This paper has been accepted for publication on the Proceedings of the 1st ACM Workshop on Open and AI RAN (Open-AI RAN '24). This is the author's accepted version of the article. The final version published by ACM is H. Cheng, S. D'Oro, R. Gangula, S. Velumani, D. Villa, L. Bonati, M. Polese, T. Melodia, G. Arrobo, C. Maciocco, "ORANSlice: An Open-Source 5G Network Slicing Platform for O-RAN" *Open-AI RAN '24: Proceedings of the 1st ACM Workshop on Open and AI RAN*, Washington, D.C., USA, November 2024.

ORANSlice: An Open-Source 5G Network Slicing Platform for O-RAN

Hai Cheng†, Salvatore D'Oro†, Rajeev Gangula†, Sakthivel Velumani†, Davide Villa†, Leonardo Bonati†, Michele Polese†, Gabriel Arrobo*, Christian Maciocco*, Tommaso Melodia†

† Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA

* Intel Labs, Intel Corporation, Santa Clara, CA, USA

ABSTRACT

Network slicing allows Telecom Operators (TOs) to support service provisioning with diverse Service Level Agreements (SLAs). The combination of network slicing and Open Radio Access Network (RAN) enables TOs to provide more customized network services and higher commercial benefits. However, in the current Open RAN community, an open-source end-to-end slicing solution for 5G is still missing. To bridge this gap, we developed ORANSlice, an open-source network slicing-enabled Open RAN system integrated with popular open-source RAN frameworks. ORANSlice features programmable, 3GPP-compliant RAN slicing and scheduling functionalities. It supports RAN slicing control and optimization via xApps on the near-real-time RAN Intelligent Controller (RIC) thanks to an extension of the E2 interface between RIC and RAN, and service models for slicing. We deploy and test ORANSlice on different O-RAN testbeds and demonstrate its capabilities on different use cases, including slice prioritization and minimum radio resource guarantee.

CCS CONCEPTS

• **Networks** → **Mobile networks**; **Network experimentation**.

ACM Reference Format:

Hai Cheng, Salvatore D'Oro, Rajeev Gangula, Sakthivel Velumani, Davide Villa, Leonardo Bonati, Michele Polese, Gabriel Arrobo, Christian Maciocco, Tommaso Melodia. 2024. ORANSlice: An Open-Source 5G Network Slicing Platform for O-RAN. In *Proceedings of The 1st ACM workshop on Open and AI RAN 2024 (ACM Open AI RAN '2024)*. ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Network slicing has been identified as a key technology to deliver bespoke services and superior performance in 5th generation mobile networks (5G). Specifically, Radio Access Network (RAN) slicing makes it possible to dynamically allocate a certain amount of RAN resources, e.g., Physical Resource Blocks (PRBs), to each slice based on their Quality of Service (QoS), current network conditions, and traffic load.

ACM Open AI RAN '2024, November 18th, 2024, Washington, D.C., USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00

https://doi.org/XXXXXXX.XXXXXXX

The importance of network slicing is emphasized by the O-RAN ALLIANCE, which has defined it as a critical use case and technology in the context of Open RAN systems [12]. The O-RAN architecture foresees slicing in the RAN and controlled through an xApp deployed on the near-real-time (Near-RT) RAN Intelligent Controller (RIC). The latter is connected to the RAN through the E2 interface, whose functionalities can be specified through *Service Models (SMs)*. For slicing, the E2SM Cell Configuration and Control (E2SM-CCC) [14] allows for near-real-time adaptation of slicing parameters via xApps [15].

Despite the potential of network slicing, which has generated tremendous momentum and technological advancements, practical deployment of RAN slicing in commercial 5G networks remains largely unrealized. Numerous studies and research have demonstrated the benefits of RAN slicing, exploring various aspects, including the use of optimization [8] and Artificial Intelligence (AI)-based solutions [20]. However, most of the existing implementations of RAN slicing are confined to research environments and bench setups.

RAN software stacks, such as OpenAirInterface (OAI) [9] and srsRAN [18], have been instrumental in advancing the field by welcoming contributions from the open-source community, for example, to integrate and release network slicing technologies for both 5G [6] and 4G [4] systems. Examples of this are provided by SCOPE [4] and FlexSlice [6]. The former is a network slicing framework based on srsRAN 4G, which, however, uses custom SMs to communicate with the xApps. The latter is a RAN slicing framework with a recursive radio resource scheduler based on OAI 5G. However, only a single slice can be associated to each User Equipment (UE), and it lacks integration with the O-RAN E2SM.

