5G-EmPOWER: A Software-Defined Networking Platform for 5G Radio Access Networks

Estefanía Coronado*, Shah Nawaz Khan‡, Roberto Riggio*
*Wireless and Networked Systems, FBK CREATE-NET, Trento, Italy
Email: {e.coronado, rriggio}@fbk.eu
‡Ericsson Research, Stockholm, Sweden
Email: shah.khan@ericsson.com

***Abstract*—Software-Defined Networking (SDN) is making their way into the fifth generation of mobile communications. For example, 3GPP is embracing the concept of Control-User Plane Separation (a cornerstone concept in SDN) in the 5G core and the Radio Access Network (RAN). In this paper we introduce a flexible, programmable, and open-source SDN platform for heterogeneous 5G RANs. The platform builds on an open protocol that abstracts the technology-dependent aspects of the radio access elements, allowing network programmers to deploy complex management tasks as *policies* on top of a programmable logically centralized controller. We implement the proposed solution as an extension to the *5G-EmPOWER* platform and release the software stack (including the southbound protocol) under a permissive APACHE 2.0 License. Finally, the effectiveness of the platform is assessed through three reference use cases: active network slicing, mobility management, and load-balancing.**

***Index Terms*—Network management, network programmability, SDN, 5G, LTE, open-source, experimental evaluation.**

I. Introduction

Mobile networks are currently witnessing a dramatic increase in the amount of data traffic exchanged by their users. This trend can be mainly ascribed to the rapid adoption of data-hungry mobile applications and services (e.g., Netflix). The fifth generation of the mobile network (5G) is expected to provide adequate support for such applications while enabling new service categories like massive Machine Type Communications (mMTC) and Ultra-Reliable Low Latency Communications (UR-LLC). Purpose-built radio interfaces (e.g., mmWave), heterogeneous Radio Access Networks (RANs), mixed small and macro cell coverage, and network virtualization are just some of the technological enablers that will be used to achieve the aforementioned goals. The immediate consequence of this evolution is the need of network management platforms to deal with a mixture of technologies characterized by very diverse protocol stacks and vendor-specific interfaces.

Software-Defined Networking (SDN) is set to play a key role in taming the growing complexity of 5G networks. SDN has already delivered similar promises in data centres and wired networks where the separation of the control plane from the user plane through a well-defined interface (i.e., the southbound interface with OpenFlow [1] playing the role of de-facto standard) has simplified the network architecture and its management. Attempts to bring similar concepts to mobile networks can be found in the literature for the radio [2], [3] and the core network [4]. Meanwhile, SDN solutions for non-cellular wireless networks have also appeared [5], [6], [7].

Little research has been performed in SDN abstractions specifically tailored to heterogeneous mobile networks. To the best of our knowledge most of, if not all, the works in the literature are: (i) theoretical/conceptual [2], [8], [9], (ii) focused on a single technology [5], [6], [7], or (iii) not abstracted enough from the underlying radio access technology [10], [11]. Moreover, if SDN is to be applied to 5G systems, it is of capital importance to allow network programmers to specify the behaviour of the network in a declarative way without having to deal with technology-dependent implementation details.

In this work we introduce an open-source SDN platform for heterogeneous RANs consisting of: (i) a radio access agnostic Application Programming Interface (API) that clearly separates control plane from user plane; (ii) a software agent able to operate with several radio access nodes (Wi-Fi and LTE); and (iii) a proof-of-concept *Software-Defined RAN (SD-RAN) Controller* implementation. The proposed platform also supports multi-tenancy and active RAN slicing. We validate these capabilities using three reference use cases namely RAN slicing, mobility management and load-balancing. This platform extends our previous work [5], which supported only IEEE 802.11-based WLANs, by building on a generic architecture compliant with 4G and 5G networks. Furthermore, the contributions of this paper in terms of southbound protocol definition, agent/controller design and implementation, as well as performance evaluation and statistical validation, are novel. Finally, to the extent of our knowledge, this is the first and only open-source SDN experimental platform simultaneously managing both Wi-Fi and LTE networks. We release both the protocol and the agent/controller implementation under a permissive APACHE 2.0 license[1].

The paper is structured as follows. Section II discusses the related work. Section III presents an overview of the proposed solution. The design and implementation details are provided in Sec. IV and Sec. V, respectively. Section VI reports the performance evaluation. The reference use cases are presented in Sec. VII. Finally, we draw our conclusions in Sec. VIII.

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).

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

II. Related Work

Control and User Plane Separation (CUPS) has become a cornerstone of the 3GPP 5G architecture [12], [13]. However, the full control/user plane separation is not trivial, and is being the subject of extensive research [14], [15]. CUPS is seen as one of the fundamental enablers for network programmability and end-to-end network slicing. Industry and standardisation entities are aware of the importance of network slicing in the 5G vision [16], which is reflected in the actions undertaken by the International Telecommunications Union (ITU) [17], the Next Generation Mobile Networks (NGMN) alliance [18] and the 5G-PPP European program [12].

Network slicing can be performed in two dimensions: vertically and horizontally. Vertical slicing focuses on the core domain for the segregation of vertical services, while horizontal slicing delves into resource sharing between nodes and network devices. Initially, the slicing concept concerned the 5G core network due to the efforts of 3GPP to reshape it towards a modular architecture supporting granular network functions. This laid the foundations of DECOR (3GPP Release 13 [19]) and evolved DECOR (eDECOR) included in Release 14 [20], a service-based core network intended to meet the needs of services and users with diverse characteristics.

In the core network the Evolved Packet Core (EPC) facilitates the mobility of the User Equipments (UEs), and provides Quality of Service (QoS) and policy controls. However, the scalability of the EPC may be limited due to the frequent synchronization required between its components. This problem is alleviated in PEPC [21] by consolidating the state of the EPC in one location and refactoring the way in which it is accessed. The virtualization of the EPC as a service on the cloud has also been investigated in EASE [22] with the purpose of creating on-demand cloud-based mobile core networks. However, its performance may be adversely affected when larger amount of data needs to be processed in the Virtual Machines (VMs). Similarly, SCALE [23] pursues higher scalability by virtualizing the Mobility Management Entity (MME) element in the EPC to replicate the state of the devices across several VMs, which makes this approach not suitable for services with low latency requirements. In addition, other works have explored customization and flexibility improvements in the core slicing [24], [25], [26], [27].

The origin of RAN slicing lies in the static principle of RAN sharing [28], [29]. In this respect, 3GPP defines two types of architecture according to the sharing level, namely Multi-Operator Core Networks (MOCN) and Multi-Operator RAN (MORAN) [30]. In the former the spectrum is shared among operators, while in the latter each of them is assigned dedicated frequency bands. Besides the properties offered in RAN sharing, network slicing in the RAN domain pursues performance and functional isolation, as well as service differentiation. Nevertheless, although these properties are already established in the wired domain, they are still an open issue in the wireless network segment given the scarce and limited frequency spectrum resources [31], [32].

From a system perspective, the CUPS concept becomes a key enabler for RAN slicing. This is the target of SoftRAN [2], where control operations can be centralized or distributed according to time requirements. This approach is also explored in Softmobile [9] by abstracting the control plane into several layers with the aim of issuing the control functions thorough APIs. Conversely, FlexRAN [10] implements a custom SD-RAN platform that enables RAN programmability and introduces a south-bound API to enforce various levels of centralization for allocating the resources of the slices. In terms of resource scheduling, RadioVisor [33] extends SoftRAN to enable resource sharing between control functions and to perform resource allocation according to the traffic demand of each slice. Nevertheless, isolation between slices is not ensured. The platform proposed in [34] allows flexible slice definition based on descriptors that characterize the policies and resources to be used. However, the resources are preallocated at the eNodeBs (eNBs) according to the specific policy of the slices. Although most of these works consider isolation and resource allocation features across network slices, they ignore the signalling needed for ensuring the control/user plane separation in the traditional RAN architecture.

In this context, abstracting and allocating the radio resources is of vital importance. The hypervisor, added at the MAC layer, is responsible for the abstraction, isolation and sharing of the resources. The relevance of this component is showcased in [35], where the hypervisor makes possible the virtualization of LTE eNBs. However, the work mainly focuses on algorithms for scheduling the air interface. Hydra [36] goes one step further to analyse the abstraction of physical radio resources into multiple virtual radio resources on top of the same eNB. In addition, other solutions can be found in the literature [37]. Building on the hypervisor's functions, the MAC scheduler must allocate the resources according to the SLA of the slices. In [38] a RAN slicing architecture introduces a two-level MAC scheduler to abstract Physical Resource Blocks (PRBs) among slices. Nevertheless, the resource customization is not considered. Orion [11] presents an hypervisor able to ensure slice-specific logic and resource isolation by mapping PRBs into virtual RBs through a set of abstractions. However, it does not take into account the CP/UP separation in disaggregated RAN deployments. In [39] authors propose a RAN slicing system that allows defining CP/UP functions for each slice, therefore enabling slice customization, isolation and resource sharing. Moreover, it allows specifying if the CP and the UP are shared or separately processed.

Despite the improvements, none of these works provides an SDN platform for heterogeneous Radio Access Technologies (RATs) that tackles the issue of separating control plane from user plane (and associated challenges such as protocol implementation) in a technology-agnostic manner. Moreover, many are shown from the conceptual standpoint, and not all are experimentally deployed. As additional value, the proposed platform supports multi-tenancy and active RAN slicing.

