiple *Traffic Rules* with different *match* properties can be redirected to the same slice (i.e., can have the same *tag*).

The previous instruction is performed by the `create_tr()` command shown in Fig. 5 and triggers a `send_intent()` message to the backhaul controller through an intent-based networking interface. This message has the following structure:

```
{
"src_dpid": "00:00:00:00:00:0A",
"src_port": 1,
"dst_dpid": "00:00:00:00:00:0A",
"dst_port": "LOCAL",
"match": {"dport": "8080"},
"tag": 0x40
}
```

The pair *(src dpid,src port)* identifies the backhaul port of the radio node, while the pair *(dst dpid, dst port)* identifies the virtual port to which the node is attached. This is usually the switch local port. The semantic of the message is that the in-bound traffic arriving on the backhaul port matching a rule must be tagged with the corresponding value. Moreover, a `send_flow_mod()` message is sent to the Open vSwitch to set the OpenFlow rule as described in the next section.

D. Tagging Procedure

The described design requires the modification of the DSCP code in the IP header. However, traffic sent from the EPC to the radio nodes is encapsulated in a GTP packet, which hampers the access to such a field. This packet structure is depicted in Fig. 6, where it can be seen that two IP headers are present. Once at the radio node, packets are decapsulated until reaching the MAC level and are allocated the radio resources. Notice that at this point they are just conventional IP packets. For that reason, the tagging *action* must be performed on the inner IP header of the encapsulated GTP packet.

The *tag* must be set before the packets reach the corresponding radio node. However, at this point, the inner IP header is not accessible in the GTP packet. For that reason, we have introduced a new virtual network element called Virtual GTP (vGTP) between the radio nodes and the EPC with the aim of removing the GTP header, and allowing the Open vSwitch to: (i) compare the header fields with the *match* in the rule entries; and (ii) to *tag* the packets with the corresponding DSCP code. The implementation of this element is based on Click [40], a framework for writing multi-purpose packet processing engines. As shown in Fig. 6, this element is composed of two Click sub-elements. When the vGTP receives a packet from the EPC, it is passed to the `GTPdecap` element to remove the GTP header and forward the IP packet to the Open vSwitch. Once the packet has been processed (i.e., accordingly *tagged*), it is passed to the VGTP element which, by using the `GTPEncap` element, re-encapsulates the packet and forwards it to the radio node.

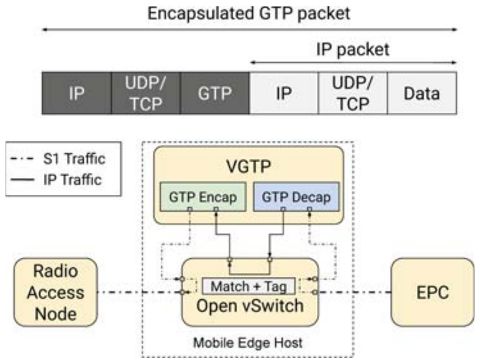


Fig. 6. Topology and procedure used for the *flow* matching and tagging.

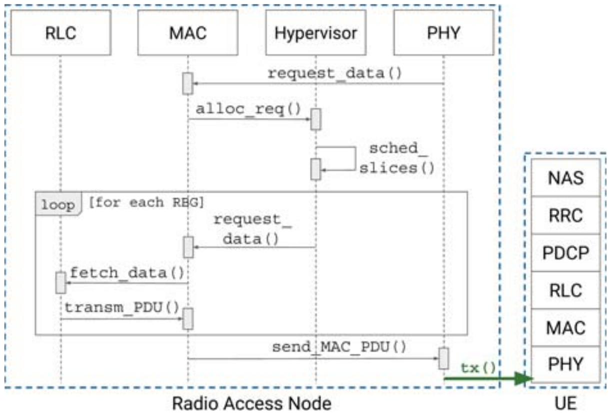


Fig. 7. Resource allocation procedure followed by the RAN slicing solution.