Contribution and Novelty. We advance the state-of-the-art by developing and implementing network slicing models compliant with the 3rd Generation Partnership Project (3GPP) and O-RAN specifications. *First*, we extend the 5G protocol stacks of OAI to support RAN slicing. We extend the original proportional-fair scheduler for UE scheduling to a two-tier radio resource scheduler for RAN slices and UEs. We also develop multi-Packet Data Unit (PDU) support for the OAI softwarized 5G UE, which enables multiple concurrent slices on the same UE. *Second*, we implement an E2SM-CCC-based SM and xApp for RAN slicing control. The implemented SM is O-RAN-compliant and is aligned with 3GPP specifications to enable closed-loop control for RAN slicing via xApps in the Near-RT RIC. *Third*, to ensure the robustness and effectiveness of our implementation, we conduct extensive testing and validation on Arena [3], a SDR-based wireless testbed located in an office environment, and X5G [19], an O-RAN-compliant production-ready private 5G network testbed. We show that our implementation can be used to

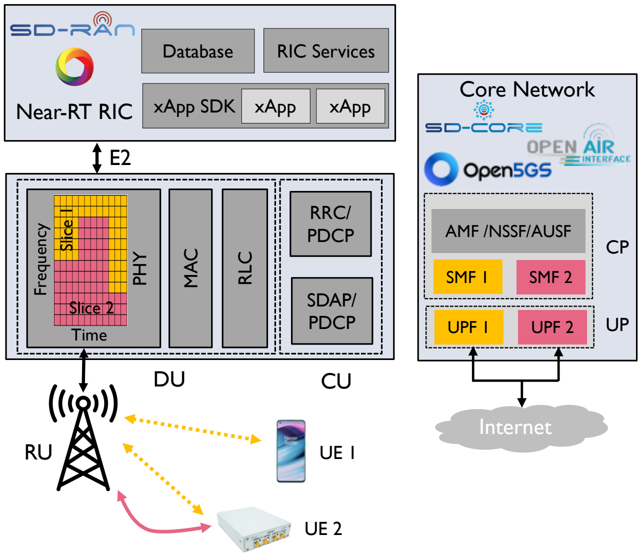

Figure 1: End-to-end network slicing in O-RAN.

enforce and control slicing policies with both Commercial Off-the-Shelf (COTS) 5G modules and softwarized UEs, as well as different radio devices. We also evaluate our implementation on different O-RAN RICs including the Aether SD-RAN and the O-RAN Software Community (OSC) Near-RT RICs.

By releasing ORANSlice to the community, we hope to bridge the gap between research and practice, providing a valuable resource for further exploration and development. [1]

Related Work. Efforts similar to ours have been documented in the literature. ProSlice in [10] developed a customized E2SM and xApp to support RAN slicing. However, ProSlice does not follow the 3GPP RAN slicing model and the E2SM is not O-RAN-compliant. Besides, ProSlice is not open-source. RadioSaber [7] is a channel-aware RAN slicing framework implemented in a RAN simulator. Finally, Zipper [2] is a RAN slicing framework based on closed-source commercial protocol stacks.

The closest works to ours are [10] and [6]. Compared to these works, ORANSlice advances the state-of-the-art by introducing support for multi-PDU and thus multi-slice management on the same UE. This enhancement allows for more efficient and flexible handling of different types of traffic, catering to the diverse needs of modern applications where diverse classes of traffic with different QoS requirements coexist on the same device and need to be satisfied simultaneously. ORANSlice will be publicly released upon acceptance of this paper, providing the research community with a 3GPP- and O-RAN-compliant implementation of RAN slicing on a 5G protocol stack. In this way, ORANSlice can be used to design novel 5G and network slicing functionalities, and test them with Software-defined Radios (SDRs), commercial Radio Units (RUs), or through the RFSim simulator, which allows OAI Next Generation Node Base (gNB) and UE to communicate over an Ethernet connection.

[1] https://github.com/wineslab/ORANSlice

2 O-RAN AND NETWORK SLICING

Before providing details on the implementation of ORANSlice, in the following we provide useful background on Open RAN, network slicing (with a specific focus on RAN slicing, as defined by the 3GPP [1]), and its role in O-RAN. The high-level system architecture of our end-to-end network-slicing solution is illustrated in Figure 1. First, we introduce the network architecture, with RAN, Near-RT RIC, and Core Network (CN). Then, we elaborate on how network slicing is supported by the three sub-systems. Finally, we present RAN slicing within the O-RAN architecture.