III. System Overview

In this section we provide a conceptual overview of the *5G-EmPOWER* platform. As already stated in the introduction, in this paper we report only on the LTE related aspects

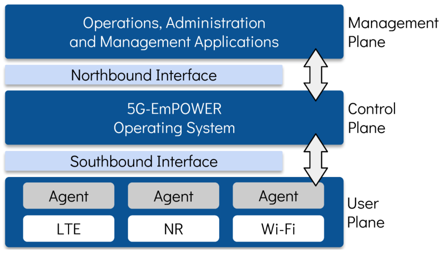

Fig. 1. The high-level view of the *5G-EmPOWER* architecture.

of *5G-EmPOWER*. Nevertheless, all the concepts presented hereby are RAT-agnostic and can be applied to other types of RANs, e.g., Wi-Fi and 5G New Radio (NR).

The system architecture of *5G-EmPOWER* follows the SDN structure defined in [40], [41] and is composed of three layers: user plane, control plane and management plane (Fig. 1). The user plane encompasses the network elements in the RAN, including LTE eNBs, Wi-Fi Access Points (APs), and possibly 5G gNodeBs (gNBs). The control plane consists of the *5G-EmPOWER* Operating System (OS), which is connected to the *5G-EmPOWER* Agents, one for each network element in the RAN (the reason for using the term operating system will be clarified in the next section). The *5G-EmPOWER* Agents receive the commands and configuration instructions issued by the operating system using the *OpenEmpower* Protocol [42] introduced in Sec. IV-B, which acts as southbound API (i.e., like the OpenFlow protocol for wired networks), and provides it with statistics and events from the network elements. Operation, administration and management applications reside at the management plane, which communicates with the control plane through the northbound interface.

The northbound interface provides useful abstractions for RAN management applications allowing them to access and manipulate the state of the network without dealing with the complexity of the underlying radio technology, thus enabling a vertical management design. In addition, it isolates and protects users and applications from each other, offering them a secure and dedicated view of the network. The applications supported by *5G-EmPOWER* range from basic monitoring applications that simply gather statistics from the network elements and report them to other applications (e.g., monitoring applications like Prometheus), to full fledged radio resource management applications implementing complex Self-Organizing Network (SON) features.

IV. Design Aspects

A. Control and Management Operations

All the layers in the access stratum of an LTE eNB (PHY, MAC, RLC, PDCP and RRC) can be decomposed into two planes: control plane and data plane. The former (control plane) is responsible for allocating resources that are then used by the latter (data plane). Notice that control operations have tight latency constrains, e.g., MAC scheduling decisions must be made with a granularity of 1ms to allow the eNB to cope with the stochastic channel fluctuations. Therefore, additional latency can severely impact the performance of the data plane.

In this paper we define control as the set of real-time operations executed at all layers of the LTE stack. Conversely, we use the term management for monitoring (checking if the operating conditions of a certain policy are still met) and repairing (reconfiguring or swapping the policy when its operating conditions are not satisfied) operations. For instance, a certain scheduling algorithm could be optimized for a uniform distribution of clients across the sectors of a mobile cell. However, if the client distribution is not uniform, a different policy could be required [43].

As a result, in our architecture we leave control functionalities close to the air interface, while we disaggregate the management functionalities from the eNB and move them to the management plane running on top of the *5G-EmPOWER* Operating System. We use the term operating system instead of SDN controller precisely to highlight this difference between control and management operations and to avoid the confusion of terming controller an element that is not actually implementing network control (in the mobile networking sense of the term), but is rather *enabling* management operations.

5G-EmPOWER is in charge of tasks that naturally belong to an operating system, namely: (i) allocating resources on the data plane on behalf of the users, which are generically referred to as anything sitting above *5G-EmPOWER* and consuming its northbound interface; (ii) providing isolation between users and their applications; (iii) multiplexing different users; (iv) abstracting network resources so that users do not have to handle RAT-specific details; (v) providing common services to the various applications and users; and (vi) providing the mechanisms whereby new types of devices can be added to the platform (i.e., like a device driver).

B. The OpenEmpower Protocol

The *5G-EmPOWER* platform clearly decouples control-plane operations, which are left at the air interface, from management-plane operations, which are consolidated on top of the operating system layer. An agent is introduced in the eNB to implement the management actions defined by the operating system. Communication between the agent and the operating system is performed through the *OpenEmpower* protocol. The *5G-EmPOWER* Operating System provides a reference implementation of the *OpenEmpower* protocol. However, implementations for other SDN solutions are possible, e.g., ONOS or OpenDayLight.

Even though OpenFlow is one of the most popular options adopted for implementing the so-called southbound interface, its features are mostly targeted at wired networks and are poorly suited for controlling heterogeneous radio access networks. In this context, the *OpenEmpower* protocol allows remote management of RAN elements without making assumptions about any particular type, i.e., it can be used on Wi-Fi APs, LTE eNBs, or 5G gNBs. At the time of writing, we have a reference implementation of the protocol for OpenWRT-based Wi-Fi APs and for LTE small cells based on the srsLTE stack [44]. Moreover we also have an implementation for a few commercial 4G/5G eNBs. The *OpenEmpower* protocol is layered on top of the Transmission Control Protocol (TCP)

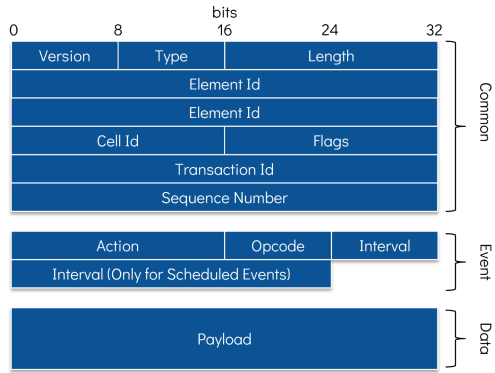

Fig. 2. The *OpenEmpower* message structure.

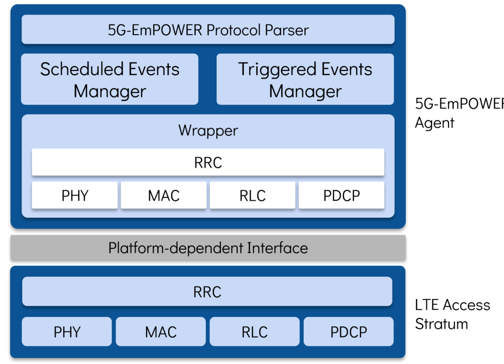

Fig. 3. The *5G-EmPOWER* Agent structure.

and can use the Transport Layer Security (TLS) protocol. The management plane should listen on TCP port 4433 for RAN elements that want to set up a connection.

The protocol is built around 3 major events or message types whose meaning is the following:

- *Single Events*. These are simple standalone events requested by the operating system plane and notified back immediately by the agent. No additional logic is bound to this message and the operating system decides when it is the time to issue the next event. Examples include RAN element capabilities requests and handover requests.
- *Scheduled Events*. These are events initiated by the operating system plane and then executed periodically by the agent. Examples include the PRB utilization requests, which require the agent to periodically send a PRB utilization report to the operating system plane.
- *Triggered Events*. These events enable/disable a certain functionality at the agent. They specify a condition that, when verified, triggers a message from the agent to the OS. Examples include the RRC measurements requests.

All the messages in *OpenEmpower* start with a common header that specifies the protocol version, the event type, the message length, the RAN element ID and the cell ID[2], the transaction ID, and a sequence number. The transaction ID is a 32-bits token associated with a certain request. Replies must use the same ID as in the request in order to facilitate pairing. This is necessary because all the communications using the *OpenEmpower* protocol are asynchronous. The sequence number is a counter incremented by one every time a message is generated by either an agent or the operating system plane.

The common header is followed by one of the three possible events headers. Each event header specifies the type of action, an operation code (opcode), and (in the case of a scheduled event) the event scheduling period. The opcode value depends on the particular type of action and can be used to indicate both error/success conditions or the type of operation (create, retrieve, update, or delete). Finally, after the event header, there is the body of the message, which differs from action to action. Figure 2 sketches the structure of an *OpenEmpower* message.

[2]Notice how the field cell ID can be used to address both an LTE cell (i.e., a sector) in an eNB or also a Wi-Fi interface in a multi-standard AP.

C. The 5G-EmPOWER Agent

The *5G-EmPOWER* Agent is in charge of managing the LTE user plane. An eNB integrating the *5G-EmPOWER* Agent gains the ability to interact with the *5G-EmPOWER* Operating System. The architecture of the *5G-EmPOWER* Agent is composed of two parts, the platform independent *5G-EmPOWER Agent* itself and the platform dependent *Wrapper*. The *Agent* consists of: (i) a protocol parser responsible for serializing and de-serializing the *OpenEmpower* messages, and (ii) two managers (one for single/scheduled events and one for triggered events). The *Wrapper* is responsible for translating *OpenEmpower* messages into commands for the LTE stack. Figure 3 sketches the structure of the *5G-EmPOWER* Agent.

The *Wrapper* defines a set of operations that an eNB must support to be part of a *5G-EmPOWER*-managed network, including, for example, getting/setting certain parameters from the LTE stack, triggering UE measurements reports, accounting for UE attach/detach events, issuing commands (e.g., perform a handover), and reconfiguring particular access stratum policies. These operations are invoked by the *5G-EmPOWER* Operating System through the *OpenEmpower* Protocol.