E. Resource Partitioning and Scheduling

The Hypervisor at the MAC level can be customized to follow diverse scheduling policies in order to fulfil the slice service requirements. Furthermore, it allows managing different types of radio resources (e.g., PRBs, airtime, etc.).

In the downlink, traffic coming from the GTP tunnels is pushed down through the stack in the radio node until the RLC layer. In parallel, the PHY layer fetches new data to the upper layers to deliver a new frame. This operation is sketched in the `request_data()` function in Fig. 7. Then, the MAC layer instructs the Hypervisor to assign the resources for the incoming flows (`alloc_req()`). Finally, based on the view of the active slices (comprising the number of UEs, the resource configuration, and the next packets to be transmitted) the Hypervisor must allocate the radio resources for the next TTI in the frequency domain (`sched_slices()`).

Notice that although the previous operation assigns the resources for the next TTI, the data still needs to be fetched from the RLC layer according to the resource configuration. As illustrated in Fig. 7, for each RBG allocated in the TTI, the resource manager must request this data to the upper layers. The MAC layer is responsible for communicating with the RLC through the `fetch_data()` function. After that, the whole data structure is encoded and transmitted in the air following the PHY specific logic.

TABLE I
TRAFFIC RULE CONFIGURATIONS FOR THE PERFORMANCE EVALUATION.

Traffic Rule	Tag	Match
TR1	0x01	"tp_dst=5008, nw_dst=10.20.30.2"
TR2	0x02	"tp_dst=5008, nw_dst=10.20.30.3"
TR3	0x01	"tp_dst=5008, nw_dst=10.20.30.4"
TR4	0x01	"tp_dst=5008, nw_dst=10.20.30.3"
TR5	0x03	"tp_dst=5008, nw_dst=10.20.30.4"

TABLE II
SCENARIOS PROPOSED FOR THE PERFORMANCE EVALUATION.

Test	Slice 0x01		Slice 0x02		Signal Quality	Traffic Rule
	RBGs	UE IDs	RBGs	UE IDs		
E1	7	1	6	2	Equal	1,2
E2	10	1	3	2	Equal	1,2
E3	7	1,3	6	2	Equal	1,2,3
E4	10	1,3	3	2	Equal	1,2,3
E5	7	1	6	2	Different	1,2
E6	7	1,3	6	2	Different	1,2,3
E7	7-10	1,3	6-3	2	Equal	1,2,3

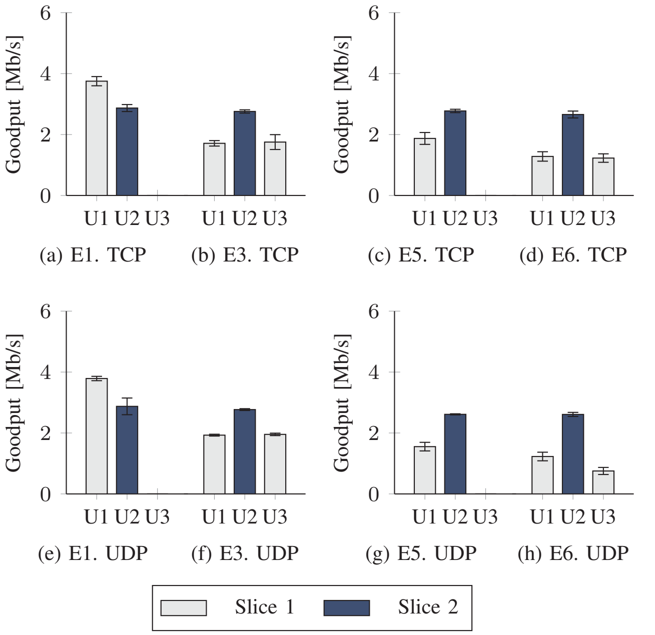
V. EVALUATION METHODOLOGY

The evaluation follows the setup depicted in Fig. 3 aiming to validate the 3GPP design principles of RAN slicing, namely flexible slice deployment, resource and performance isolation, service differentiation, and dynamic slice elasticity.

