Demo: Efficient Multi-Service RAN Slice Management and Orchestration

Robert Schmidt and Navid Nikaein
EURECOM, Sophia-Antipolis, France
E-Mail: robert.schmidt@eurecom.fr, navid.nikaein@eurecom.fr

***Abstract*—The 5G mobile network is supposed to handle a variety of services with different requirements. By means of virtualization, network slices form customized virtual networks transporting services with associated service guarantees. Especially the radio access network (RAN) requires an efficient multiplexing of multiple services onto the sparse radio resources. In this demo, we show how a RAN can be dynamically customized without service interruptions for different slices. In particular, our solution considers the slice requirements and adapts the slicing algorithm without interrupting other slices in the network. This allows an efficient resource usage while respecting isolation and performance requirements, in particular latency. Furthermore, dynamic end-to-end slicing is enabled by automatically adding core networks as required by the slice owner. Finally, this solution allows to compare different slice algorithm implementations.**

I. Introduction

Network slicing is a key enabler for the service-oriented 5G network. It allows to overlay a physical network with multiple virtual networks (slices) in order to share resources efficiently while providing functional flexibility. This is driven by an increasing business aspect in which mobile network operators allow third parties such as verticals to customize a network towards their services, such as enhanced mobile broadband (eMBB) and ultra-reliable low-latency communications (URLLC).

In particular, the scarce radio resources in the radio access network (RAN) need to be shared between multiple service providers while guaranteeing isolation and allowing to customize functionality. In the last years, a number of architectures and slice algorithms have emerged. However, the realization of encapsulated slices within the RAN with support of varying quality of service (QoS) levels, in particular latency, is still an open problem. Also, experimental research possibilities for practical slicing implementations are limited. Finally, it is difficult to directly compare existing slicing algorithms since their implementation was only in simulators or over different technologies.

II. Demo Description and Contribution

We extended our previous contribution [1], [2] and designed and implemented a RAN slice architecture in OpenAirInterface [3] as shown in Figure 1. The customizable slice scheduler identifies the radio resources to be used by each slice according to an algorithm as detailed below. Each slice can then allocate resources to its associated users by using a custom packet scheduler. The result of possibly multiple slice allocations in a subframe is then merged and user data is jointly multiplexed onto the physical resources. A simple interface between the slice scheduler and the individual packet schedulers allows a clear separation of concern between slice resource allocation and user resource allocation, making it possible to independently design and implement the respective algorithms.

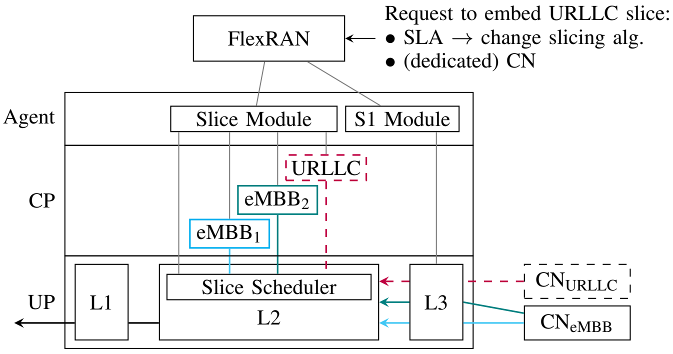

Fig. 1. The slice scheduler multiplexes the scheduling results of slice-specific packets schedulers (eMBB, URLLC) onto physical resources. All are configured through the slice module. Upon the request to embed a new slice, the RAN connects to a CN, changes the slice algorithm and adds the slice.

Through the use of the software-defined RAN controller FlexRAN [4], slices can be dynamically added, removed or modified by the help of a slice management module in the RAN controller agent. It is possible to dynamically change parameters of a slice or its packet scheduler. Also, the slice scheduling algorithm might be changed through the controller "on-the-fly" to fulfill all the slices' service level agreements (SLA). This requires the slice parameters to be mapped from one slicing algorithm to another. As an example, consider static slicing according to a percentage. This might be mapped to an equivalent rate in another algorithm. On the contrary, a latency threshold cannot be expressed with static slicing.

Using this architecture, in this demo we make the following contributions:

1) **Dynamic and efficient slice management**. The controller can embed new slices and modify their parameters. Furthermore, packet schedulers can be customized to suit a particular slice owner's need, allowing slice-specific scheduling strategies and taking into account service-specific performance requirements.

