

# An SDN-based framework for Slice Orchestration using In-Band Network Telemetry in IEEE 802.11

Pedro Heleno Isolani\*, Jetmir Haxhibeqiri<sup>†</sup>, Ingrid Moerman<sup>†</sup>, Jeroen Hoebeke<sup>†</sup>, Johann M. Marquez-Barja\*, Lisandro Z. Granville<sup>‡</sup>, and Steven Latré\*

\* University of Antwerp - imec, IDLab - Department of Mathematics and Computer Science, Department of Electronics-ICT  
E-mail: {pedro.isolani, johann.marquez-barja, steven.latre}@uantwerpen.be

<sup>†</sup> University of Ghent - imec, IDLab - Department of Information Technology  
E-mail: {jetmir.haxhibeqiri, ingrid.moerman, jeroen.hoebeke}@ugent.be

<sup>‡</sup> Federal University of Rio Grande do Sul - Institute of Informatics, Computer Networks Group  
E-mail: {granville}@inf.ufrgs.br

**Abstract**—The fifth generation of mobile networks (5G) and the Software-Defined Radio Access Networks (SD-RAN) architecture envision to support lower latency, enhanced reliability, massive connectivity, and improved energy efficiency. In this context, low latency is considered crucial and Ultra-Reliable Low Latency Communication (URLLC) as one of the key enablers. Currently, IEEE 802.11 networks cannot be programmed fine-grained enough nor manage multiple networks at runtime. Besides, in such scenarios, the coarse-grained level of monitoring information has been hindering troubleshooting and management. In this paper, we present an SDN-based framework where fine-grained End-to-End (E2E) network statistics can be gathered using In-band Network Telemetry (INT) and used for network control and management. With such fine-grained network information, we show how our system can enhance the Quality of Service (QoS) delivery through slice orchestration in IEEE 802.11 Radio Access Networks (RANs).

## I. INTRODUCTION

Nowadays, different services and applications often have to share the same wireless infrastructure and Industrial, Scientific, and Medical (ISM) radio bands, making it very challenging to meet diverging Service Level Agreements (SLAs). The fifth generation of mobile networks (5G) envisions Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communication (URLLC), and Massive Machine Type Communications (mMTC) to be deployed in the coming years. In this context, detecting performance degradation is yet challenging. Although Software-Defined Networking (SDN) simplifies or even solves critical management activities by itself, the instability caused by the presence of wireless interference and physical obstacles has hampered the management of such networks. Therefore, there is a need for programmable and fine-grained monitoring features to enable better control and understanding regarding network behavior.

Many monitoring techniques have been addressed for wireless networks, such as End-to-End (E2E) active probing and periodic statistics collection from network devices. However, given the complexity and dynamic wireless environment, those techniques result in high overhead and lack of flexibility over the data collection. Therefore, a novel monitoring technique,

known as In-band Network Telemetry (INT), has been applied to address flexible and more granular monitoring about the E2E flow dynamics, state of the wireless links, and per-hop reliability [1]. With such fine-grained information of the network, management systems might benefit from it to enhance the Quality of Service (QoS) delivery.

In addition to monitoring, many proposals have been addressing network slicing as a mean for providing precise networking resources in IEEE 802.11 Radio Access Networks (RANs) [2]. Besides operating independently from one another, network slices provide precise networking resources and traffic isolation among users and services [3]. The fact that network slices might be instantiated, modified, or terminated dynamically implies a strong decoupling between software-based functions and underlying network infrastructure. Therefore, SDN-tailored systems are envisioned to ease the creation of logical and isolated networks via slice abstractions [4]. In this context, E2E latency, accessibility, and throughput requirements have to be addressed simultaneously [5]. Therefore, in this paper, we present an SDN-based framework for slice orchestration using INT monitoring techniques. In summary, our framework is capable of:

- Fine-grained statistics gathering via INT-enabled nodes;
- Statistics monitoring with configurable polling intervals;
- Statistics data processing via an SDN controller.
- Periodic statistics analysis against current application QoS requirements;
- Dynamic slice orchestration and Medium Access Control (MAC) reconfiguration in IEEE 802.11 RANs.