The *Wrapper* is structured in as many submodules as the layers in the LTE access stratum plus an additional module for the Radio Resource Configuration (RRC) functions. The implementation of these submodules is the responsibility of the eNB vendor and is platform-dependent. For example, an eNB stack implemented in Erlang will require a different *Wrapper* from another one written in C language. As a result, our architecture does not mandate for a particular communication mechanism between the *Wrapper* and LTE stack. As part of our open-source release, we provide a reference implementation of the *5G-EmPOWER* Agent and the *OpenEmpower* protocol in C language, which is outlined in detail in Sec. V-A.

V. Implementation Decisions

A. Structure of the 5G-EmPOWER Agent

The *Wrapper* imposes a very limited list of requirements on the LTE stack implementation. Specifically, at startup, the eNB (running the *5G-EmPOWER Agent*) must advertise its presence to the *5G-EmPOWER* Operating System by a *Hello Request* message. After, the eNB must follow an authentication process with the *5G-EmPOWER* OS. Currently, the authentication is

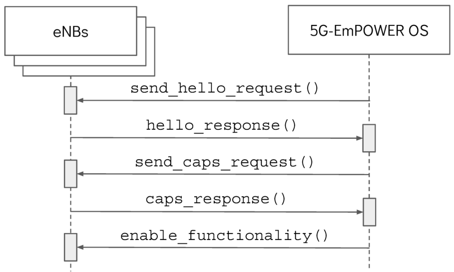

Fig. 4. Initial handshake between eNB and *5G-EmPOWER* Operating System.

based on Access Control Lists (ACLs) stored at the OS, which requires the network administrator to register just once the MAC address of the device through the web service introduced in Sec. V-B. However, it should be noted that a high-level security transaction running on top of a secure transport protocol can be added to improve this mechanism.

Once the eNB is accepted, the *5G-EmPOWER* Operating System sends a *Hello Response* message, and then a *Capabilities Request* message. Conversely, if it is not accepted, the *5G-EmPOWER* Operating System silently terminates the TCP connection. The *Capabilities Response* message contains information such as the eNB ID, the number of cells and their ID, and the uplink/downlink centre frequencies and bandwidth. This allows the OS to build a map of the RAN. Notice that the *Hello Request* message is periodically sent by the agents as heartbeat message. This process is depicted in Fig. 4.

The *Wrapper* also defines a set of optional operations. An implementation of the *Wrapper* can support all, none, or just some of such operations. The list of operations supported by a *Wrapper* is provided as part of the *Capabilities Response* message (as shown in Fig. 4). A partial list of these operations is reported in Table I. Each operation is associated with one or more *OpenEmpower* message types. For example, the UE Report operation is associated only with the *Triggered* message type. This is due to the fact that this operation generates an output only when a UE attaches to the eNB. Likewise, the MAC Report operation is associated to both the *Single* and the *Scheduled* types. This is because an eNB can be asked to periodically report the PRB utilization or to provide an instantaneous report of the current PRB utilization.

The *5G-EmPOWER* Agent also allows the replacement and/or reconfiguration of a control policy operating within any of the access stratum layers. For example, the *5G-EmPOWER* Agent could be instructed to replace the PRB scheduling from pure round robin to a weighted fair queuing policy. Then, the weights used by the new scheduling policy could be adjusted by the *5G-EmPOWER* Operating System using the same mechanisms. This design follows the separation highlighted in the previous sections whereby latency-sensitive operations are left at the air interface while the operating system plane implements management operations (i.e., monitoring and repairing). Notice that the benefits of this approach are twofold: (i) the latency and the bitrate constrains imposed on the signaling channel between Agents and Operating System are relaxed (see Sec. VI); and (ii) the eNB vendor has a tight control on what is exposed to the operating system layer.

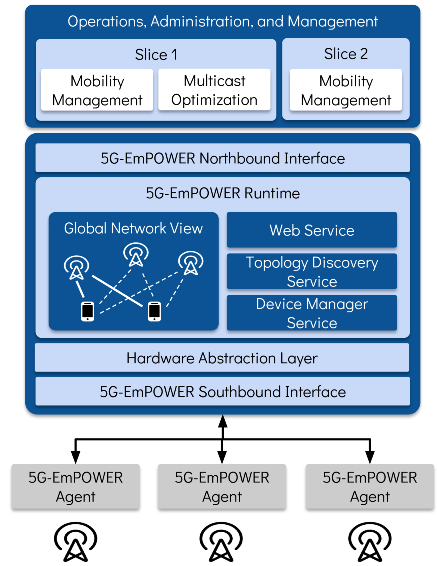

Fig. 5. The *5G-EmPOWER* Operating System.

B. The 5G-EmPOWER Operating System

The *5G-EmPOWER* Operating System provides a framework for managing heterogeneous RANs using the *OpenEmpower* protocol together with a collection of built-in functionalities and services that are useful to write management applications. The high-level view of the architecture is sketched in Fig. 5. The *5G-EmPOWER* Operating System is implemented in Python using the Tornado Web Server as web framework. The main reason for choosing Tornado is its non-blocking network I/O, which allows serving incoming requests while others are being processed. Managed memory, no-compilation time, dynamic typing, and precise error reporting are just some of the reasons that led us to pick Python as language of choice for *5G-EmPOWER* to maximize the developers' productivity.

Application Programming Interface (API). The API defined by the *5G-EmPOWER* Operating System has been designed with the express goal of shielding developers from the implementation details of the underlying wireless technology. Domain Specific Languages (DSLs) are known to be particularly suitable in such cases given that are tailored to a specific application domain. Common examples are the Cascade Style Sheet, used for formatting web pages, or SQL, used for relational databases. DSLs enable writing very concise programs. For example, implementing a handover for the UE *ue* to the cell *cell* could be a simple assignment operation $ue \leftarrow cell$.

DSLs impose a high entrance barrier and may not support the most common programming constructs expected from modern programming languages. Conversely, a language like Python enables certain language features (meta-programming), which allows writing code that resembles the domain-specific DSL syntax (these DSLs are typically called embedded or internal DSLs). For instance, the previous handover operation can be implemented in a single line of code, $ue.cell = new\_cell$, by relying on the ability of Python to map properties to operations. This assignment wraps the logic

TABLE I
ACTIONS SUPPORTED BY THE *5G-EmPOWER* PROTOCOL.

Operation	Event Type	Description
Hello	Single	Periodic heartbeat message sent by the eNB to the operating system.
Handover	Single	Triggers an X2 handover. The message specifies the UE RNTI and the target eNB / Cell.
MAC Reports	Scheduled/Single	Collects the PRB utilization statistics from the MAC scheduler (uplink/downlink).
UE Reports	Triggered	Triggers a message when UEs attach/detach from an eNB.
RRC Measurements	Triggered	Instructs a UE to start RSRP/RSRQ measurements on one or more channels and with certain interval.

required to trigger an X2 handover at the eNB hosting the UE toward a specific cell. This design choice allows programmers to leverage a high-level declarative API while being able to use any Python construct, such as threads, timers, sockets, etc.

Modular Architecture. With the exception of the logging subsystem (which must be available before any other module is loaded), every task supported by the *5G-EmPOWER* Operating System is implemented as a *plug-in* (i.e., a Python module) that can be loaded at runtime. Modules can be built-in and launched at bootstrap time or started and stopped at runtime. Each module consists of a Manifest file containing the module meta-data (version, dependencies, etc.), and one or more Python scripts. Every module is required to implement a `launch` method called when the module is loaded in order to perform the initialization tasks. The parameters accepted by the `launch` method are defined by the module.

The `launch` method must return an instance of either the base class *EmpowerApp* or the base class *EmpowerService*. Modules extending the *EmpowerApp* base class are instantiated within a certain slice. In this case the *tenant_id* parameter must be provided in order to specify on which slice the module shall operate. Notice that in this case the same module can be instantiated on multiple slices. Conversely, modules extending the *EmpowerService* base class can be used to implement common system components that can be reused across different slices. An optional *period* parameter can be used to specify the period of the control loop.

Developers are free to decide how their network management applications are deployed. For example, an application can be spread across several modules or can be implemented in a single one. This approach is similar, in principle, to the Network Function Virtualization (NFV) paradigm, where complex services can be deployed by combining several Virtual Network Functions (VNFs). Likewise, in the *5G-EmPOWER* Operating System, complex network management applications can be designed by instantiating different modules. This allows setting up a network monitoring application to perform a site survey or to roll-out new features at runtime by selecting them from an "app store". Figure 5 shows a scenario where the *5G-EmPOWER* Operating System is running three applications across two network slices with one application (the Mobility Manager) instantiated on both slices. Moreover, the figure depicts three important service modules, namely:

- *Device Manager Service*. Tracks the eNBs active in the RAN. This includes the IP address, the identifier, the last seen date, and the list of capabilities. The device manager also exposes an API allowing applications to receive events when new eNBs join or leave the network.
- *Topology Discovery Service*. Collection of modules that combines data from UE measurements (RSRP/RSRQ), eNB measurements, and external sources (e.g., spectrum databases) to build the Global Network View that can then be used by applications to implement management policies. Moreover, it allows retrieving network information as a list of links and nodes, and registering events to be notified when links are added or removed.
- *Web Service*. This module provides the interface that allows users to interact with the *5G-EmPOWER* Operating System. The module is split into two submodules: the REST server and the front-end Graphical User Interface (GUI). The benefit of this approach is the independence from the GUI given that any client that can consume a REST service can interact with the Operating System.