978-1-7281-4973-8/20/$31.00 © 2020 IEEE

2) **Multiple slicing algorithms**. We implemented (i) static resource reservation for each slice in every subframe, (ii) NVS [5] and (iii) a QoS-sensitive (latency) slicing algorithm [2]. These algorithms guarantee resource isolation to a varying degree between multiple slices. Also, this allows a fair comparison of these algorithms.
3) **Description of different slice requirements**. Through a descriptor, an SLA for each slice is expressed. Consider the scenario in Figure 1. Two eMBB slices are present and are multiplexed using a static slicing algorithm. A third slice owner wishes to embed a URLLC slice. The controller decides that the current static slicing is not appropriate for this scenario and switches to a QoS-aware slicing algorithm which can express the latency requirements by this slice while keeping the SLAs of the two eMBB slices.
4) **Encapsulation of slices**. The interface between the slice algorithm and the slice packet schedulers allows a clean separation of concern with encapsulated state of each slice with respect to its users and resources. This creates network functions that can be added, removed and customized to correspond to slice owner needs without impacting the overall network.
5) **End-to-end slicing**. We added support to dynamically add and remove core network (CN) connections to OpenAirInterface through an S1 management module, highlighting the operational aspect of our solution. This allows to connect on demand to slice-specific dedicated CNs under the control of an orchestrator. Consider again Figure 1: the URLLC slice requires a CN with short packet handling delays. Before embedding the slice, the base station connects to CN_{LL} before adding the slice.

III. Demo Impact

Note that a number of contributions follow from this demo. The above architecture allows the research community to readily implement and evaluate algorithms for both slicing or packet scheduling [6] in a real base station, both to date rather difficult to implement in OpenAirInterface. Further, it is feasible to reimplement the Orion [7] architecture (of which the code has not been published) to functionally isolate slices from each other.

IV. Demo Setup

The demo will concentrate on the life cycle of a URLLC slice. Before adding the slice, the controller will connect the base station to an indicated CN and possibly dynamically change the slicing algorithm to a QoS-based one. Then, the URLLC slice is added in the base station. As soon as no users are active anymore, the outlined changes will be rolled back by the controller.

Users in the URLLC slice will experience an improved performance: their latency will be kept low, while an optimal system resource usage will be guaranteed. Visitors of the demo will be able to see this on a dashboard, tracing key performance indicators.

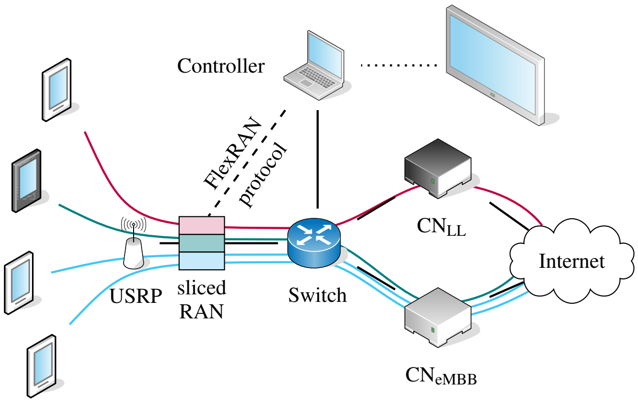

Fig. 2. Demo setup. An additional screen will show the slice life cycle and slice performance indicators.

We will use the demo setup illustrated in Fig. 2, consisting of LTE phones, an Ettus USRP B210-mini frontend with radio equipment, four computers and a switch. The USRP is connected to one computer running OpenAirInterface as the RAN with slicing capability and data plane functionality. Via the switch, the RAN is connected to two CNs, each being a standalone CN (consisting of HSS, MME, and S/P-GW) and connecting the end-to-end slices to the internet.

The demo is controlled from a laptop hosting the FlexRAN controller. The controller reconfigures the RAN (slicing and CN management) through the FlexRAN protocol. A dashboard on a separate screen allows the visitor to follow the demo.

Acknowledgment

This work received funding from the European Union's Horizon 2020 Framework Programme under grant agreement No. 762057 (5G-PICTURE), No. 761913 (SliceNet), and No. 857201 (5G-VICTORI).

References

[1] 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, Jun. 2017.
[2] R. Schmidt, C.-Y. Chang, and N. Nikaein. "Slice Scheduling with QoS-Guarantee Towards 5G", *IEEE Global Communications Conference (Globecom) 2019*, Waikoloa, HI, USA, December 2019.
[3] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A flexible platform for 5G research", in *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 33–38, Oct. 2014.
[4] R. Schmidt, C.-Y. Chang, and N. Nikaein, "FlexVRAN: A flexible controller for virtualized RAN over heterogeneous deployments, in *ICC 2019 - IEEE International Conference on Communications (ICC)*, Shanghai, China, May 2019, pp. 1–7.
[5] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A substrate for virtualizing wireless resources in cellular networks", *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1333–1346, Oct. 2012.
[6] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in LTE cellular networks: Key design issues and a survey", *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 678–700, Jul. 2013.
[7] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture", in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, ser. MobiCom 17*, Snowbird, Utah, USA: ACM, 2017, pp. 127–140.