Scientific Research Publishing

Journal of Computer and Communications, 2025, 13(7), 295-318
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Low-Cost Experimental 5G System to Foster Digital Transformation in 4G Uncovered Areas

Mahamadou Diawara[1], Andre Faye[2*]

[1]UFR de Sciences Appliquées et de Technologie, Gaston Berger University, Saint-Louis, Senegal
[2]Institut Polytechnique de Saint-Louis (IPSL), Gaston Berger University, Saint-Louis, Senegal
Email: diawara.mahamadou@ugb.edu.sn, *andre.faye@ugb.edu.sn

How to cite this paper: Diawara, M. and Faye, A. (2025) A Low-Cost Experimental 5G System to Foster Digital Transformation in 4G Uncovered Areas. *Journal of Computer and Communications*, **13**, 295-318.
https://doi.org/10.4236/jcc.2025.137015

Received: June 20, 2025
Accepted: July 26, 2025
Published: July 29, 2025

Abstract

The digital divide resulting from low network coverage, with disparities between urban and rural areas, is an obstacle to economic, healthcare, and education activities in sub-Saharan countries. Addressing the growing demand for connectivity with reliable, low-latency, and high-bandwidth network access is an urgent need to promote economic growth in these regions. However, broadening network coverage with reliable and sufficient network resources to satisfy the demand is not always economically viable for mobile network operators. As such, making the leap from 3G to standalone 5G through private initiatives could be the right approach to enable rural communities to access connectivity-based services such as distance education and remote medical care, despite the cost of 5G deployment. Thus, low-cost private networks capable of delivering customized solutions can help bridge the digital divide, and 5G SA presents a great opportunity to exploit. It is therefore important to set up testbeds to deploy 5G SA networks to study and test use cases, particularly in the absence of a 4G LTE network. Network slicing has emerged as a solution to fully exploit the potential of 5G networks, especially with regard to flexibility and programmability, as well as resource and infrastructure sharing. The network slicing approach enables the creation of multiple logical networks, each dedicated to a particular service or specific application within a single physical infrastructure. We present a low-cost end-to-end 5G network slicing testbed for private networks and performance tests with respect to energy consumption, the ability to cover the main development axes of 5G, and common user needs such as VoNR, video streaming, and drone-based network use cases for IoT applications in areas lacking network coverage.

Keywords

5G SA, End-to-End Network Slicing, IoT, Private Network

1. Introduction

The increasing demand for advanced mobile services, such as video streaming, VoIP, augmented reality (AR), and virtual reality (VR), as well as progress in IoT, has spurred global initiatives to develop 5G [1] [2], marking a new era of faster and more advanced mobile connectivity.

The growing demand for mobile connectivity, coupled with the proliferation of smart devices and the explosion of bandwidth-intensive applications such as live video streaming and online gaming, has highlighted the challenges that 5G must address, notably due to the heavy use of IP video traffic and the continued increase of mobile subscribers worldwide. Indeed, according to Cisco data [3], IP video traffic now represents a significant portion of consumer Internet traffic (82% in 2022), with an even larger share on mobile networks (78% in 2022).

This trend coincides with an increase in the number of mobile subscribers worldwide, reaching 5.7 billion in 2023, accounting for 71% of the population, up from 66% in 2018 [4]. The 3GPP, in report 3GPP TR 22.891 release 14, undertook a study within the SA 1 services working group to explore business opportunities offered by 5G and identifying over 70 potential use cases. These cases were grouped into three main categories: eMBB, mMTC, and URLLC. eMBB delivers ultra-high throughput and capacity to support data-intensive applications, such as ultra HD video streaming and virtual reality. mMTC focuses on large-scale connectivity of IoT devices, while URLLC aims to provide extremely reliable and ultra-low latency communications for real-time critical applications.

The 5G addresses a growing demand for connectivity in an increasingly connected society with smart networks and smart cities. Applications range from telemedicine to education sectors. Furthermore, it offers promising prospects in areas such as drone networks, especially for interconnecting IoT applications, thus offering significant potential for development in Africa. Drones can be integrated into a network as base stations, relays, or flying terminal nodes. For instance, the use of drones equipped with intelligent sensors and cameras to connect IoT applications offers significant advantages, especially in terms of surveillance in challenging environments and absence of infrastructure.

Both industry and academia view 5G as the foundation of future networks capable of meeting a diversity of performance and service requirements. However, questions remain regarding how to fully exploit its potential, especially with regard to flexibility and programmability. In this context, softwarisation and virtualisation are promising software solutions to reveal this potential as they made it possible to implement easy resource and infrastructure sharing.