C. RAN Slicing

Network slicing enables the creation of logical networks customized with precise network resources and isolation properties, optimized to fulfil diverse performance requirements and to operate independently on a common infrastructure. Following this idea, *5G-EmPOWER* leverages network programmability to extend the SDN slicing concept of the wired domain to the wireless access segment in the case of the LTE RAN. The solution aims at ensuring efficient sharing of the physical infrastructure by different services providers. In particular, we assume that an infrastructure provider owns the physical LTE eNBs, which are leased to the service provider. However, details about pricing, although important, are out of the scope of this paper. Finally, notice that although the prototype focuses on LTE RANs, the design principles are general and easily extensible to other radio access technologies.

The *5G-EmPOWER* Operating System can accommodate multiple virtual networks or slices on top of the same physical infrastructure. Our slicing mechanism aims at achieving three goals: (i) performance isolation, (ii) slice customization, and (iii) efficient resource utilization. The first goal means that misbehaving slices should not affect the performance of other slices. The second goal means that slices should be allowed to freely allocate their resources. The third goal means that the slicing operation must efficiently use the radio resources.

Slices are created from a *slice descriptor* provided by the slice owner to the infrastructure provider, which specifies the Service Level Agreement (SLA) and the list of UEs to be mapped to each slice. The SLA identifies the service level requirements requested by the slice owner (e.g., the aggregated throughput, the number of PRBs, etc.). The *5G-EmPOWER* Operating System does not enforce a particular SLA. Instead,

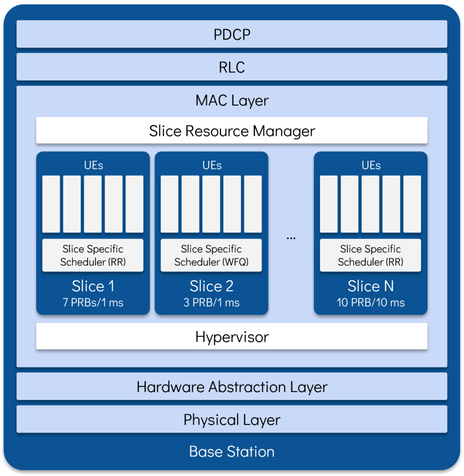

Fig. 6. The *5G-EmPOWER* slicing model.

a flexible framework is provided in order to allow implementing customized admission control and radio resource allocation mechanisms. A high-level representation of the *5G-EmPOWER* slicing model is depicted in Fig. 6. We shall now describe all the components in detail.

The *Slice Resource Manager* is responsible for the life-cycle management of the slices in a given eNB. Upon receiving a request for creating a new slice, the slice resource manager checks if the slice can be accepted using the admission control mechanism defined by the infrastructure manager. For example, if the resource allocation is done in terms of PRBs, the slice resource manager checks if there are enough PRBs to allocate the new slice. As a result, the slice is either admitted of rejected due to insufficient resources.

The *Hypervisor* is in charge of allocating the resources to each slice in order to meet their SLA. Specifically, it translates physical resources into virtual resources and hands them to a *Slice Specific Scheduler*. The decisions made by the *Slice Specific Scheduler* (which operates on virtual resources) are translated into physical resource commands by the *Hypervisor* before being delivered to the data plane. Every scheduling window, the *Hypervisor* computes the resources to be allocated to each slice. The scheduling window is a per-slice configurable parameter with the duration of a Transmission Time Interval (TTI), i.e., 1ms, or any of its multiples (e.g., 10ms or 100ms).

Each slice can use a different policy to schedule its UEs. Slice owners can select the UE scheduler from a list of available schedulers or can provide a new one as a plug-in. Notice that the *Hypervisor* gives to the *Slice Specific Scheduler* an abstracted view of the radio resources. This view includes only the resources available to that specific slice and thus omits those that are either allocated to other slices or dedicated to other purposes, e.g., random access, broadcasting, and control channels. The resource distribution of a slice is purely virtual and does not specify where they will be allocated in the physical resource grid. This allows the *Hypervisor* to re-assign the physical resources allocated to each slice in real-time.

Figure 7 shows an example of the *5G-EmPOWER* multi-level scheduling model for the specific case of an LTE cell with 5 MHz of bandwidth. In this example the owner of slice A requested 8 PRBs every TTI while the owner of slice B requested 3 PRBs. The remaining PRBs are left unused. As can be seen, the physical resource grid is abstracted by the *Hypervisor* into virtual PRBs. Virtual PRBs are then grouped by the *Hypervisor* into virtual PRB groups to model the constraints imposed by the particular physical layer used by the LTE cell. We remind the reader that the minimum granularity at which resources can be allocated in LTE depends on many factors such as allocation type and cell bandwidth. For example, an LTE cell with 5 MHz of bandwidth has 25 PRBs grouped into 13 PRB Groups with 2 PRB for each group (except the last one that has only one). Resource allocation must be done at the granularity of PRB Groups.

VI. Evaluation

In this section we study the overhead of the *5G-EmPOWER* Agent and the OS. First, we compare the performance of a vanilla eNB with an eNB running the *5G-EmPOWER* modules. After, we analyse the scalability of the *5G-EmPOWER* OS for an increasing number of eNBs and UEs.

A. *Overhead of the 5G-EmPOWER Agent*

The objective of this evaluation is to assess the impact of the *5G-EmPOWER* Agent on the radio access nodes with respect to a base system through three different scenarios. First, we study the performance of the base system (i.e., the vanilla LTE stack). Second, we evaluate the base system when the *5G-EmPOWER* Agent is running. Finally, building on this last scenario, we study the overhead associated to the RAN slicing features enabled by the *Hypervisor*. To do so we define two distinct configurations: (i) a single slice using 50% of the resources, leaving the remaining 50% free; and (ii) two slices with a proportional partitioning that allocates 50% of the PRBs to each of them. In all the scenarios 1 and 2 UEs have been connected to the network. Since the same conclusions can be inferred for a higher number of users, this study has been left aside in the interest of clarity. Measurements are 60 s long and are repeated 10 times. The Key Performance Indicators (KPIs) used are memory and CPU utilization.

The network topology considered is shown in Fig. 8, which comprises an LTE EPC, an LTE eNB, and the *5G-EmPOWER* Operating System. The EPC and the eNB are connected through the S1 interface, while the *OpenEmpower* Protocol handles the communication between the eNB and the *5G-EmPOWER* Operating System. The eNB has a capacity of 25 PRBs (i.e., 5 MHz bandwidth). The eNB and the EPC are deployed on Intel NUCs equipped with an i7 Intel processor and 16 GB of RAM running Ubuntu 18.04. NextEPC [45] is used as LTE EPC, while srsLTE [44] is used for implementing the LTE stack. Finally, the 2 UEs used are Huawei P10 Plus running Android 7.0.

For the results validation is equally important to demonstrate that there is no difference across the runs performed for the same experiment. Statistical tests are a powerful analysis tool that allows making inferences over the data. In particular, in this work we use them to evaluate the statistical differences

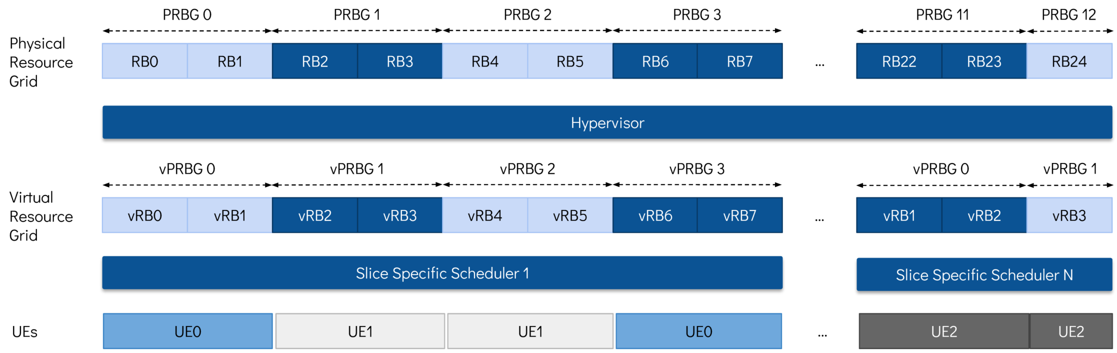

Fig. 7. Example of the different scheduling levels used by the *5G-EmPOWER Hypervisor*.

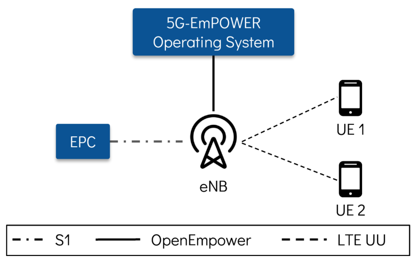

Fig. 8. Network topology used for the performance and overhead evaluation.

TABLE II
RESULTS OF THE SHAPIRO-WILK ANALYSIS FOR THE NORMALITY OF THE CPU UTILIZATION MEASUREMENTS OF THE *5G-EmPOWER* AGENT.