The setup comprises an LTE eNB connected to the EPC via a Mobile Edge Host running Open vSwitch. The EPC and the radio access nodes are deployed on Intel NUCs with an i7 Intel processor and 16 GB of RAM running Ubuntu 18.04.1. The 5G-EmPOWER and the Ryu controllers are placed on the same node running the EPC. The eNB has a capacity of 25 PRBs (grouped in 13 RBGs) that corresponds to a 5 MHz bandwidth. NextEPC [36] is used as EPC, while the LTE stack is based on srsLTE [35]. Finally, 3 UEs have been considered in the experiments by using Huawei P10 Plus smartphones.

The *Traffic Rules* presented in Table I have been defined to be used during the experiments, being them composed of a *match* based on the flow destination port and address. Then, Table II lists the scenarios assessed in the evaluation, including the information about 2 network slices (i.e., the resources allocated and the UE IDs connected), the signal quality, and the *Traffic Rules* used in each case to allocate the corresponding flows. Within each slice, UEs are scheduled in a Round Robin fashion. Notice that the signal quality referred to as equal or different. In the former, all the UEs perceive good and similar channel quality, while in the latter the UEs in one of the slices are under bad channel conditions.

For each experiment TCP and UDP traffic is generated by Iperf from the EPC towards the UEs. The measurements have a duration of 30s and are the average of 10 runs. The goodput at the UEs side and the overall goodput per slice have been used as Key Performance Indicators (KPIs).



(a) E1. TCP (b) E3. TCP (c) E5. TCP (d) E6. TCP

(e) E1. UDP (f) E3. UDP (g) E5. UDP (h) E6. UDP

Fig. 8. Goodput measured for two slices with the same radio resources for a variable number of UEs and signal quality.

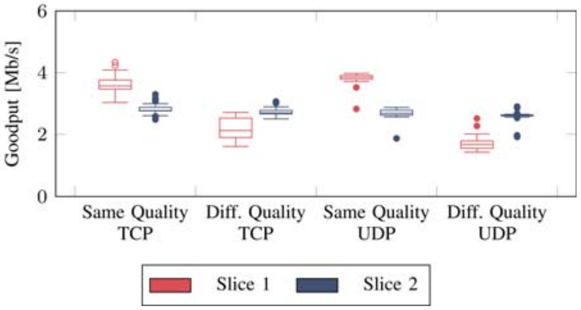
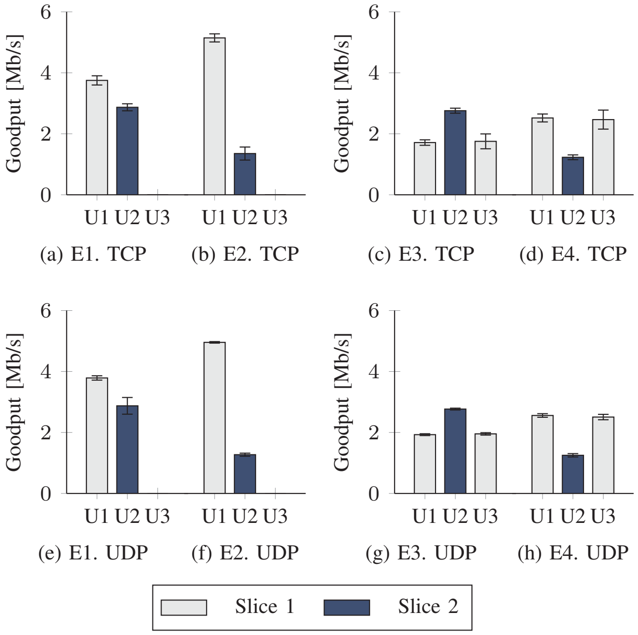


Fig. 9. Goodput comparison for two slices under different signal qualities: poor conditions for *Slice 1* and good conditions for *Slice 2*.



(a) E1. TCP (b) E2. TCP (c) E3. TCP (d) E4. TCP

(e) E1. UDP (f) E2. UDP (g) E3. UDP (h) E4. UDP

Fig. 10. Goodput comparison for a variable number of UEs and different radio resource allocation for each slice.