2.1 O-RAN Architecture

As shown in Figure 1, Open RAN gNBs are disaggregated into an RU, a Distributed Unit (DU), and Central Unit (CU) [17]. These elements connect to a core network with multiple Network Functions (NFs) managing mobility, session, authentication, billing, and routing to the public Internet. O-RAN also introduces the concept of RICs, i.e., software components that host applications to monitor, control and optimize RAN functions [15]. The Non-RT RIC is responsible for the optimization and control of RAN functions (or resources) on a time scale larger than 1 second via rApps. The Near-RT RIC is responsible for RAN optimization and control via xApps with a time requirement between 10 milliseconds to 1 second. Via xApps and rApps, the RIC leverages Key Performance Measurements (KPMs) and AI algorithms to compute control policies and enforce actions on the RAN.

2.2 End-to-end Network Slicing

A network slice is an end-to-end logical network spanning both RAN and CN. It can be dynamically created and configured to provide bespoke services to serve a diverse set of applications and use cases. In a cellular network, each network slice is uniquely identified by a Single Network Slice Selection Assistance Information (S-NSSAI), which consists of a Slice/Service Type (SST) and a Slice Differentiator (SD). SST is an 8-bit mandatory field identifying the slice type. SD is a 24-bit optional field that differentiates among slices with the same SST. It is worth mentioning that the same UE could be subscribed to up to 8 slices, which benefits applications with different network requirements.

RAN Slicing. Radio resources are organized in a time and frequency grid (Figure 1). Each grid element is referred to as a PRB and is used to schedule UE transmissions and broadcast control messages, among others. Network slicing in the RAN (i.e., RAN slicing), refers to the problem of allocating (and dedicating) such radio resources (i.e., PRBs) to different slices according to certain slicing policies.

Core Network Slicing. The 5G CN enables secure and reliable connectivity between the UE and the Internet. As shown in Figure 1, the 5G CN is decoupled into User Plane (UP) and Control Plane (CP). To enable network slicing in the CN (i.e., core slicing), dedicated NFs in both CP and UP need to be created for each slice. For example, a dedicated Session Management Function (SMF) and User Plane Function (UPF) pair can be created for each slice (Figure 1). The other NFs in the CN, such as the Access and Mobility Management Function (AMF), can instead be shared.

Multi-slice Support. 3GPP specifies a set of procedures to link UE traffic to a certain network slice. Specifically, during the PDU session establishment phase, the UE can specify the S-NSSAI of the target network slice. Upon establishment, the PDU session is assigned an IP, allowing applications in the UE to access the network service of the network slice by binding to that IP. This procedure makes it possible to establish multiple PDU sessions on the same UE, and each PDU can be bound to a dedicated network slice tailored to the considered use case or application executed by the UE.

2.3 RAN Slicing in O-RAN

The O-RAN WG3 has identified RAN slicing as a key use case in the context of the Near-RT RIC specifications [13]. It has also released the O-RAN service model E2SM-CCC [14] which details the structure and procedures necessary to enable RAN slicing in O-RAN following the 3GPP specifications described above. This is performed via xApps executing at the Near-RT RIC, where the limited radio resources are managed in near-real-time according to QoS requirements and highly varying RAN load and conditions.

3 ORANSLICE IMPLEMENTATION

This section details the design and implementation for the missing blocks required to support the RAN slicing use cases based on the O-RAN specifications.

From a software point of view, the open-source ecosystem already offers all architectural blocks necessary to instantiate and operate a 5G network. For example, disaggregated 5G base stations can be instantiated via OAI [9] and srsRAN [18], which also offer O-RAN integration and functionalities. A 5G core network can be instantiated via OAI, Open5GS [16], or SD-Core [11], all of which support core slicing. Similarly, the OSC and Aether offer open-source implementations of Near-RT RICs. OpenRAN Gym, an open-source project for collaborative research in the O-RAN ecosystem, provides components to connect across RAN and RICs [5].

What is missing is an open-source, 3GPP- and O-RAN-compliant implementation of RAN slicing functionalities and the support for multi-slice applications at the same UE. In this paper, we design and develop ORANSlice, an open-source extension to OAI to fill the gap between O-RAN and end-to-end network slicing.

