

# End-to-end network slicing testbed with the COMETS platform

Youssef Drif<sup>\*†</sup>, Emmanuel Lavinal<sup>†</sup>, Emmanuel Chaput<sup>†</sup>,  
Pascal Berthou<sup>‡</sup>, Boris Tiomela Jou<sup>§</sup>, Olivier Grémillet<sup>\*</sup>, Fabrice Arnal<sup>\*</sup>

<sup>\*</sup>IRT Saint-Exupéry, France; Email: firstname.lastname@irt-saintexupery.com

<sup>†</sup>IRIT, Toulouse, France; Email: firstname.lastname@irit.fr

<sup>‡</sup>LAAS, Toulouse, France; Email: firstname.lastname@laas.fr

<sup>§</sup>Airbus Defence and Space, Toulouse, France; Email: firstname.lastname@airbus.com

**Abstract**—The integration of Non Terrestrial Networks (NTNs) in 5G networks is the result of many recent research results. This has led us to actively work on a deeper satellite backhaul integration based on network slicing. To evaluate the relevance of our work, we have developed a flexible container-based network slicing platform (COMETS) which is able to execute custom scenarios and evaluate their performances. In this demonstration, we present an end-to-end network slicing scenario where we deploy multiple 5G slices with a satellite network integrated as a slice-aware backhaul link. We focus on the respect of end-to-end slice Key Performance Indicators (KPIs) and Quality of Service (QoS) within each slice. In the COntainer MEsh for nETwork Slicing (COMETS) platform we have integrated multiple existing simulation tools, extended the functionalities of the 5G Core Network (CN) and developed our own tools which fully automate the deployment, execution and performance evaluation of the slices specified in the scenario’s description.

**Index Terms**—5G, Non Terrestrial Network, Network Slicing, Network Function Virtualization (NFV)

## I. END-TO-END NETWORK SLICING

THE Release 17 of the 3GPP specifications started to integrate satellite networks as NTNs. The upcoming releases are expected to improve and bring more seamless integration. In this context, we have worked on satellite integration in 5G networks as an enhanced backhaul link using network slicing. Network slicing is a paradigm which partitions the network into optimized end-to-end “slices” for specific network services. As defined in the 5G specification, a network slice is the composition of multiple network slice subnets, each usually belonging to a specific network domain [1]. Network slices can be characterized by multiple attributes such as the isolation level, the security level or the resource allocation. However, the main challenge faced by this paradigm is the slice continuity across the various network domains and also the QoS continuity within each slice. Indeed, the proposed integration schemes until now do not permit to achieve such an end-to-end integration. Therefore, in [2] we have focused on end-to-end slicing and QoS continuity across network domains. Our integration model and architecture are based on the existing 3GPP [1], [3]–[7] specifications to be fully compatible with the 5G standard. In our approach, we consider the end-to-end slice as the composition of the Radio Access Network (RAN) network slice subnet, our defined “slice-aware” NTN link and the CN network slice subnet as shown in Figure 1. In order

to have slice and QoS continuity, the NTN control and 5G networks must be interconnected. Each flow within the 5G network must be fully identified within the satellite network. Thus, we have introduced two Network Functions (NFs): the 5G and the NTN Quality of Service Functions (QOFs) deployed in the 5G CN and satellite management system respectively, that communicate over the *ntnqof* interface. When a PDU session is created/modified/deleted within the 5G network, the 5G QOF sends all the related PDU information to the NTN QOF. Then, the NTN QOF populates the slice and QoS tables of each Slice Classifier. It programs the Classifier using the *sc* interface. Slice Classifiers are the interconnection components we have introduced between the 5G RAN, the NTN and the 5G CN. These components are QoS aware, they ensure slice isolation, as well as slice and QoS continuity by translating flows between domains.

This demo paper showcases the work presented in [2] by exposing the COMETS platform that we have developed to evaluate multiple slicing scenarios. This flexible platform is container-based and allows to define custom 5G-Satellite scenarios and experiment them. It is extensible and can integrate other tools as long as they can be containerized. Figure 2 shows the COMETS architecture and the steps it follows to evaluate network slicing scenarios.