VI. Experimental Results

A. *Resource and Performance Isolation*

These experiments assess the isolation capabilities of our solution. To this end, the radio resources are equally distributed into two slices. However, it should be noted that a system bandwidth of 5 MHz includes 13 RBGs (i.e., 25 PRBs). As a consequence of this odd number, as shown in Table II, the slices are assigned 7 and 6 RBGs, respectively, which may lead to slight differences in the results shown above. However, other resource configurations are explored in Sec. VI-B. The slices isolation is examined in Fig. 8 from two angles: number of active UEs and signal quality of such UEs.

One the one hand, the Hypervisor guarantees that the number of users in one slice does not affect the performance of other slices. In experiment *E1*, depicted in Fig. 8a (TCP) and Fig. 8e (UDP), a single UE is connected to each slice, allowing them to fully use the resources in each slice. This is done by setting the *Traffic Rules TR1* and *TR2*. By contrast, in experiment *E3* in Fig. 8b (TCP) and Fig. 8f (UDP), an additional UE (U3) is attached to *Slice 1* by setting the *Traffic Rule TR3*, which allocates the flows towards this UE in such a slice. This change demonstrates that the connection of a new UE in *Slice 1* (and the consequent resource distribution between the UEs in the slice) does not alter *Slice 2*.

On the other hand, full slice isolation must ensure that misbehavior in one slice does not influence others. The presence of UEs with poor signal quality is an example of this situation. This effect is studied in experiments *E1* and *E3*, where the UEs experience good signal conditions, and in *E5* and *E6*, where the UEs in *Slice 1* have low signal quality. In Figs. 8c and 8d, and Figs. 8g and 8h, for TCP and UDP respectively, it is proved that these problems affect just such a slice, while the performance of the other slice is maintained regardless of the number of UEs and the traffic type. The same conclusions can be drawn from Fig. VI-A, where the overall performance of each slice for the described experiments is displayed.

B. *Functional Isolation and Resource Management*

The functional isolation and the ability of the system to meet specific SLAs is the aim of these experiments. In this respect, we examine the impact of varying the resource configuration policies. To do this, we modify the slice descriptors discussed in the previous section by allocating greater resources to *Slice 1* (10 RBGs) and the rest to *Slice 2* (3 RBGs). Figure 10 sketches such a comparison for different number of UEs.

In Fig. 10 experiments *E1* and *E2* show the network status when connecting 1 UE to each slice, whereas experiments *E3* and *E4* extend this scenario by adding a UE to *Slice 1*. The *Traffic Rules* used are the same described in the previous

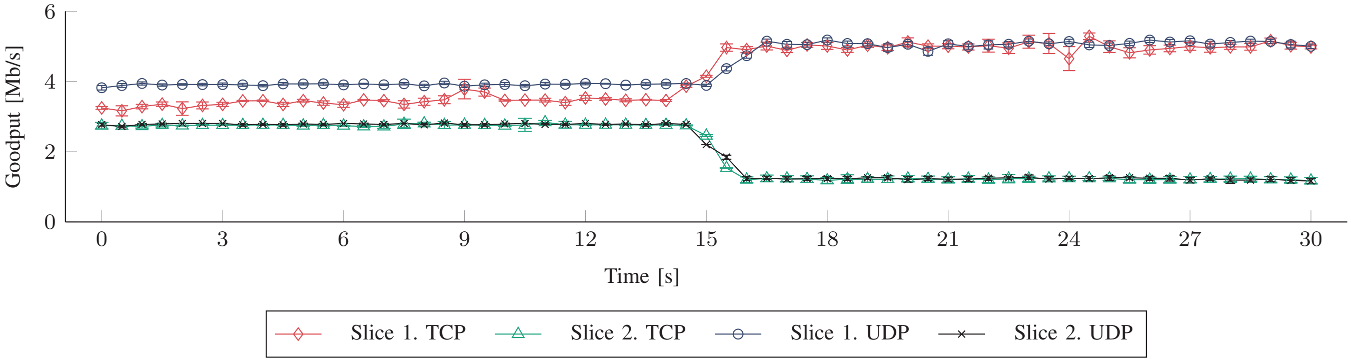


Fig. 11. Goodput evolution over time for two slices when dynamically varying the radio resource assignment (E7).