Scheme	**# Slices**	**P-value**	
		1 UE	**2 UEs**
srsLTE without Agent	-	7.77×10^{-8}	2.13×10^{-9}
srsLTE with Agent	-	1.66×10^{-10}	9.90×10^{-11}
RAN Slicing	1	3.69×10^{-8}	2.09×10^{-6}
	2	6.13×10^{-16}	3.07×10^{-9}

between the data sets, i.e., if the N runs of the same experiment are statistically equal. However, the tests to be used depend not only on the objective but also on the distribution and parity of the data, and the number of samples and data sets.

The data normality is an underlying assumption of most statistical tests, which makes its analysis become a prerequisite. The Shapiro-Wilk test is able to assess the normality of numerical variables (below 2000 samples). It establishes as null hypothesis, H_0, that the data is normally distributed, while the alternate hypothesis, H_1, affirms that a normal distribution is not followed. The results of the Shapiro-Wilk test are presented in Table II for a 95% Confidence Interval (CI). In this table it is shown that the significance value ($p-value$) in all the cases is below 0.05, which rejects the null hypothesis and proves that the data does not follow a normal distribution.

While parametric tests assume the data normality, non-parametric tests do not imply any distribution constraint. Based on the results of the Shapiro-Wilk test, we can conclude that non-parametric tests fit better the problem. Furthermore, it is worth considering the parity of the data, i.e., if there is a relation between an entry $(X_0, X_1, ..., X_n)$ in a data set and the entry in the same position in another data set $(Y_0, Y_1, ..., Y_n)$ for the same experiment. This type of problems are called paired. Since the result at a given time must match for different executions, time-dependent data falls in this category, as it is the case of the experiments in this section.

TABLE III
RESULTS OF THE FRIEDMAN ANALYSIS FOR THE MEDIAN OF THE CPU UTILIZATION MEASUREMENTS OF THE *5G-EmPOWER* AGENT.

Scheme	**# Slices**	**P-value**	
		1 UE	**2 UEs**
srsLTE without Agent	-	0.117	0.181
srsLTE with Agent	-	0.134	0.094
RAN Slicing	1	0.067	0.399
	2	0.456	0.642

Most of the normality tests are just valid for comparing two data sets. However, in this case, 10 repetitions per experiment are performed. From these constraints it can be said that the Friedman test is the only one suitable to tackle this problem. This test establishes as null hypothesis, H_0, that the medians of the data sets are statistically equal, while the alternate hypothesis, H_1, states that strong differences exist between them. Table III shows the results of this analysis, where it can be seen that regardless of the system evaluated, the $p-value$ is higher than 0.05. Consequently, the null hypothesis is accepted, proving that there is no statistical difference across the executions and the repeatability and accuracy of the results.

Figure 9 depicts the outcomes of the CPU utilization. As can be seen, these measurements are around 60% and 70% for 1 and 2 UEs, respectively, for the srsLTE stack, which is actually a concern also reached in the literature [46]. On this basis, we have examined the impact of introducing the *5G-EmPOWER* Agent, as well as of enabling the RAN Slicing system. Although for 1 UE the overhead of the *5G-EmPOWER* Agent is slightly greater, when enlarging the number of UEs, the CPU utilization of the vanilla srsLTE system increases considerably, while it is almost negligible when the *5G-EmPOWER* Agent or the *Hypervisor* (for the RAN Slicing features) are active. Furthermore, it is shown that instantiating several slices have little effect on the performance. Consequently, it can be concluded that varying

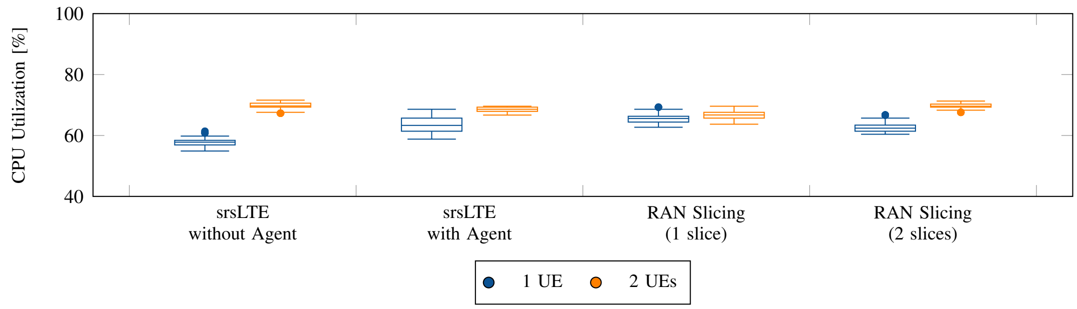

Fig. 9. Results in terms of CPU utilization of a vanilla eNB when introducing the *5G-EmPOWER* Agent in various setup configurations.

TABLE IV
MEMORY CONSUMPTION OF A VANILLA ENB WHEN INTRODUCING THE *5G-EmPOWER* AGENT FOR VARIOUS SETUP CONFIGURATIONS.

Scheme	**# Slices**	**Memory Utilization [%]**	
		1 UE	**2 UEs**
srsLTE without Agent	-	4.3	4.4
srsLTE with Agent	-	4.4	4.5
RAN Slicing	1	4.4	4.5
	2	4.4	4.5

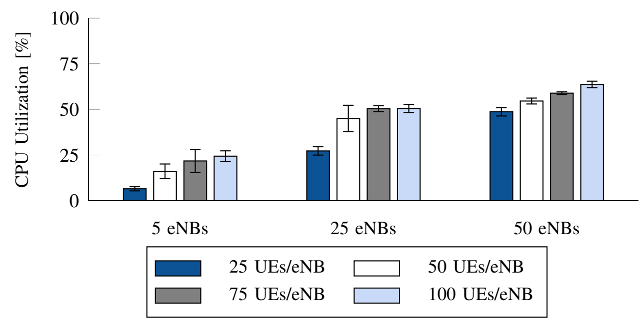

Fig. 10. CPU utilization of the *5G-EmPOWER* Operating System for an increasing number of eNBs and UEs.

the number of slices and UEs does not lead to a significant impact to the CPU performance of the eNB. The memory consumption is also an important KPI for understanding the overhead of *5G-EmPOWER*. Table IV reports these results, showing that when the *5G-EmPOWER* Agent is active, the memory consumption grows by 0.1%. This increase is insignificant especially because it remains constant regardless of the number of slices and UEs present.

B. Scalability of the 5G-EmPOWER Operating System

This evaluation analyses the ability of the *5G-EmPOWER* Operating System to manage dense deployments. The prototype presented in the previous section is based on an open-source implementation on srsLTE. Nevertheless, at present srsLTE and other open-source LTE stacks have limited support for more than 2 UEs [46], [47]. Thus, to evaluate the scalability of the *5G-EmPOWER* OS, we have implemented an eNodeB simulator. Such a simulator communicates with the OS through the *OpenEmpower* protocol. The OS is deployed on an Intel NUC with an i7 Intel processor, 16 GB of RAM and running Ubuntu 18.04.1, while the eNB and UE instances are deployed on another identical NUC.

In this deployment we simulate the connection of 5, 25, and 50 eNBs to the *5G-EmPOWER* OS. Then, we simulate a set of UEs ranging from 25 to 100 in steps of 25 UEs per eNB, reaching a total of 5000 UEs. These experiments (repeated 10 times) aim to evaluate the scalability of the *5G-EmPOWER* OS in scenarios with a heavy information interchange between the agents and the OS (i.e., at the southbound interface). One of these scenarios is found on handover processes. As will be detailed in Sec. VII-B, a handover involves a message interchange between the agents and the *5G-EmPOWER* OS for the UE context release and the connection update. To this end, and during 15 s, we trigger from the *5G-EmPOWER* OS periodic handovers of the UEs across the eNBs, increasing considerably the amount of data transmitted and received through the southbound interface. The most significant outcomes concern the CPU utilization as a result of the message parsing done by the OS. These outcomes are sketched in Fig. 10, where it can be seen that this KPI is below 70% even when the OS handles the requests from 50 eNBs and 5000 UEs. Moreover, the impact on the network is negligible, reaching a peak of 53.49KB in the southbound in the most dense scenario. Furthermore, the memory consumption is constant at 0.3% for all the experiments. Therefore, in summary, this analysis demonstrates that *5G-EmPOWER* provides the capabilities required to guarantee the network scalability.

VII. USE CASES

The elasticity of our architecture allows deploying flexible 5G networks and a wide range of services. In this section these features are validated through three use cases: (i) RAN slicing; (ii) mobility management; and (iii) load balancing.

A. RAN Slicing

Besides instantiating several virtual networks, RAN slicing must ensure functional and performance isolation across the slices. Following the same deployment presented in Fig. 8 we define a set of scenarios to demonstrate how our architecture is able to offer such capabilities. We remind the reader that this deployment comprises the EPC, an LTE eNB running the *5G-EmPOWER* Agent, the *5G-EmPOWER* OS and 2 UEs.