## II. SYSTEM OVERVIEW

In this work, we consider a scenario where network slices are defined by a centralized entity, in SDN-enabled 5G network infrastructure. Therefore, this entity has the view and control over all network resources. In this scenario, multiple *tenants* (i.e., virtual operators or service providers) share the same infrastructure and have their specific SLA. These SLAs are translated into QoS requirements which the network has

to support (e.g., minimum throughput, maximum allowed E2E latency, and acceptable packet loss ratio). To meet such requirements, we propose the use of network slicing along with SDN centralized management. Figure 1 illustrates the SDN architecture along with the components of our framework.

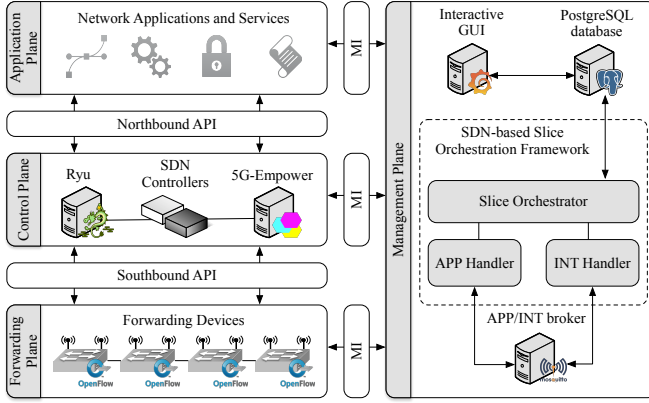


Fig. 1. SDN architecture along with the components of our framework.

To perform network slicing at the control and forwarding planes, our framework relies on the 5G-EMPOWER platform<sup>1</sup>, including a Software-Defined Radio Access Networks (SD-RAN) controller, a backhaul controller implementation of the SDN controller Ryu, and an agent that runs at the RAN itself. To gather and maintain INT measurements, several extensions had to be implemented not only at the control but also in the forwarding plane. In this case, Station (STA) and Access Points (APs) are prepared to handle and append monitor data into INT-enabled packets. To maintain the monitoring information gathered over time and provide it to the interactive Graphical User Interface (GUI), we added an instance of an MQ Telemetry Transport (MQTT) broker<sup>2</sup> and a PostgreSQL database<sup>3</sup>. To visualize the INT measurements at runtime, we built an interactive GUI using Grafana<sup>4</sup>. At the *management plane*, we included three main components to enable management via slice orchestration: *Slice Orchestrator*, *Application Handler*, and *INT Handler*.

#### A. Application Handler

To share the application requirements between the end devices, RAN, and the management system, two Application Programming Interfaces (APIs) are implemented. The first enables the communication between the application layer and the INT layer in the same device. The API is used to instruct the INT layer regarding the data flow to be monitored and the INT parameters. The parameters include namespace identifier, periodicity, application identifier related to the source and destination IP addresses and ports, and transport protocol. On the other hand, the second API is used for transmitting the

application requirements from the INT-enabled sink node to the management system. To achieve this, this API is implemented based on a lightweight open-source message broker that implements the MQTT protocol, called APP broker. In this manner, the application requirements and the INT layer parameters are known to both source and destination nodes as well as to the control and management components.

#### B. INT Handler

INT is a novel monitoring technique that overcomes several problems of typical monitoring techniques. In most cases, there is a coarse-grained level of the monitoring information that hinders troubleshooting and management. With INT, it is possible to gather per-flow, E2E, and per-link fine-grained monitoring information down to a packet-level. In addition to wired networks, we have extended INT to wireless networks as well [1]. The INT layer was implemented using the Click modular router<sup>5</sup> and can behave as a source, intermediate, or sink node. The source and sink nodes communicate with the *Application Handler* via an API that enables gathering information of the flows and setting the monitoring granularity (e.g., one INT-enabled packet every second). Notwithstanding, as the intermediate node only processes the INT header and appends the required monitoring data, the node does not need an API towards the *Application Handler*. At the end of each flow, sink nodes communicate with *INT handler* using an MQTT broker, called INT broker. In this manner, the monitored information is available for all other components.