section (i.e., *TR1*, *TR2* and *TR3*). Notice that changes in the resource configuration are performed in the slice descriptors (from the SD-RAN controller with the *Slice Policy* abstraction) and do not involve modifications in the *Traffic Rules*. These results reveal that the goodput of the UEs is consistent with the resource distribution assigned to the slices in each scenario (e.g., Fig. 10a with respect to Fig. 10b). Moreover, it is worth highlighting that this capability remains regardless of the type of service (i.e., TCP or UDP) and the UEs in the network.

C. Flexible Resource Rescheduling

These experiments examine the capacity to modify the resource allocation in real-time. Taking as a reference the two slices defined before (2 UEs connected to *Slice 1* and one UE connected to *Slice 2*), in experiment *E7* shown in Fig. 11 we initially apply a configuration allocating 7 and 6 RBGs for *Slice 1* and *Slice 2*, respectively. To conduct these tests, the *Traffic Rules TR1*, *TR2* and *TR3* have been applied.

During $30s$, and every $0.5s$, we have measured the goodput of the UEs in each slice. After $15s$, the resource configuration is modified to provision *Slice 1* and *Slice 2* with 10 and 3 RBGs, respectively. From the results it can be concluded that the system is able to dynamically reallocate the radio resources in order to meet new SLAs in a completely transparent manner for the UEs, without interrupting any ongoing transmission and without hampering the network performance.

D. Slice Deployment

These experiments aim to demonstrate that the number of slices in the system does not determine the communications performance. In this regard, Fig. 12 sketches the CDF of the goodput achieved by the UEs when deploying 1, 2 and 3 slices. In the case of a single slice, the 3 UEs are connected to such a slice. The *Traffic Rules TR1*, *TR3* and *TR4* are applied for this purpose. Conversely, the scenario with 2 slices considers the same configuration of experiment *E3* (i.e., 2 UEs in *Slice 1* deployed with 7 RBGs, and 1 UE in *Slice 2* set up with 6 RBGs). Finally, when instantiating 3 slices, 1 UE is attached to each of them. To this end, the *Traffic Rules TR1*, *TR2* and *TR5* ensure the correct mapping of traffic flows into the slices.

As can be observed, the average goodput in the three cases remains unaffected by the creation of additional slices in the network, and just small differences are displayed in the results due to the different behaviour of TCP and UDP services. Therefore, it can be concluded that the solution does not induce additional complexity or performance drops when managing multiple slices, thus enabling high scalability.

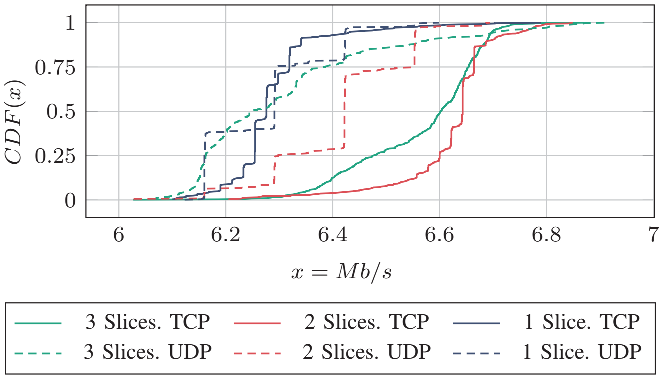


Fig. 12. CDF of the goodput when equally dividing the radio resources between a variable number of network slices.

E. Service differentiation and transparency

Although the destination ports and addresses of the flows have been used as *match* condition in the *Traffic Rules*, the examples just merely intend to prove the ability and the advantages of using flow-based slicing, since any other configuration would have led to the same results. As a matter of fact, any *match* statement can be set on the OpenFlow Header for service differentiation and/or to fulfil the necessities of services and operators required in each moment, thus allowing the partitioning of the flowspace in a fully customizable manner.