The results are the average of 10 runs in which a downlink TCP stream is transmitted to the UEs using iperf3. Each test

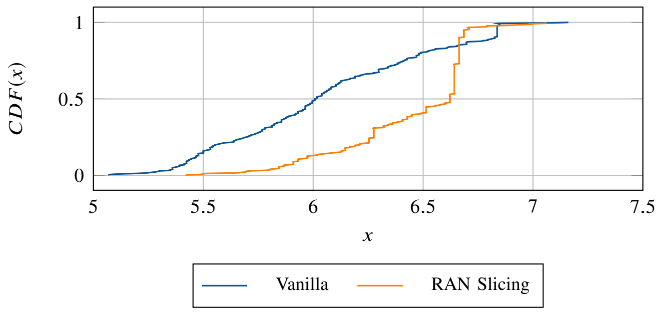

Fig. 11. CDF of the performance of one UE using the Vanilla and the RAN slicing system.

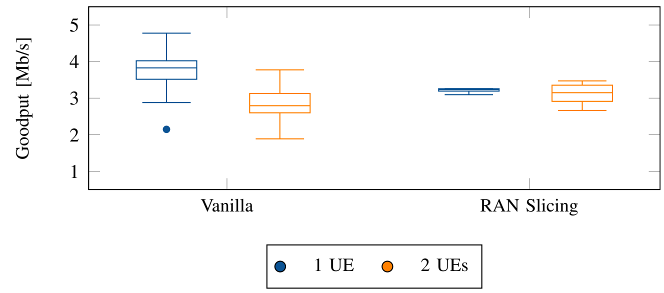

Fig. 12. Performance comparison for 2 UEs with the same signal quality and 2 network slices with 50% of the resources.

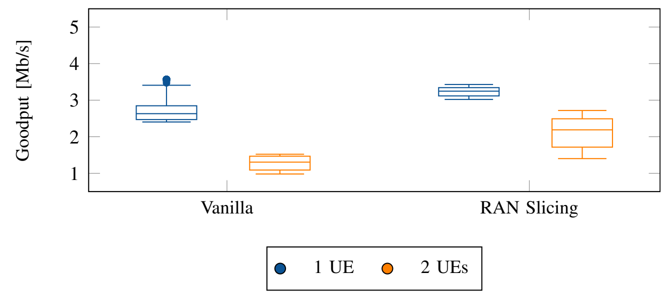

Fig. 13. Performance comparison for 2 UEs with different signal quality with 2 network slices with 50% of the resources.

is 60s long. Notice that the KPIs are focused on the goodput achieved while ensuring performance and functional isolation. Moreover, to draw a fair comparison, the measurements are repeated for a vanilla system just encompassing the EPC and a srsLTE-based eNB. In the case of the RAN slicing solution, two slices are accommodated in the network and 1 UE is connected to each of them. Conversely, in the vanilla system, 2 UEs are permanently attached to the network.

To determine the equality of the experiments, the statistical tests described in Sec. VI-A are performed. First of all, we use a Shapiro-Wilk test to evaluate the normality of the data, obtaining the results presented in Table V. These values denote that the experiments are not in line with a normal distribution given that in all cases $p{-}value < 0.05$ for a 95% CI. Based on this, we conduct a Friedman test to assess the equality of the medians across the 10 executions of the same experiment. It is important to remember that the Friedman test sets as null hypothesis that no statistical differences are found. Since in Table VI we can observe that $p{-}value > 0.05$, the Friedman test accepts the null hypothesis and demonstrates the equality of the medians of the data sets, thus 10 runs being enough to prove the accuracy of the experimental results.

Performance isolation. The first part of the evaluation assesses the performance isolation through three experiments where the resources are proportionally assigned to two slices, i.e., each slice is allocated 50% of the radio resources. It is worth recalling that the measurements are performed in the 5 MHz band, which implies a resource capacity of 25 PRBs per eNB. Given the constraint from the LTE stack from scheduling whole PRBs, assigning half of the resources to each slice leads to the situation in which one slice uses 13 PRBs, whereas the other utilizes 12 of them. For that reason, slight differences may be found in the plots shown below. Nevertheless, additional scheduling configurations are further studied in the evaluation of the functional isolation. Concerning the performance isolation, the characteristics and results of the experiments are presented below.

The first experiment is denoted as the base case since just 1 UE and 1 slice (in the case of the RAN slicing solution) are considered. Figure 11 sketches the CDF of the goodput achieved by the vanilla and the RAN slicing system. Although the two distributions follow a similar pattern, the probability of higher bandwidths is also higher for the RAN slicing solution. Furthermore, the range of values embraced by our solution is narrower than by the vanilla system, which demonstrates the ability to reduce the performance variability.

The second and third experiment intend to evaluate how the performance isolation is maintained when varying the signal quality of the UEs. In the second experiment, the two UEs have the same signal quality, while in the third one, one UE experiences bad channel conditions (we achieve this by placing the UE further away from the eNB than the other UE). In the RAN slicing system, the UEs are located on different slices. The results for the same signal quality are presented in Fig. 12. In particular, a higher difference in the performance of the two UEs can be appreciated for the vanilla system since resources are not properly scheduled. Conversely, our architecture is able to adequately allocate the resources, hence making equal the performance of the slices upon equal channel conditions. The necessity for isolation features is even more noticeable in Fig. 13, where the UE 2 receives a lower signal strength from the eNB. In comparison with the outcomes in Fig. 12, these measurements clearly show how in the vanilla system this issue not only concerns the UE having signal problems, but also impacts the performance of the first UE. By contrast, the RAN slicing solution isolates the problem to the slice affected, and maintains the performance for the remaining slices.

Functional isolation. The second half of the evaluation comprises three experiments that cover the functional isolation capabilities. With performance isolation we refer to the ability of ensuring that the performance of a slice is not affected by the resource allocation decisions in other slices. The first experiment highlights the slice customization capabilities while seeking full isolation. The second experiment focuses on the performance when accommodating several UEs in the same slice. Finally, the last experiment aims to show the flexibility

TABLE V
RESULTS OF THE SHAPIRO-WILK ANALYSIS FOR THE NORMALITY OF THE PERFORMANCE AND ISOLATION MEASUREMENTS.

	Scenario				P-value	
	# UEs	**Signal Quality**	**# Slices**	**Resource Assignment**	**UE 1**	**UE 2**
Vanilla	1	-	-	-	2.16×10^{-6}	-
	2	Same	-	-	9.77×10^{-9}	1.08×10^{-4}
	2	Different	-	-	9.77×10^{-9}	1.08×10^{-4}
RAN Slicing	1	-	1	50%	3.85×10^{-16}	-
	2	Same	2	50% - 50%	3.82×10^{-9}	2.20×10^{-16}
	2	Different	2	50% - 50%	8.87×10^{-12}	3.17×10^{-12}
	2	Same	1	50%	3.31×10^{-6}	2.20×10^{-16}
	2	Same	2	70% - 30%	1.02×10^{-6}	6.93×10^{-12}
	2	Same	2	50% - 50% Dynamic	9.77×10^{-9}	1.08×10^{-4}

TABLE VI
RESULTS OF THE FRIEDMAN TESTS FOR THE MEDIANS OF THE PERFORMANCE AND ISOLATION MEASUREMENTS.

	Scenario				P-value	
	#UEs	**Signal Quality**	**#Slices**	**Resource Assignment**	**UE 1**	**UE 2**
Vanilla	1	-	-	-	0.057	-
	2	Same	-	-	0.680	0.118
	2	Diff.	-	-	0.059	0.194
RAN Slicing	1	-	1	50%	0.162	-
	2	Same	2	50% - 50%	0.294	0.182
	2	Diff.	2	50% - 50%	0.708	0.321
	2	Same	1	50%	0.898	0.922
	2	Same	2	70% - 30%	0.539	0.213
	2	Same	2	50% - 50% Dynamic	0.680	0.119

of the platform to reallocate the radio resources when some of the slices are idle. The measurements are performed on the same network setup used in the previous evaluation.

Starting from the 50%/50% resource allocation of the previous experiments, the *5G-EmPOWER* OS is configured to assign 70% of the PRBs to one slice, and the remaining 30% to the second one. These resource distributions, depicted in Fig. 14, illustrate the inter-slice functional isolation of the architecture. In particular, it can be seen that the hypervisor guarantees that the slices (and hence the UEs) obtain just the radio resources configured. In the first scenario, each UE uses half of the resources, whereas in the second one the goodput of the first slice is higher due to the resource distribution.

To study the behaviour of several UEs in the same slice, the second experiment maintains the resource configuration that assigns half of the resources to each slice. This scenario is compared to a setup in which the 2 UEs are connected to one slice while the other remains idle. It is important to mention that this test deals with functional isolation aspects. Hence, the resource assignment is static, and their flexible reallocation will be further discussed later. From the results sketched in Fig. 15 we can observe that the resource configuration is respected within the slice, which leads to an equal division of the bandwidth between the UEs in that slice.

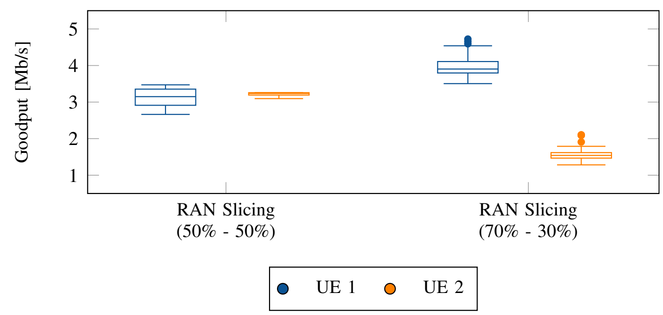

Fig. 14. Performance evaluation of different resource distributions for the RAN slicing solution with 2 UEs and 2 network slices.