3.1 RAN Slicing Enabled Protocol Stacks

The protocol stack of ORANSlice is based on OAI [9]. To enable RAN slicing, ORANSlice advances and extends the functionalities of the Medium Access Control (MAC) layer, including slice information of PDU sessions of each UE and a re-designed two-tier radio resource scheduler. Moreover, we extend the OAI nrUE to support the instantiation and management of multiple PDU sessions for different slices.

Slice-Aware MAC. In OAI, the MAC layer implements a *proportional-fair* scheduling algorithm to allocate radio resources between different UEs. However, to realize RAN slicing, the MAC scheduler needs to be aware of PDU-slice associations so as to properly allocate resources among users.

There are two basic types of RAN slicing schemes. The first one implements *slice-isolation*, where a fixed amount of resources are

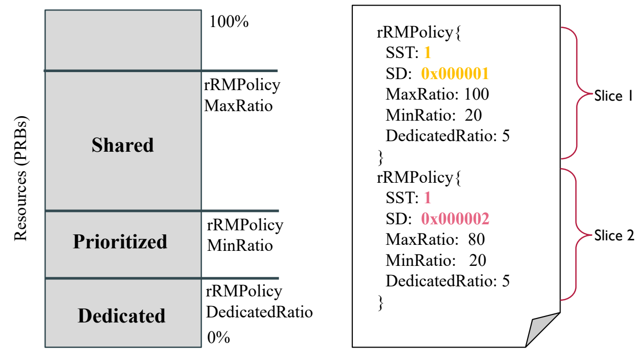

Figure 2: Illustration of RRM Policy Ratio Message

exclusively dedicated to each slice to guarantee resource availability [4]. This method completely prevents resource sharing, has a low resource utilization efficiency, as unused PRBs are not reallocated to other slices, but has a low-complexity implementation. The second one is the *slice-aware* scheme, where resources can be exclusively allocated to each slice, or shared among slices with a priority-based access mechanism. The slice-aware scheme increases the resource utilization efficiency as PRBs allocated to a slice, but unused, can be distributed across other slices on demand.

Since the second approach is more efficient, it has been selected by the 3GPP [1] and the O-RAN ALLIANCE [12] as a reference implementation for RAN slicing and, for this reason, we consider this in ORANSlice. In the context of RAN slicing, the 3GPP introduces the concept of Radio Resource Management (RRM) RAN slicing policy [1], as illustrated in Figure 2. The `rRMPolicyDedicatedRatio` represents the dedicated percentage of PRBs allocated to the network slice and is exclusively allocated to the slice even if the slice has no active traffic demand. The 3GPP also defines the concept of prioritized PRB access via two parameters: `rRMPolicyMinRatio` and `rRMPolicyMaxRatio`. The former represents the minimum percentage of PRBs that would be prioritized for allocation to the network slice. The latter represents the maximum percentage of PRBs that can be allocated to the slice. At a high level, the percentage of PRBs that falls within the prioritized range (i.e., defined as the difference between the maximum ratio and the dedicated ratio) is guaranteed to be allocated to the slice, but only if the slice has active users requesting PRBs.

ORANSlice extends the *proportional-fair* scheduler of OAI to integrate the slice-aware scheme described above. As we will describe later, this is achieved by implementing a two-tier resource allocation mechanism that considers: (i) inter-slice resource allocation and sharing according to RAN slicing policies, which will be elaborated in Section 3.2; and (ii) intra-slice resource allocation to schedule transmissions for all UEs that belong to the same slices.[2]

UE Slicing Support in OAI nrUE. The other important contribution of ORANSlice is the introduction of support for multiple slices on the same UE. Prior to ORANSlice, OAI softwarized 5G UE, i.e., OAI nrUE, supported only one PDU session per UE, limiting each UE to a single active slice at any time. To enable multi-slice support,

[2]We would like to mention that support for the dedicated ratio defined by the RRM policy in Figure 2 will be added in the future.

we have extended nrUE functionalities to: (i) enable the coexistence of multiple PDU sessions on each UE; (ii) instantiate/delete new PDUs on-demand; and (iii) assign each PDU to a slice. We already contributed these functionalities to OAI nrUE repository and they are available to the community as open-source components. It is worth mentioning that currently available 5G smartphones do not support these features, and we hope that this will help the community to investigate and explore RAN slicing topics that go beyond single UE-slice associations.