#### C. Slice Orchestrator

This component is responsible for defining how network applications are mapped into different network slices. To this end, two steps are considered: *flow handling* and *slice airtime reconfiguration*.

1) *Flow Handling*: First of all, flows belonging to all network applications have to be identified and dedicated and isolated slices have to be created when needed. To do such a thing, this component accesses flow information provided by the *APP handler* component. This information includes the application identifier, transport protocol, source and destination IP addresses and ports in addition to the application requirements such as maximum allowed jitter and E2E latency. Then, the *Slice Orchestrator* component sends instructions via the backhaul controller and SD-RAN controller to install OpenFlow rules on the OpenFlow-enabled devices, i.e., APs running instances of the OpenvSwitch. Thus, traffic rules define which frames belong to which slices of the network.

2) *Slice Airtime Reconfiguration*: With the monitoring information maintained by the *INT handler* and the application requirements handled by the *APP handler*, reconfiguration of the network slices might be necessary. In this step, calculated metrics from all the applications are compared against the required values for the QoS. For instance, the maximum allowed jitter is compared with the calculated jitter from

<sup>1</sup><https://github.com/5g-empower/5g-empower.github.io>

<sup>2</sup><https://mosquitto.org/>

<sup>3</sup><https://www.postgresql.org/>

<sup>4</sup><https://grafana.com/>

<sup>5</sup><https://github.com/kohler/click>

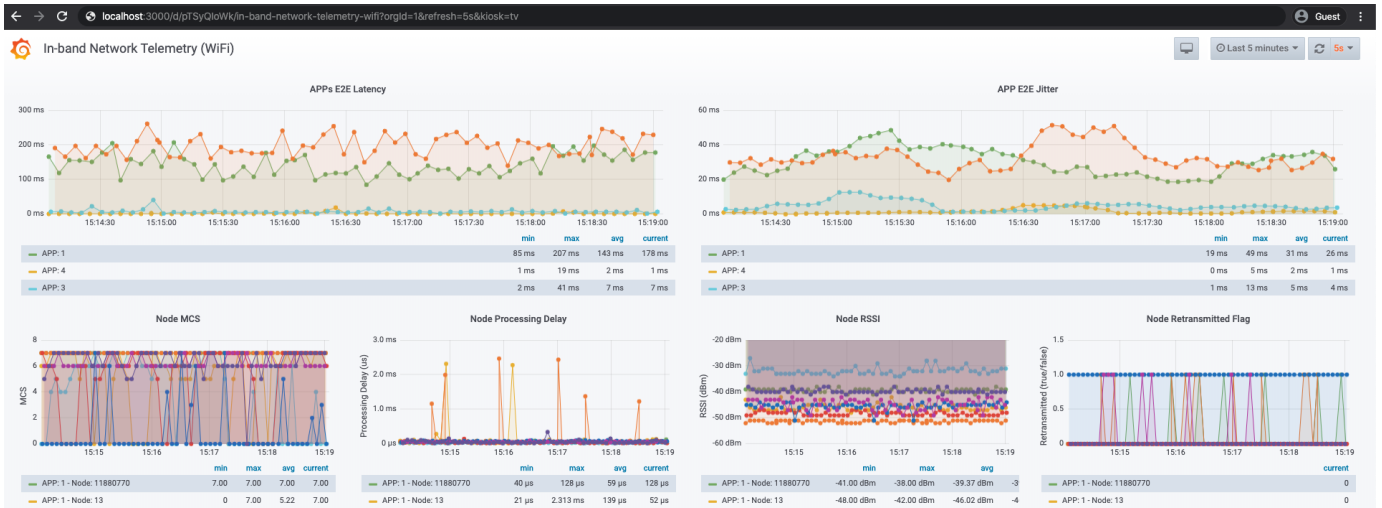


Fig. 2. Interactive GUI screenshot.