## II. PLATFORM IMPLEMENTATION

We have developed an automated platform which is able to deploy customized test scenarios [8]. Based on a scenario definition file, COMETS generates the corresponding container-based testbed with all its requirements for the scenario performance evaluation. The testbed relies on Docker and Docker Compose to run and orchestrate all the containers. Our 5G CN is based on free5GC [9] as it is fully open source. We have altered the Session Management Function (SMF) and added our 5G QOF [10]. The NTN QOF is also based on the existing framework developed by the free5GC team to easily implement any container-based NF. We added a hook function into the SMF which is triggered every time a PDU session is created, modified or deleted. The User Equipments (UEs) and the gNB are based on UERANSIM [11], an open source Release 17 compliant 5G RAN. We have used Trunks [12], a lightweight DVB-S2/RCS2 satellite link simulator

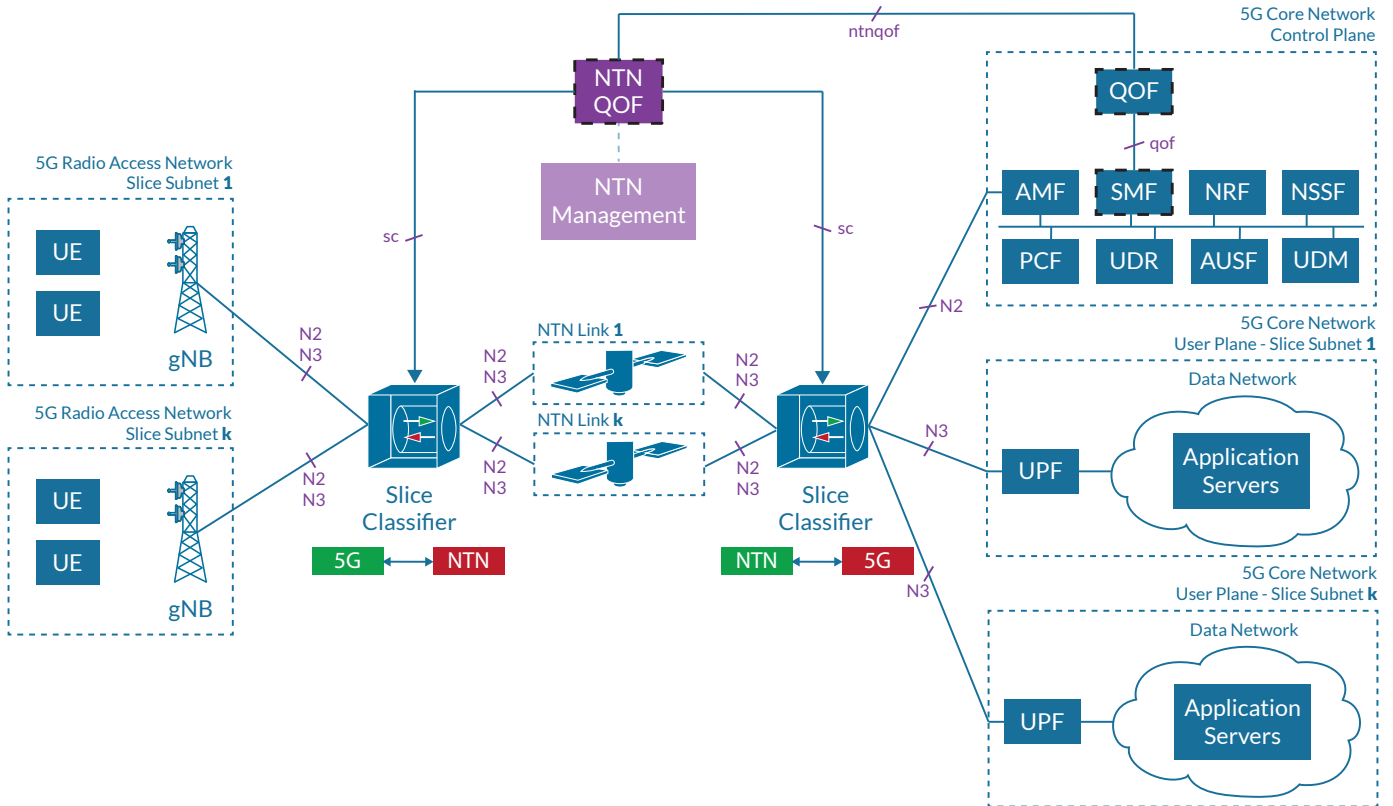


Fig. 1. Functional Architecture

developed during our research, based on the Linux tc tool [13] to simulate Low Earth Orbit (LEO) and Geostationary Orbit (GEO) satellite systems. We then generate traffic using the IPerf tool [14] with different traffic characteristics for each type of application that we support: Web, Streaming and VoIP. IPerf clients are deployed on each UE and they connect to their IPerf server in the data network behind each User Plane Function (UPF). Each IPerf server is dedicated to a UE and exposes an application. The IPerf probes are then automatically parsed to generate plots for further analysis.

### III. DEMONSTRATION SCENARIO

In this demonstration, we compare two scenarios: the “slice-unaware” (traditional satellite backhaul) integration and the “slice-aware” approach. For each scenario, we generate three end-to-end slices with different characteristics and evaluate their performances. The goal of this demonstration is to evaluate metrics related to the 5G