3.2 Implementing the E2SM-CCC

To enable the O-RAN RAN slicing use case, we also developed the corresponding E2 SM to realize the O-RAN E2SM-CCC service model. We developed two versions of the E2SM-CCC. One has been integrated with the Linux Foundation's Aether SD-RAN Near-RT RIC [11] and already made public,[3] while a simplified version has been integrated with the OSC Near-RT RIC. Due to space limitations, in this paper we describe and present results based on the latter only.

E2SM-CCC Service Model. The E2 SM describes how a specific RAN function (the *service*) within an E2 node interacts with the Near-RT RIC and its xApps. The E2 SM consists of an E2 Application Protocol (E2AP) and a data schema accepted by both the RAN function and Near-RT RIC. The flexible and programmable E2SM-CCC is critical for the autonomous and intelligent RAN control loop in O-RAN.

To support RAN slicing reconfiguration via xApps, we have implemented a simplified version of the E2SM-CCC service model integrated with the OSC Near-RT RIC to support 3GPP-compliant RAN slicing reconfiguration via xApps.

Only the *O-RRMPolicyRatio* configuration of E2SM-CCC is supported in the implemented service model. Other configurations defined in E2SM-CCC are omitted as they are beyond the scope of this paper. Protobuf is used to send control messages that include the RRM policy necessary to update RAN slicing strategies enforced by the RAN. We are currently extending our implementation to use ASN.1 definitions to encode/decode the message payload, as well as to adopt the full O-RAN E2SM-CCC specifications.

3.3 RAN Slicing xApp and Data-driven Control Automation

We developed a RAN Slicing xApp to compute RRM policies necessary to update RAN slicing strategies. Specifically, the RAN Slicing xApp periodically reads the RAN Key Performance Indicators (KPIs) from an InfluxDB database, processes them, and generates the slicing policies that are sent to the gNB through the E2 interface via a RIC Control message. Control messages from the xApp are serialized into Protobuf objects and sent via the SMs described above.

The data from the gNB is periodically sent to a dedicated OSC KPM xApp that reads KPMs received over the E2 termination through a RIC Indication message. The KPM xApp inserts the received RAN KPMs into the InfluxDB database that is leveraged by the RAN Slicing xApp. Similarly to the RIC Control messages, the payload of the RIC Indication messages is serialized into Protobuf objects.

[3] https://github.com/onosproject/onos-e2-sm/pull/392

4 TESTBEDS AND EXPERIMENT RESULTS

In this section, we demonstrate ORANSlice's portability by deploying an end-to-end O-RAN cellular network on two testbeds: Arena and X5G, discussed in Section 4.1. We demonstrate the functionality of ORANSlice by performing two experiments. In the first experiment (Section 4.2), we show an end-to-end network slicing deployment with KPM and RAN slicing xApp running in the OSC Near-RT RIC. The goal of this first experiment is to demonstrate the RAN slicing functionalities enabled by ORANSlice. In the second experiment (Section 4.3), we show a simple, yet illustrative, example of how to guarantee access to a minimum amount of radio resources via tailored RAN slicing policies in ORANSlice. In this latter experiment, we also demonstrate the multi-slice support offered by ORANSlice. In addition to the Over-the-Air (OTA) transmission experiments on two testbeds, we also replicate the two experiments in OAI RFSim mode to show the ability of ORANSlice working without physical UE or radio devices.

4.1 Multi-vendor O-RAN Testbeds

To demonstrate ORANSlice on the aforementioned testbeds, we deploy an end-to-end O-RAN 5G network system that matches the architecture illustrated in Figure 1. This system consists of the following four elements: UEs, gNB, Near-RT RIC, and 5G CN. While we use exactly the same UEs, Near-RT RIC and 5G CN across the OTA testbeds, Arena and X5G testbed use different gNB architectures as described below.

Arena Testbed and gNB. The Arena testbed is a remotely accessible wireless testing platform located in a large indoor office environment. It features 24 Universal Software Radio Peripheral (USRP) X310 and X410 connected to Dell EMC PowerEdge R340 servers via a 10/100 Gbps switch. The antennas are hung off the ceiling of an office space and connected to the USRPs for a total of 64 antennas in an 8×8 grid. For the 5G RAN deployed on the testbed, the USRP serves as RU and the OAI 5G protocol stack running on the Dell server serves as DU/CU. We consider a configuration with a bandwidth of 40 MHz (106 PRBs in 30 KHz subcarrier spacing).