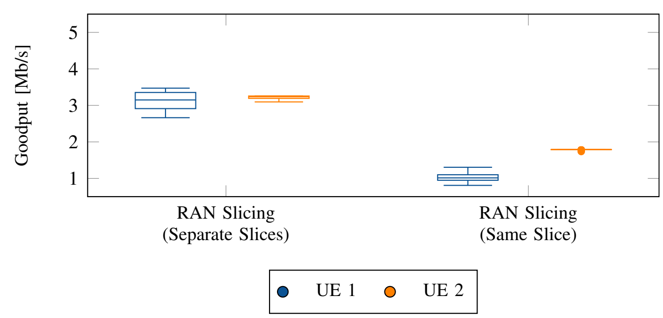

Fig. 15. Performance evaluation of the RAN slicing solution with 2 UEs and 2 network slices. In the first test the UEs are located in separate slices, while in the second one they share the same slice.

The last experiment intends to prove the elasticity provided by our RAN slicing solution. To this end, we build on the scenario made up of 2 slices and 2 UEs (i.e., 1 UE per slice) in which at the beginning the hypervisor configures an equal division of the resources. By contrast, as depicted in Fig. 16, in the first half of this test there is no traffic in the second slice (towards UE 2). In view of this, the scheduler is able to reallocate part of the idle resources from other slices to increase the performance. Notice that not all the idle resources are rescheduled since a minimum amount of PRBs must be reserved to keep alive the connection to the UEs in such slices. After 30s, a downlink transmission is started in the second slice and, as a result, the scheduler reassigns the resources to ensure the allocation agreement of each slice.

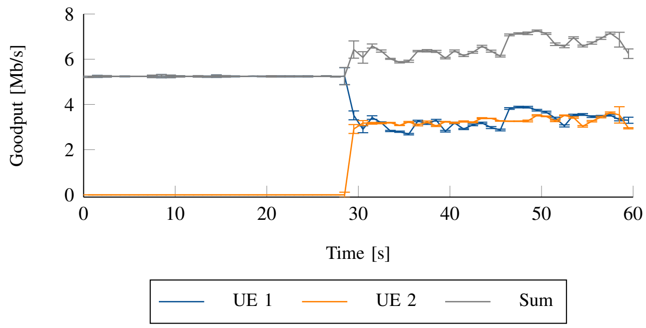

Fig. 16. Dynamic resource reallocation of 2 slices when a new UE is attached.

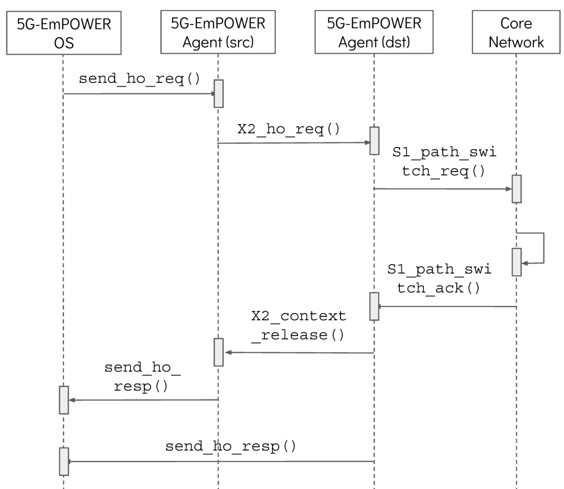

Fig. 17. Procedure handled by *5G-EmPOWER* to perform an X2 handover.

The previous experiment endorses how our architecture makes an efficient reallocation of the radio resources across co-existing slices with the aim of increasing the network performance. Moreover, it enables innovative models of radio resource sharing among the tenants such as those envisioned in the cognitive radio networks domain.

B. Mobility Management

On-demand mobility management is a must in 5G systems since constant UE mobility and switches across base stations are common scenarios to be supported by any practical SDN implementation. To illustrate this, we set up an scenario composed of 2 eNBs and 1 UE, maintaining the topology depicted in Fig. 8. The handover process is sketched in Fig. 17, where it is shown that when the *5G-EmPOWER* Operating System instantiates a handover, the *5G-EmPOWER* Agent running in the source eNB sends a message to the Agent in the destination eNB in order to trigger the barrier setup in the core network. After that, the *5G-EmPOWER* Agent on the destination informs the source one about the context release. Once this is done, the Operating System performs the operation in a transparent way for the UE.

The eNBs periodically (every 1s) report to the *5G-EmPOWER* OS the Reference Signal Received Power (RSRP) that the UEs receive from each eNB so that when the signal level is below the one from other eNB a handover is performed. This behaviour is depicted in Fig. 18. First, the UE is connected to the eNB 1. Then, as can be seen, it moves closer to the second eNB. After 18s, the signal from the second eNB is improved with respect to the current one, triggering a handover instantiated from the *5G-EmPOWER* Operating System to the eNB 2, which proves the ability of our platform to ensure seamless mobility management.

C. Load balancing

To be effectively used, networks need to scale out and balance the traffic load. This use case shows the ability of our solution to distribute the traffic of the UEs across eNBs taking into account traffic load and channel quality. To illustrate this, we set up an scenario using the EmPOWER eNB simulator described in Sec. VI on an Intel NUC (i7 Intel processor, 16 GB of RAM running Ubuntu 18.04.1). This scenario is composed of 15 UEs and 4 eNBs, which interact with the *5G-EmPOWER* Operating System through the corresponding Agent. The duration of the experiments is 30s.

The decision of the load balancing algorithm is based on the following conditions: (i) a UE can use a maximum of 5 PRBs in order to avoid consuming all the resources in an eNB; and (ii) in the case of enough resources at a given eNB, a UE will be connected to the eNB offering best channel quality. As depicted in Fig. 19, at the beginning of the test, the 15 UEs are attached to the same eNB. Since each eNB is configured with 25 PRBs (i.e., 5 MHz bandwidth) in average each UE can be assigned less than 2 PRBs. After 12s, we enable the load balancing "app". Following the aforementioned procedure, the algorithm distributes the UEs across the eNBs until they use the maximum number of PRBs permitted. This effect can be observed at 27s in Fig. 19.

VIII. Conclusions

The current SDN landscape lacks the tools, i.e., network controllers, data plane abstractions, and programming interfaces, able to manage heterogeneous mobile RANs. In this paper we cover this gap by introducing a complete platform named *5G-EmPOWER* that provides developers with expressive tools to handle the state of the network while hiding the implementation details of the underlying technology. The proposed platform catches the three fundamental pillars that compose a network management loop, namely collecting the network status, specifying the desired behaviour, and disseminating the new configuration.

5G-EmPOWER specifically accounts for the stochastic nature of the wireless links and for the significant heterogeneity that characterize modern RANs. This is indispensably translated into policies separation, i.e., the knobs to be turned to obtain the desired behaviour, and to putting the former in the hands of the network programmers who are not necessarily network experts while leaving the latter to equipment vendors. This essentially follows the personal computing trajectory where a set of open interfaces (i.e., the Intel x86 instruction set) paved the way to a rich ecosystem composed of several operating system and millions of applications and services.

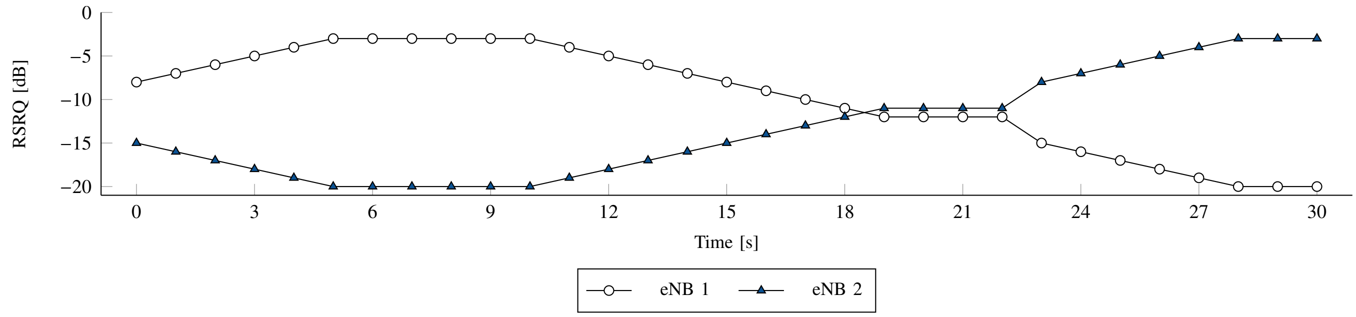

Fig. 18. Reported RSRQ by a UE with regard to 2 eNBs. When the current measured signal is lower than the one from another eNB a handover is performed.