The dynamic resource reconfiguration is usually limited by the slicing strategy and the underlying technology. However, the slicing principles introduced in this work rely on the custom-built allocation of the RAN resources. Consequently, the system operations are completely transparent for the UEs, making it appropriate and fully compliant with the 4G and 5G mobile network architectures. In fact, if a UE simultaneously uses two applications categorized in two different slices (due to the *Traffic Rules* configuration), it would have the impression of being connected to a single slice.

VII. Conclusions

In this paper we have proposed a novel flow-based slicing solution able to support 4G and 5G networks, offering the required flexibility to meet the tight necessities of the future mobile radio access networks. To make this possible, we introduce a set of new network abstractions. On the one hand, the *Slice Policy* abstraction enables the creation of customized network slices with distinctive resource management policies. On the other hand, the *Traffic Rule* abstraction relies on OpenFlow principles to map a precise portion of the flowspace to a certain slice. The evaluation of the experimental prototype conducted on a real-world testbed demonstrates the effectiveness of the system and the isolation features.

As future work, we intend to explore various aspects. Firstly, we plan to provide uplink slicing capabilities in parallel with the current downlink solution. Secondly, we aim to extend the network abstractions to support diverse RATs (such as LTE and Wi-Fi) in the same slice. Finally, we aim to incorporate other schedulers to manage the UEs in each slice, as well as to stress the system with a higher number of users and slices.

References

[1] NGMN, "Description of Network Slicing Concept," 2016, Accessed on 16.05.2019. [Online]. Available: https://www.ngmn.org/fileadmin/user_upload/160113_Network_Slicing_v1_0.pdf

[2] M. Richart, J. Baliosian, J. Serrat, and J. L. Gorricho, "Resource Slicing in Virtual Wireless Networks: A Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, 2016.

[3] 3GPP, "TR 23.501. System Architecture for the 5G System," 2018.

[4] 3GPP, "TR 23.502. Procedures for the 5G System," 2018.

[5] 3GPP, "TR 38.801 V2.0.0. Study on New Radio Access Technology; Radio Access Architecture and Interfaces," 2017.

[6] 3GPP, "TR 38.133 V15.2.0. 5G NR. Requirements for support of radio resource management," 2018.

[7] 5GPPP Architecture Working Group, "View on 5G Architecture," 2017, Accessed on 16.05.2019. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2018/01/5G-PPP-5G-Architecture-White-Paper-Jan-2018-v2.0.pdf

[8] E. Pateromichelakis, J. Gebert, T. Mach, J. Belschner, W. Guo, N. P. Kuruvatti, V. Venkatasubramanian, and C. Kilinc, "Service-Tailored User-Plane Design Framework and Architecture Considerations in 5G Radio Access Networks," *IEEE Access*, vol. 5, pp. 17 089–17 105, 2017.

[9] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, and G. S. Paschos, "The algorithmic aspects of network slicing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 112–119, 2017.

[10] 3GPP, "TR 23.707. Architecture enhancements for dedicated core networks; Stage 2," 2015.

[11] 3GPP, "TR 23.711. Enhancements of Dedicated Core Networks selection mechanism (Release 14)," 2016.

[12] Z. A. Qazi, M. Walls, A. Panda, V. Sekar, S. Ratnasamy, and S. Shenker, "A High Performance Packet Core for Next Generation Cellular Networks," in *Proc. of ACM SIGCOMM*, New York, NY, USA, 2017.

[13] A. Banerjee, R. Mahindra, K. Sundaresan, S. Kasera, K. Van der Merwe, and S. Rangarajan, "Scaling the LTE Control-plane for Future Mobile Access," in *Proc. of the ACM CoNEXT*, Heidelberg, Germany, 2015.

[14] Z. A. Qazi, P. K. Penumarthi, V. Sekar, V. Gopalakrishnan, K. Joshi, and S. R. Das, "KLEIN: A Minimally Disruptive Design for an Elastic Cellular Core," in *Proc. of ACM SOSR*, Santa Clara, CA, USA, 2016.