X5G Testbed and gNB. The X5G testbed is a private, multi-vendor 5G network. The RAN protocol stack consists of: (i) NVIDIA Aerial, a GPU-accelerated L1 Physical (PHY) layer; (ii) higher layers implemented via OAI and running on X86 CPU; and (iii) a Foxconn RU operating in the n78 band and connected to the DU via the O-RAN fronthaul. In this testbed, we consider a bandwidth of 100 MHz with 273 PRBs and 30 KHz subcarrier spacing.

5G UE. We test both COTS and softwarized UEs. The COTS Sierra Wireless EM9191 5G module is selected to perform flexible experiments since it supports multiple PDU sessions and customized S-NSSAI for each PDU session. We also tested ORANSlice with the OAI nrUE, a softwarized UE that we have extended to enable multi-PDU sessions support and thus UE multi-slicing.

5G Core Network. To validate slicing operations, we tested ORANSlice with OAI CN, Open5GS, and SD-Core. These are modular core networks and support network slicing natively.

Near-RT RIC. The Near-RT RIC and the xApps of Figure 1 are deployed via Red Hat OpenShift. Specifically, we deployed the "E" release of the OSC Near-RT RIC together with the OSC KPM xApp and the RAN Slicing xApp.

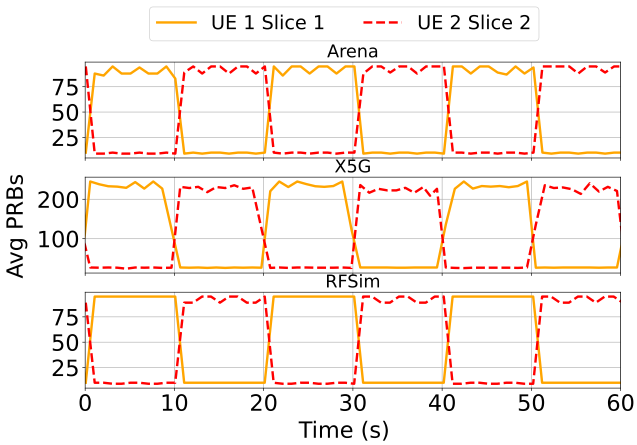

Figure 3: Average number of DL PRBs allocated to each slice in the first experiment.

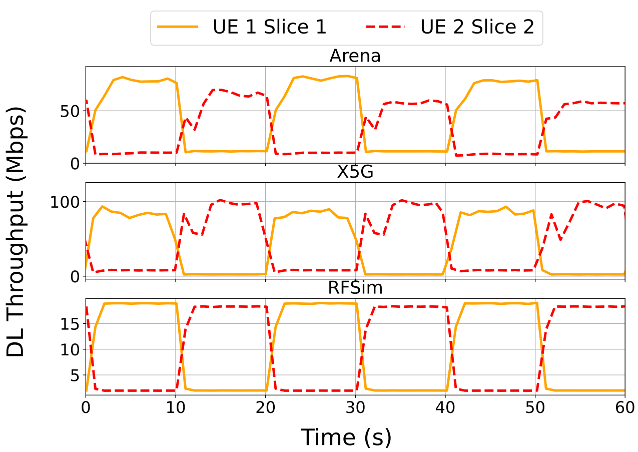

Figure 4: DL Throughput for each slice in the first experiment.

4.2 Testing RAN Slicing Control

In the Near-RT RIC, the deployed KPM xApp and RAN Slicing xApp work together to enable RAN slicing control. Every 0.5 seconds, the KPM xApp acquires KPIs for all connected UEs from the E2 interface. This includes user information (e.g., Radio Network Temporary Identifier (RNTI) and S-NSSAI) and KPMs such as Block Error Rate (BLER), Modulation and Coding Scheme (MCS), and throughput. This data is stored in an InfluxDB database in the form of time series data.

The RAN Slicing xApp reads the KPMs from the database and calculates the average downlink throughput for each slice for the previous 5 seconds. Then, it identifies the slice with the lowest and highest reported throughput and sets their `rRMPolicyMaxRatio` to 90% and 10%, respectively. The goal of this experiment is to demonstrate the correctness of the RRM policy update. How to use ORANSlice to satisfy a target QoS will be shown in Section 4.3.