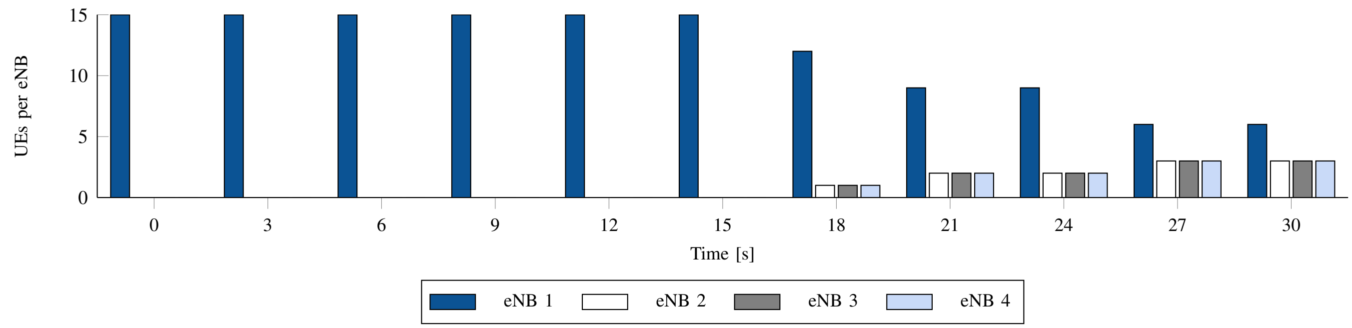

Fig. 19. Number of UEs connected to each eNB for a 30s test while they are distributed across the network to balance the traffic load.

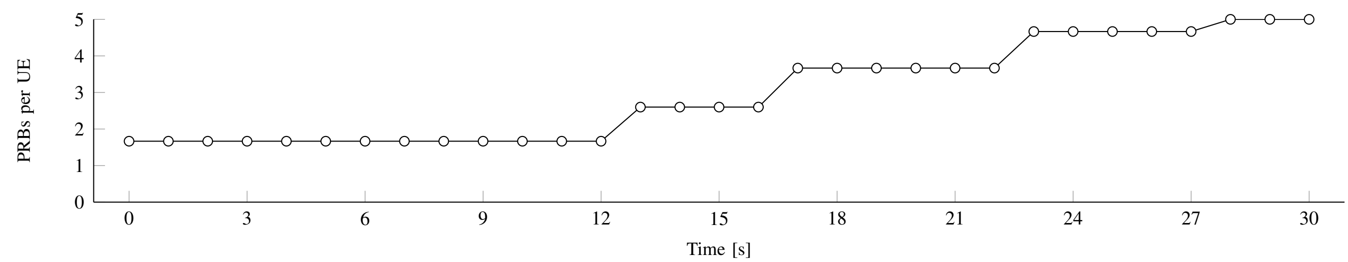

Fig. 20. Average number of PRBs per UE assigned in 4 eNBs for during 30s while 15 UEs are distributed across the network to balance the traffic load.

A proof-of-concept platform has been developed and released under a permissive open-source license. An extensive evaluation campaign has been conducted to demonstrate the low overhead introduced by the proposed platform. Moreover, we reported on three important use cases: RAN slicing, mobility management, and load balancing. As future work we plan to enhance the RAN slicing subsystem to introduce more elaborate SLA policies, e.g., admission control. We also plan to extend our platform to other radio access technologies and to add support for *Slice Specific Schedulers* written in high-level languages, e.g., Erlang or LISP.

References

[1] 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.

[2] 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.

[3] M. HadZialic, B. Dosenovic, M. Dzaferagic, and J. Musovic, "Cloud-RAN: innovative radio access network architecture," in *Proc. of IEEE ELMAR*, Zadar, Croatia, 2013.

[4] X. Jin, L. Li, L. Vanbever, and J. Rexford, "SoftCell: scalable and flexible cellular core network architecture," in *Proc. of ACM CoNEXT*, Santa Barbara, California, USA, 2013.

[5] 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.

[6] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise WLANS with Odin," in *Proc. of ACM HotSDN*, Helsinki, Finland, 2012.

[7] J. Schulz-Zander, C. Mayer, B. Ciobotaru, S. Schmid, and A. Feldmann, "OpenSDWN: Programmatic Control over Home and Enterprise WiFi," in *Proc. of ACM SOSR*, Santa Clara, California, USA, 2015.

[8] H. Ali-Ahmad, C. Cicconetti, A. de la Oliva, M. Drxler, R. Gupta, V. Mancuso, L. Roullet, and V. Sciancalepore, "CROWD: An SDN Approach for DenseNets," in *Proc. of EWSDN*, Berlin, Germany, 2013.

[9] T. Chen, H. Zhang, X. Chen, and O. Tirkkonen, "Softmobile: control evolution for future heterogeneous mobile networks," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 70–78, 2014.

[10] 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.

[11] 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.

[12] 5GPPP Architecture Working Group, "View on 5G Architecture," 2017.

[13] 3GPP, "TR 23.707. Architecture enhancements for control and user plane separation of EPC nodes," 2017.

[14] P. Arnold, N. Bayer, J. Belschner, and G. Zimmermann, "5G radio access network architecture based on flexible functional control/user plane splits," in *Proc. of EuCNC*, Oulu, Finland, 2017.

[15] 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.

[16] 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.

[17] ITU-T Focus Group , "TIMT-2020 Deliverables," 2017, Accessed on 27.03.2019. [Online]. Available: http://www.itu.int/pub/T-TUT-IMT-2017-2020

[18] NGMN, "Description of Network Slicing Concept," 2016, Accessed on 27.03.2019. [Online]. Available: https://www.ngmn.org/fileadmin/user_upload/160113_Network_Slicing_v1_0.pdf

[19] 3GPP, "TR 23.707. Architecture enhancements for dedicated core networks; Stage 2," 2015.

[20] ——, "TR 23.711. Enhancements of Dedicated Core Networks selection mech- anism (Release 14)," 2016.

[21] 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.

[22] T. Taleb, M. Corici, C. Parada, A. Jamakovic, S. Ruffino, G. Karagiannis, and T. Magedanz, "EASE: EPC as a service to ease mobile core network deployment over cloud," *IEEE Network*, vol. 29, no. 2, pp. 78–88, 2015.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] 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.

[30] 3GPP, "TR 23.251. Network sharing; Architecture and functional description," 2018.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel, "LTE wireless virtualization and spectrum management," in *Proc. of IEEE WMNC*, Budapest, Hungary, 2010.

[36] M. Kist, J. Rochol, L. A. DaSilva, and C. B. Both, "HyDRA: A hypervisor for software defined radios to enable radio virtualization in mobile networks," in *Proc. of IEEE INFOCOM WKSHPS*, Atlanta, GA, USA, 2017.

[37] ——, "SDR Virtualization in Future Mobile Networks: Enabling Multi-Programmable Air-Interfaces," in *Proc. of IEEE ICC*, Kansas City, MO, USA, 2018.

[38] 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.

[39] C. Y. Chang and N. Nikaein, "RAN Runtime Slicing System for Flexible and Dynamic Service Execution Environment," *IEEE Access*, vol. 6, pp. 34 018–34 042, 2018.

[40] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology," RFC 7426, 2015, Accessed on 27.03.2019.

[41] J. A. Wickboldt, W. P. D. Jesus, P. H. Isolani, C. B. Both, J. Rochol, and L. Z. Granville, "Software-defined networking: management requirements and challenges," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 278–285, 2015.

[42] "The OpenEmpower Protocol," Accessed on 27.03.2019. [Online]. Available: https://github.com/5g-empower/5g-empower.github.io/wiki/EmPOWER-Protocol

[43] R. Cohen and G. Grebla, "Joint Scheduling and Fast Cell Selection in OFDMA Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 114–125, 2015.

[44] "srsLTE," Accessed on 27.03.2019. [Online]. Available: http://www.softwareradiosystems.com/

[45] "NextEPC," Accessed on 27.03.2019. [Online]. Available: http://nextepc.org/

[46] F. Gringoli, P. Patras, C. Donato, P. Serrano, and Y. Grunenberger, "Performance Assessment of Open Software Platforms for 5G Prototyping," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 10–15, 2018.

[47] G. Garcia-Aviles, M. Gramaglia, P. Serrano, and A. Banchs, "POSENS: A Practical Open Source Solution for End-to-End Network Slicing," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 30–37, 2018.

BIOGRAPHIES

Estefanía Coronado is an Expert Researcher in the Wireless and Networked Systems (WiN) Unit at FBK CREATE-NET (Italy). In 2018 she completed her PhD at the University of Castilla-La Mancha (Spain) on the multimedia transmissions over SD-WLANs using Machine Learning techniques. She received the M.Sc. degrees in Computer Engineering and Advanced Computer Technologies in 2014 and 2015 from the same university. She published 10 papers in internationally refereed journals and conferences. Her research interests include wireless and mobile communications, 5G systems, SDN/NFV and network management.

Shah Nawaz Khan is a senior researcher at Ericsson Research working on 5G network architecture with emphasis on automation and orchestration. His current research interests include network virtualization and automation, RAN-cloud integration enabled by AI/ML and multi-domain resource orchestration. He holds a PhD degree in Computer Systems engineering and has wealth experience in academia and research organizations. He has worked on several European research projects on 5G networks covering topics such as SDN, NFV, spectrum sharing and self-organization, and is the author of several research publications on these topics.

Roberto Riggio is the Head of the Wireless and Networked System Unit at FBK CREATE-NET and Coordinator of the 5G-CARMEN H2020 project. His research interests include mobile networks, network slicing, light-weight virtualization, and distributed management and orchestration of network services. He has published more than 100 papers, receiving several research awards, and has generated more than 3M Euro in competitive funding. He serves in the TPC/OC of leading conferences and is associate editor of several journals including the IEEE Transactions on Network and Service Management.