Demo: A Network Slicing Solution for Flexible Resource Allocation in SDN-based WLANs

Estefanía Coronado*, Blas Gómez‡ and Roberto Riggio*
*Smart Networks and Services (SENSE). Fondazione Bruno Kessler, Trento, Italy
Email: {e.coronado, rriggio}@fbk.eu
‡High-Performance Networks and Architectures, University of Castilla-La Mancha, Albacete, Spain
Email: Blas.Gomez@uclm.es

***Abstract*—WLANs need to support an increasing number of applications each characterised by different requirements in terms of bitrate, latency, and reliability. Network slicing is an emerging trend that enables such a vision by creating virtual networks, i.e., the slices, that are tailored to specific requirements of the services. In this paper, we demonstrate a programmable SDN-based solution that enables customisable end-to-end slice definition encompassing both the wired and the wireless segments of an Enterprise WLAN. We release the entire implementation under a permissive license for academic use.**

***Index Terms*—Software-Defined Networking, Network Slicing, IEEE 802.11, Enterprise WLANs.**

I. Introduction

Services requirements in WLANs can differ in terms of bandwidth, latency, and bitrate. This proves that a single rigid network architecture is not enough to support such a variety, and calls for a service-oriented resource provisioning. To make this possible, the physical network must be abstracted into multiple *end-to-end* virtual networks or slices, for what Software-Defined Networking (SDN) is considered one of the most promising technological enablers.

In this paper we demonstrate a programmable end-to-end slicing solution for SDN-based WLANs [1]. This solution has three main features: (i) *programmability*, it allows programmers to specify how a portion of the flowspace shall be treated; (ii) *isolation*, it ensures that slices are isolated from the logical and the performance standpoints; and (iii) *customisation*, it allows defining specific traffic policies for each slice.

IEEE 802.11e [2] supports traffic prioritization, but not end-to-end network slices. Similarly, OpenFlow [3], the de-facto standard for SDN, does not encompass the wireless access segment. The solution showcased in this paper is implemented as an extension of the *5G-EmPOWER* Software-Defined Multi-access platform [4] to provide slicing capabilities over Wi-Fi networks. We release the entire implementation under a permissive APACHE 2.0 license for academic use[1].

II. End-to-end Network Slicing

The network slicing architecture builds upon a programmable hypervisor on top of a hardware abstraction layer that detaches the complexity of the Linux Wi-Fi stack. The hypervisor supervises the life-cicle of the slices and is responsible for their creation, monitoring and management. In this architecture, shown in Fig. 1, slices are defined as a set of network resources. Access Points (APs) can support a variable number of slices and each slice can handle several users.

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

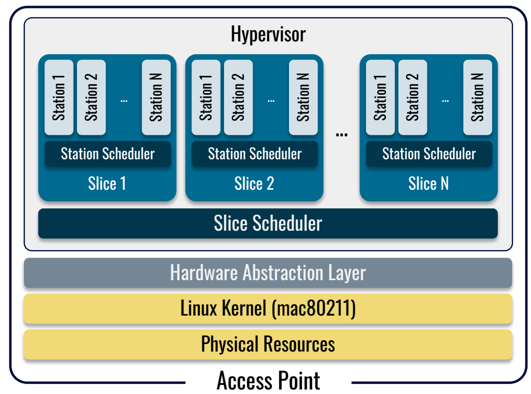

Fig. 1: High-level view of the architecture for network slicing.

A. The Traffic Rule Abstraction

A *Traffic Rule* specifies how an AP shall treat a particular *flow*. A *flow* refers to the traffic matching certain conditions as defined by the rules used in OpenFlow switches. Each rule is composed of a *Match* that identifies the flow, and an *Action* that specifies the operation to be performed on each packet. The *Match* includes a set of link, network, and transport header fields. Some examples are shown in Fig. 2.

In this work the *Action* performed on the frames fitting a *Match* is to tag them by using an OpenFlow switch, as shown in Fig. 3. For compatibility with the 802.11 frames, the tag is set in the IP DSCP field. However, this value is unrelated from the QoS associated to the DSCP, and is used as a simple identifier. Actually, other fields can be chosen for this matter. The tag unequivocally identifies a *Traffic Rule* in a Wi-Fi network (represented by an SSID) on a specific AP. Notice that an AP can deploy several SSIDs. The *Traffic Rules* are created by *5G-EmPOWER* and managed by a backhaul controller (in this work, the Ryu Controller), who applies the rules previously defined. Then, once tagged, the frames are dispatched to the hypervisor at the corresponding AP.

	Ethernet			IP				Transport		
	MAC src	MAC dst	...	IP src	IP dst	IP dscp	....	TCP/UDP sport	TCP/UDP dport	...
Rule 1	*	*	*	*	*	*	*	*	*	*
Rule 2	*	*	*	*	192.168.0.1	*	*	*	8080	*

Fig. 2: Examples of OpenFlow *match* fields.