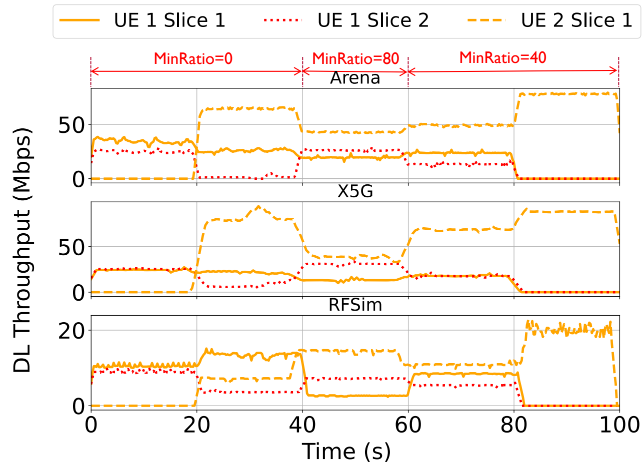

Figure 5: DL throughput for each slice and UE in the second experiment.

We consider two slices and two UEs, each associated to one slice. The control logic computes a new RRM policy every 10 seconds. As shown in Figure 3, the averaged PRBs per frame for UE 1 and UE 2 changes every 10 seconds. For the Arena testbed (which can allocate up to 106 PRBs), the PRBs of each UE varies between 10 and 90. For the X5G testbed, the PRBs of each UE range from 27 to 230 (the gNB in this testbed can allocate up to 273 PRBs). RFSim has the same 5G numerology as Arena, and the average number of allocated PRBs follows that of the same pattern. In all cases, the number of PRBs allocated to each slice is approximately equal to 10% and 90% of the number of available PRBs in each testbed, which proves that the RRM policy is applied correctly. In Figure 4, we also show the impact that the slicing policy has on downlink throughput of each slice, which depends on the amount of PRBs available to the slice. We see the downlink throughput for the two UEs varies following the number of PRBs in Figure 3. The difference between the throughput values of UEs is related to the differences in the hardware and RF channel peculiar to each testbed.

4.3 Multi-Slice with Guaranteed PRBs

In this experiment, we show how ORANSlice handles multi-slice applications and guarantees minimum PRB allocation by fine-tuning `rRMPolicyMinRatio` in the RRM policy. We consider two slices and 2 UEs, and UE 1 activates two PDUs each with a different S-NSSAI. This means that UE 1 has active PDUs on both slices. We set `rRMPolicyMinRatio = 0` and `rRMPolicyMaxRatio = 100` for all slices, i.e., no minimum PRB guarantee.

The experiment evolution is illustrated in Figure 5. In step 1 (0 ∼ 20s), UE 1 establishes two PDU sessions, with one PDU session associated with each slice. For each PDU, UE 1 generates TCP downlink data via `iPerf3`, while UE 2 is inactive. We notice that the throughput levels achieved by the two PDUs of UE 1 are comparable thanks to the proportional-fair scheduler. In step 2 (20 ∼ 40s), UE 2 establishes a PDU session associated to slice 1, and starts an `iPerf3` TCP downlink data transmission. The throughput of UE 1's PDU associated with slice 2 decreases to about 1 Mbps, due to the resource competition caused by the new establishment of UE 2's PDU on slice 1. Moreover, since `rRMPolicyMinRatio` = 0 for all slices, the gNB does not attempt to improve the throughput of UE 1's PDU associated with slice 2 as the RRM policy does not specify any guaranteed PRB provisioning for slice 2. In step 3 (40 ∼ 60s), the `rRMPolicyMinRatio` of slice 2 is set to 80, which makes the throughput of UE 1's PDU associated to slice 2 increase thanks to the minimum PRBs guarantee. Similarly, in step 4 (60 ∼ 80s), the `rRMPolicyMinRatio` of slice 2 is decreased from 80 to 40 and the throughput of UE 1 on slice 2 decreases. In step 5 (80 to 100 seconds), UE 1 stops the `iPerf3` transmission at the beginning of this step, while UE 2 stops it at the end. This experiment demonstrates how fine-tuning the `rRMPolicyMinRatio` parameter for each slice can guarantee a minimum PRB level for individual slices.

5 CONCLUSIONS

This paper presents ORANSlice, an open-source 5G framework for network slicing in O-RAN. ORANSlice extends OAI to deliver 3GPP-compliant RAN slicing and support for multi-slice applications. It also integrates an E2 service model based on E2SM-CCC, and a RAN Slicing xApp working with the OSC Near-RT RIC. The RAN slicing functionalities of ORANSlice are demonstrated on two over-the-air O-RAN testbeds and on the OAI RF simulator for different use cases. We hope that ORANSlice can accelerate the RAN slicing research by providing an open-source 3GPP- and O-RAN-compliant RAN slicing framework.