[15] K. Mahmood, T. Mahmoodi, R. Trivisonno, A. Gavras, D. Trossen, and M. Liebsch, "On the integration of verticals through 5G control plane," in *Proc. of the IEEE EuCNC*, Oulu, Finland, 2017.

[16] Y. i. Choi and N. Park, "Slice architecture for 5G core network," in *Proc. of the IEEE ICUFN*, Milan, Italy, 2017.

[17] I. Afolabi, M. Bagaa, T. Taleb, and H. Flinck, "End-to-end network slicing enabled through network function virtualization," in *Proc. of the IEEE CSCN*, Helsinki, Finland, 2017.

[18] A. Khan, W. Kellerer, K. Kozu, and M. Yabusaki, "Network sharing in the next mobile network: TCO reduction, management flexibility, and operational independence," *IEEE Communications Magazine*, vol. 49, no. 10, pp. 134–142, 2011.

[19] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, 2016.

[20] 3GPP, "TR 23.251. Network sharing; Architecture and functional description," 2018.

[21] C. Liang and F. R. Yu, "Wireless Network Virtualization: A Survey, Some Research Issues and Challenges," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 358–380, 2015.

[22] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On Radio Access Network Slicing from a Radio Resource Management Perspective," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 166–174, 2017.

[23] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software Defined Radio Access Network," in *Proc. of ACM HotSDN*, Hong Kong, China, 2013.

[24] A. Gudipati, L. E. Li, and S. Katti, "RadioVisor: A Slicing Plane for Radio Access Networks," in *Proc. of ACM HotSDN*, Chicago, Illinois, USA, 2014.

[25] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks," in *Proc. of ACM CoNEXT*, Irvine, California, USA, 2016.

[26] R. Ferrus, O. Sallent, J. Perez-Romero, and R. Agusti, "On 5G Radio Access Network Slicing: Radio Interface Protocol Features and Configuration," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 184–192, 2018.

[27] A. Ksentini and N. Nikaein, "Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 102–108, 2017.

[28] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture," in *Proc. ACM MobiCom*, Snowbird, Utah, USA, 2017.

[29] P. H. A. Rezende and E. R. M. Madeira, "An adaptive network slicing for LTE radio access networks," in *Proc. of IEEE WD*, Dubai, United Arab Emirates, 2018.

[30] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen, and X. Li, "End-to-End Quality of Service in 5G Networks: Examining the Effectiveness of a Network Slicing Framework," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 65–74, 2018.

[31] M. Gramaglia, I. Digon, V. Friderikos, D. von Hugo, C. Mannweiler, M. A. Puente, K. Samdanis, and B. Sayadi, "Flexible connectivity and QoE/QoS management for 5G Networks: The 5G NORMA view," in *Proc. of the IEEE ICC*, Kuala Lumpur, Malaysia, 2016.

[32] F. Z. Yousaf, M. Gramaglia, V. Friderikos, B. Gajic, D. von Hugo, B. Sayadi, V. Sciancalepore, and M. R. Crippa, "Network slicing with flexible mobility and QoS/QoE support for 5G Networks," in *Proc. of the IEEE ICC*, Paris, France, 2017.

[33] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[34] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, "Programming Abstractions for Software-Defined Wireless Networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 146–162, 2015.

[35] "srsLTE," Accessed on 16.05.2019. [Online]. Available: http://www.softwareradiosystems.com/

[36] "NextEPC," Accessed on 16.05.2019. [Online]. Available: http://nextepc.org/

[37] "Ryu SDN Framework," Accessed on 16.05.2019. [Online]. Available: https://osrg.github.io/ryu/

[38] "The EmPOWER Protocol," Accessed on 16.05.2019. [Online]. Available: https://github.com/5g-empower/5g-empower.github.io/wiki/EmPOWER-Protocol

[39] R. Riggio, I. G. B. Yahia, S. Latr, and T. Rasheed, "Scylla: A language for virtual network functions orchestration in enterprise WLANs," in *Proc. of NOMS*, Instabul, Turkey, 2016.

[40] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, 2000.