Network slices are instantiated by *5G-EmPOWER* identified by a tuple `(SSID-tag)`. This allows frames reaching the APs to be mapped to the corresponding slice. Slices are characterised by a set of parameters that determine the treatment given to the traffic in each slice. Such parameters include:

- *EDCA*. The 802.11e EDCA parameters, including Congestion Window, Arbitration Inter-Frame Space, and Transmit Opportunities.
- *Aggregation*. The type of frame aggregation to be used, including A-MSDU, A-MPDU, or none.
- *Quantum*. The airtime (in ms) assigned in each round.

B. Network Slices Scheduling

The hypervisor coordinates the slices to perform network-wide resource scheduling based on their requirements. Moreover, each slice can handle its users differently depending on the services provided. In this respect, different scheduling policies can be configured at both levels. In this work we have used a policy named Airtime Deficit Weighted Round Robin (ADWRR) [5]. In each round the slices are assigned a fraction of airtime according to their *quantum*, since the *cost* of transmitting a frame depends on its length *and* the channel conditions of the receiver. For example, a receiver far from the AP may utilize more resources due to the use of less efficient MCSes and/or more frame retransmissions. The hypervisor leverages the rate selection statistics of the APs to estimate the required airtime. Any algorithm can be used as long as it includes for each client the link delivery probability for each MCS. In this case, we rely on Minstrel [6].

III. Demonstration

In this demo we illustrate the isolation capabilities enabled by the slicing solution and how the slice descriptors are respected regardless of the traffic transmitted. The demo shows two scenarios in which two users with the same channel quality are attached to two slices, as depicted in Fig. 3. The *station 1* located in *Slice 1* is playing a Youtube video, while the *station 2* in *Slice 2* is downloading UDP traffic using iperf.

First, *Scenario A* shows a deployment where the slices have the same priority (the airtime is equally divided dedicating $10k$ ms per slice). In the station playing the Youtube video, we enable the automatic Youtube quality (so that the resolution is chosen by the application according to the network quality), and the so-called *"stats for nerds"* to check the video resolution at each moment. After that, in *Scenario B* the proportion of airtime between the slices is modified in real-time by *5G-EmPOWER* to $0.5k - 10k\ ms$, leaving the smallest part for *Slice 1*. Upon this change, and once the Youtube buffer is empty, it can be seen that the resolution displayed has decreased with respect to the first scenario, while the amount of data downloaded by the user in *Slice 2* has greatly increased.

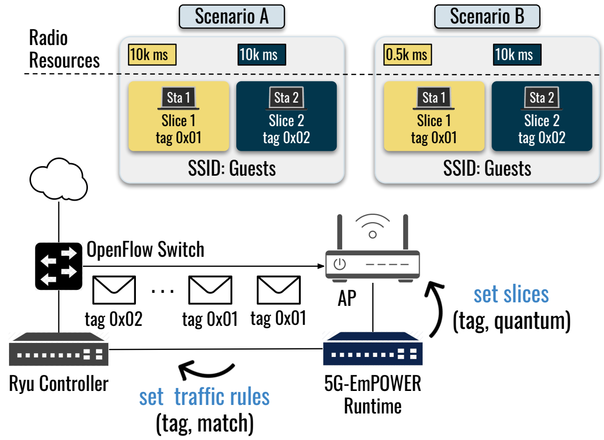

Fig. 3: Network deployment used for the demonstration.

Through this demonstration, we showcase the flexibility of the service-based slicing solution, which enables advanced QoS management, customised configuration policies, as well as their application in real-time in a seamless manner.

IV. Demo Setup Requirements

The equipment for the demo consists of one off-the-shelf Wi-Fi Access Point, three laptops and one Ethernet switch. The setup will fit on a regular office table with roughly two square meter space. The setup time is less than two hours. A regular power socket will also be required.

Acknowledgements

This work has been supported by the Spanish Regional Government of Castilla-La Mancha under the project SB-PLY/17/180501/000353, the Spanish Ministerio de Ciencia, Innovación y Universidades and the European Commission (FEDER funds) under the project RTI2018-098156-B-C52.

References

[1] E. Coronado, R. Riggio, J. Villalón, and A. Garrido, "Lasagna: Programming Abstractions for End-to-End Slicing in Software-Defined WLANs," in *Proc. of IEEE WoWMoM*, Chania, Greece, 2018.

[2] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 7: Medium Access Control (MAC) Quality of Service (QoS)*, ANSI/IEEE Std 802.11e, LAN/MAN Standards Committee of the IEEE Computer Society Std., 2005.

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[4] E. Coronado, S. N. Khan, and R. Riggio, "5G-EmPOWER: A Software-Defined Networking Platform for 5G Radio Access Networks," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 715–728, Apr. 2019.

[5] R. Riggio, D. Miorandi, and I. Chlamtac, "Airtime Deficit Round Robin Packet Scheduling Algorithm," in *Proc. IEEE MASS*, Atlanta, Georgia, 2008.

[6] D. Xia, J. Hart, and Q. Fu, "Evaluation of the Minstrel rate adaptation algorithm in IEEE 802.11g WLANs," in *Proc. of IEEE ICC*, Budapest, Hungary, 2013.