ACKNOWLEDGMENT

This work was supported by the U.S. National Science Foundation under grant CNS-2117814.

REFERENCES

[1] 3GPP. 2021. *5G Network Resource Model (NRM).* TS 28.541. 3rd Generation Partnership Project (3GPP). Version 16.4.0.

[2] Arjun Balasingam, Manikanta Kotaru, and Paramvir Bahl. 2024. Application-Level Service Assurance with 5G RAN Slicing. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX Association, Santa Clara, CA, 841–857.

[3] Lorenzo Bertizzolo, Leonardo Bonati, Emrecan Demirors, Amani Al-shawabka, Salvatore D'Oro, Francesco Restuccia, and Tommaso Melodia. 2020. Arena: A 64-antenna SDR-based Ceiling Grid Testing Platform for Sub-6 GHz 5G-and-Beyond Radio Spectrum Research. *Computer Networks* 181 (2020).

[4] Leonardo Bonati, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. 2021. SCOPE: An Open and Softwarized Prototyping Platform for NextG Systems. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*. 415–426.

[5] Leonardo Bonati, Michele Polese, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. 2023. OpenRAN Gym: AI/ML development, data collection, and testing for O-RAN on PAWR platforms. *Computer Networks* 220 (2023), 109502.

[6] Chieh-Chun Chen, Chia-Yu Chang, and Navid Nikaein. 2023. FlexSlice: Flexible and real-time programmable RAN slicing framework. In *IEEE Global Communications Conference (GLOBECOM)*. 3807–3812.

[7] Yongzhou Chen, Ruihao Yao, Haitham Hassanieh, and Radhika Mittal. 2023. Channel-Aware 5G RAN Slicing with Customizable Schedulers. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. USENIX Association, Boston, MA, 1767–1782.

[8] Salvatore D'Oro, Leonardo Bonati, Francesco Restuccia, and Tommaso Melodia. 2021. Coordinated 5G network slicing: How constructive interference can boost network throughput. *IEEE/ACM Transactions on Networking* 29, 4 (2021), 1881–1894.

[9] EURECOM. 2024. OpenAirInterface5G. https://openairinterface.org/

[10] Ahan Kak, Van-Quan Pham, Huu-Trung Thieu, and Nakjung Choi. 2022. ProSLICE: An Open RAN-based approach to Programmable RAN Slicing. In *IEEE Global Communications Conference (GLOBECOM)*. 197–202.

[11] Linux Foundation. 2024. Aether Project. https://aetherproject.org

[12] O-RAN WG1. 2024. O-RAN Slicing Architecture 13.0. O-RAN.WG1.Slicing-Architecture-R003-v13.00 TS.

[13] O-RAN WG3. 2024. Near-Real-time RAN Intelligent Controller Use Cases and Require 6.0. O-RAN.WG3.UCR-R003-v6.00 TS.

[14] O-RAN WG3. 2024. O-RAN E2 Service Model (E2SM) Cell Configuration and Control 4.0. O-RAN.WG3.E2SM-CCC-R003-v04.00 TS.

[15] O-RAN WG3. 2024. O-RAN Near-RT RAN Intelligent Controller Near-RT RIC Architecture. O-RAN.WG3.RICARCH-v06.00 TS.

[16] Open5GS team. 2024. Open5GS. https://github.com/open5gs/open5gs

[17] Michele Polese, Leonardo Bonati, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. 2023. Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. *IEEE Communications Surveys & Tutorials* 25, 2 (2023), 1376–1411.

[18] srsRAN Project. 2024. srsRAN. https://www.srslte.com/

[19] Davide Villa, Imran Khan, Florian Kaltenberger, Nicholas Hedberg, Ruben Soares da Silva, Stefano Maxenti, Leonardo Bonati, Anupa Kelkar, Chris Dick, Eduardo Baena, Josep M. Jornet, Tommaso Melodia, Michele Polese, and Dimitrios Koutsonikolas. 2024. X5G: An Open, Programmable, Multi-vendor, End-to-end, Private 5G O-RAN Testbed with NVIDIA ARC and OpenAirInterface. *arXiv:2406.15935 [cs.NI]* (June 2024), 1–15.

[20] Han Zhang, Hao Zhou, and Melike Erol-Kantarci. 2022. Federated Deep Reinforcement Learning for Resource Allocation in O-RAN Slicing. In *IEEE Global Communications Conference (GLOBECOM)*. 958–963.