the E2E jitter of the flow or the maximum allowed E2E latency of an application is compared against the median value of the last 10 second-measurements. In cases where QoS requirements are not met, slices are reconfigured in such a way that flows belonging to the QoS-restricted applications receive larger portions of airtime than others. Otherwise, resources are gradually released so the Best Effort (BE) flows can utilize the remaining resources. To achieve this, our framework triggers slice airtime MAC reconfigurations towards the RAN via the SD-RAN controller. Therefore, delay-sensitive requirements can be achieved and visualized in our interactive GUI.

### III. DEMONSTRATION

In this demo, an SDN-enabled 5G network infrastructure is built in the Industrial IoT lab<sup>6</sup> located in Ghent, Belgium. Three APs are installed to compose the IEEE 802.11 RAN and nine Intel NUC nodes, equipped with IEEE 802.11n cards, are used as STAs. A single computer, connected to the wired segment, will be used to run all the centralized controllers and management components. There will be at least 4 different flows on the network: two of them running between two nodes connected to the same AP, while another two flows will be distributed among the other APs. All flow information collected using INT as well as application requirements will be visualized in the *Interactive GUI* (Figure 2).

The dashboard comprises two categories of time series graphs: E2E application information such as latency and jitter, and link related information such as Received Signal Strength Indicator (RSSI), channel, re-transmission flag, Modulation and Coding Scheme (MCS), and data rate. In this manner, visitors will be able to check whether application requirements are met in addition to the impact of network slice orchestration.

Therefore, we first show how applications share their QoS requirements towards the *Application Handler*. Then, we will

show that using INT different per-link and per-flow information can be collected using different monitoring granularities according to the flow importance. Besides, visitors will be able to start and stop different flows and publish different application requirements to the network remotely. Next, we will show how airtime allocation via slice orchestration impacts the monitored metrics at runtime. We will focus on achieving time-sensitive requirements such as maximum E2E latency and jitter. In summary, this demo focuses on the insights on how INT-enabled monitoring supports troubleshooting and management and showing the impact of slice airtime allocation on time-sensitive metrics such as E2E latency and jitter.

### ACKNOWLEDGMENT

This research received partial funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” program, from the FWO-Flanders, under grant agreement #G055619N, and from the European Union’s Horizon 2020 Research and innovation program, under grant agreement No. 826284 (ProTego).

### REFERENCES

- [1] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, “Low overhead, fine-grained end-to-end monitoring of wireless networks using in-band telemetry,” in *CNSM2019, the 15th International Conference on Network and Service Management*, 2019, pp. 1–5.
- [2] M. Richart, J. Baliosian, J. Serrat, and J. Gorricho, “Resource slicing in virtual wireless networks: A survey,” *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, Sep. 2016.
- [3] E. Coronado, R. Riggio, J. Villalón, and A. Garrido, “Lasagna: Programming abstractions for end-to-end slicing in software-defined wans,” in *2018 IEEE 19th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2018, pp. 14–15.
- [4] E. Coronado, S. N. Khan, and R. Riggio, “5g-empower: A software-defined networking platform for 5g radio access networks,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 715–728, June 2019.
- [5] P. H. Isolani, N. Cardona, C. Donato, J. Marquez-Barja, L. Z. Granville, and S. Latré, “Sdn-based slice orchestration and mac management for qos delivery in ieee 802.11 networks,” in *2019 Sixth International Conference on Software Defined Systems (SDS)*, June 2019, pp. 260–265.

<sup>6</sup><https://www.ugent.be/ea/idlab/en/research/research-infrastructure/industrial-iot-lab.htm>