network slicing and evaluate the relevance of our integration method. Unlike the “slice-aware” mode, in the “slice-unaware” scenario, all the mechanisms we have defined in the control and data plane to ensure the end-to-end slicing are disabled. Thus, Slice Classifiers are replaced by Broadband Network Gateways (BNGs) and no changes are made to the 5G CN. Figure 3 shows a sample of the configuration file used in this demonstration. Within this file, we define the parameters for both scenarios which include the number of UEs to simulate, the number of NTN links and

their physical characteristics, the list of slices which run within each link, their related requirements as well as the applications which run within each slice. Through the demonstration, we will proceed as follows:

- Show the architecture platform and the software framework of COMETS. We then explain in detail how to customize slicing scenarios;
- Inspect the scenario configuration file and detail each parameter of both scenarios;
- Generate the testbed of each scenario composed of the UEs, the 5G RAN, the satellite network, the 5G CN and the application servers;
- Run testbeds associated to each scenario and generate probes;
- Analyze the probes results and plot Packet Delay Budget (PDB), throughput, Packet Error Rate (PER) and the Jitter of each slice and 5G QoS classes.

Each scenario generates a testbed of a total of 64 containers. As the platform is open-source, every attendee will be able to reproduce the demonstration and even play its own scenario.

### REFERENCES

- [1] 3GPP, “Management and orchestration; Concepts, use cases and requirements;” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 28.530, 12 2020, version 17.0.0. [Online]. Available: <http://www.3gpp.org/DynaReport/28530.htm>
- [2] Y. Drif, E. Lavinal, E. Chaput, P. Berthou, B. Tiomela Jou, O. Gremillet, and F. Arnal, “Slice Aware Non Terrestrial Networks;” in *2021 46th IEEE Conference on Local Computer Networks (LCN)*, 2021.

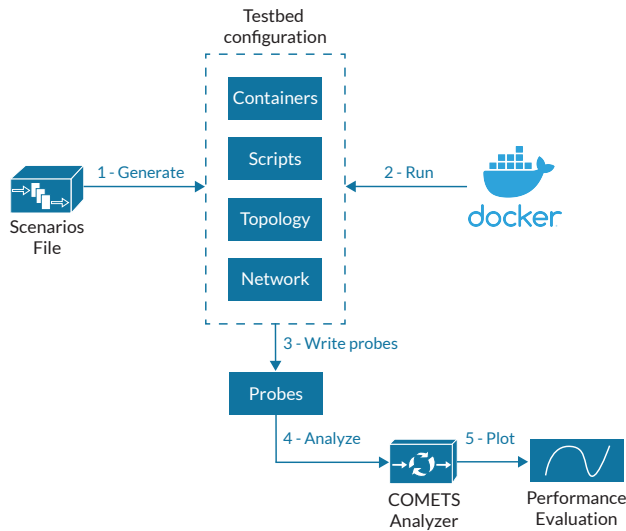


Fig. 2. COMETS Architecture

- [3] 3GPP, "System architecture for the 5G System (5GS)," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.501, 12 2020, version 16.7.0. [Online]. Available: <http://www.3gpp.org/DynaReport/23501.htm>
- [4] 3GPP, "NR; NR and NG-RAN Overall description; Stage-2," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.300, 1 2021, version 16.4.0. [Online]. Available: <http://www.3gpp.org/DynaReport/38300.htm>
- [5] 3GPP, "Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 28.541, 12 2020, version 17.1.0. [Online]. Available: <http://www.3gpp.org/DynaReport/28541.htm>
- [6] GSMA Association, "Generic Network Slice Template," GSMA, Specification NG. 116, November 2020, version 4.0. [Online]. Available: <http://www.ietf.org/internet-drafts/draft-henry-tsvwg-diffserv-to-qci-04.txt>
- [7] 3GPP, "Management and orchestration; 5G end to end Key Performance Indicators (KPI)," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 28.554, 12 2020, version 17.1.1. [Online]. Available: <http://www.3gpp.org/DynaReport/28554.htm>
- [8] Youssouf Drif and others, "Slice Aware Non Terrestrial Networks," 2021. [Online]. Available: <https://github.com/shynuu/slice-aware-ntn>
- [9] Free5GC Consortium, "free5GC R15," 2021, V3.0.5. [Online]. Available: <https://www.free5gc.org/>
- [10] Youssouf Drif and others, "5G Core NTN," 2021. [Online]. Available: <https://github.com/shynuu/5g-core-ntn>
- [11] Ali Güngör, "UERANSIM," 2021, V3.1.9. [Online]. Available: <https://github.com/aligungr/UERANSIM>
- [12] Youssouf Drif, "Trunks - Satellite Link Simulator," 2021, V2.0.0. [Online]. Available: <https://github.com/shynuu/trunks>
- [13] Linux, "Linux Traffic Control," 2021. [Online]. Available: <https://man7.org/linux/man-pages/man8/tc.8.html>
- [14] Robert McMahon, "IPerf2," 2021. [Online]. Available: <https://sourceforge.net/projects/iperf2/>

```

1  saw-ntn: # Slice Aware Scenario
2  type: "slice-aware"
3  duration: 240
4  user: 9 # Number of UE to simulate
5  links: # NTN Link list
6  - default: false # NTN Link parameters
7  forward: 100
8  return: 25
9  delay: 45
10 jitter: 5
11 acm: false
12 slices: # Slice within the NTN Link
13 - start: 0
14   end: 240
15   theta: # 5G requested parameters
16     lambda: 45
17     delta: 5
18     mu: 10
19     beta: 35
20     sigma: true
21   applications: # Applications in the
22     ↪ slice
23     - name: voip
24       data_rate: 128 # kbps
25     - name: web
26       data_rate: 3 # mbps
27 [...]
28 suaw-ntn: # Slice Unaware Scenario
29 type: "non-slice-aware"
30 duration: 240
31 user: 9
32 links: # NTN Link list
33 - default: false
34 forward: 100
35 return: 25
36 delay: 45
37 jitter: 5
38 acm: false
39 slices: # Slice within the NTN Link
40 - start: 0
41   end: 240
42   theta:
43     lambda: 45
44     delta: 5
45     mu: 10
46     beta: 35
47     sigma: true
48 applications:
49 - name: voip
50   data_rate: 128
51 - name: web
52   data_rate: 3

```

Fig. 3. Sample of the scenarios configuration