The concept of Network Slicing (NS) emerged as a potential solution capable

to address these questions and realise the 5G vision [5] [6]. It is defined by the 3GPP in its technical report 3GPP TR 23.799 release 14 as a technology enabling customised networks for diverse solutions with various requirements. The NS approach brings considerable value by addressing the specific needs for market segmentation, application providers and third-party actors who do not have their own network infrastructure. It provides resource sharing and allows deep customisation by offering increased operational flexibility, optimal resource management, and quality of service (QoS) tailored to the specific needs of each segment, thereby enhancing the overall efficiency of the network. This guaranteed and configurable QoS paves the way for numerous new markets and offers many innovative and diversified use cases. For example, telemedicine and autonomous vehicle applications require ultra-high-definition video streaming via a mobile broadband channel with ultra-low latency [7]. Additionally, the resources and processing allocated to a slice depends on the associated service. For instance, a network slice dedicated to real-time communications will receive the resources and processing necessary to meet ultra-low latency requirements. This approach is crucial for realising the potential of 5G and opening up new markets estimated at $300 billion by 2025 according to GSMA report [8]. Despite these advancements, research questions persist regarding the effective implementation of 5G NS. This stimulates global initiatives from governments, academia, and major industries to provide innovative solutions to address these challenges. In the context of environments with limited resources, deploying 5G solution is tricky. Large scale testbeds are out of price and strategies are needed to build uses cases with affordable testbeds yet with sufficient performances.

In this article, we investigate the deployment, performance evaluation and energy consumption of a 5G end-to-end network slicing with a low cost testbed applicable to drone networks for IoT applications in our endeavour to set up a 5G use case in MNO's networks uncovered areas. The remainder of this work is organised as follows. Section 2 addresses related work and open questions. In Section 3, we provide a general overview of 5G network slicing. Section 4 describes our system, scenario, and testbed. Section 5 is dedicated to the evaluation of the system's performances and the discussion of the obtained results. Energy consumption is analyzed in Section 6 and the entire system validation using applications that meet common users' needs is presented in Section 7. Finally, we conclude.

2. Related Works

In past years, several researchers have delved into the exploration of network slicing [7] [9]-[13]. These works have provided in-depth analysis of 5G NS, highlighting key concepts, underlying technologies such as SDN (Software-Defined Networking) and NFV (Network Function Virtualisation), as well as implementation challenges. In [14], Sajjad *et al.*, focused on mobility management between 5G network slices, future research directions to ensure seamless mobility and the potential of data analytics and artificial intelligence techniques to facilitate seamless

mobility between slices. Chahbar *et al.* [15] presented a review of network slice modelling studies addressing radio access, core networks, and transport networks. The integration of IETF data models into the ACTN (Abstraction and Control of Traffic Engineered (TE) Networks) architecture is discussed. Regarding security inherent to network slicing, Salahdine *et al.* [16] proposed isolation and artificial intelligence-based solutions to enhance protection against potential attacks. Progresses in NS in the contexts of IoT and intelligent transportation are presented by Khan *et al.* [17] and by Wu *et al.* [6]. They identified pending research challenges including mobility-aware slicing, dynamic spectrum slicing, adaptive security mechanisms. Azimi *et al.* [5] analysed the management of RAN resources for 5G network slicing by evaluating existing models and presenting solutions to address future challenges. These challenges include slice isolation, dynamic power allocation, mobility management, dynamic slice management and creation, adaptation to new architectures etc. Su *et al.* examined resource allocation schemes for 5G network slicing in [18], categorising mathematical models based on their characteristics. Campolo *et al.* [19] delved into the role of NS for V2X services and suggested dedicated slices designs. In [20], Habibi *et al.* addressed the architecture of the 5G network slicing system, focusing on business aspects and modelling benefits. Trivisonno *et al.* [21] analysed technical solutions endorsed by 3GPP standardisation working groups for NS, explaining architectural implications and pending issues. However, despite a large amount of studies on network slicing, it should be noted that there is a lack of experimental validation or simulation results in the literature. There is also a lack of studies in the literature on energy consumption analysis, despite it is a crucial indicator in the implementation of networks. In this work, we address the implementation, performance evaluation, and energy consumption analysis of a 5G SA network slicing testbed built using OpenAirinterface and USRP B210. The testbed is implemented so that IoT related applications requiring eMBB, mMTC and URLLC capabilities are handled all together with specific slices. Consequently, our main contributions are as follows: 1) to study, design, and implement an experimental testbed for end-to-end network slicing in a 5G network, applicable to a drone network for an IoT application; 2) to analyze the system performance in terms of throughput, jitter, and latency; 3) to evaluate the system's energy consumption before and after network slicing; 4) to validate the entire system by meeting common user requirements such as web browsing, video streaming, and VoNR.

3. Overview of 5G NS

3.1. Key Concepts and Principles

5G NS is an innovative concept introduced by the Next Generation Mobile Network (NGMN) in [22]. It provides the opportunity to create multiple autonomous logical networks within a shared physical infrastructure. Three distinct layers appear in the NS process [23], as illustrated in **Figure 1**. The upper, middle and lower layer group the provided services to end users, the collection of resources to offer

different services and the entire collection of resources to be shared, respectively.

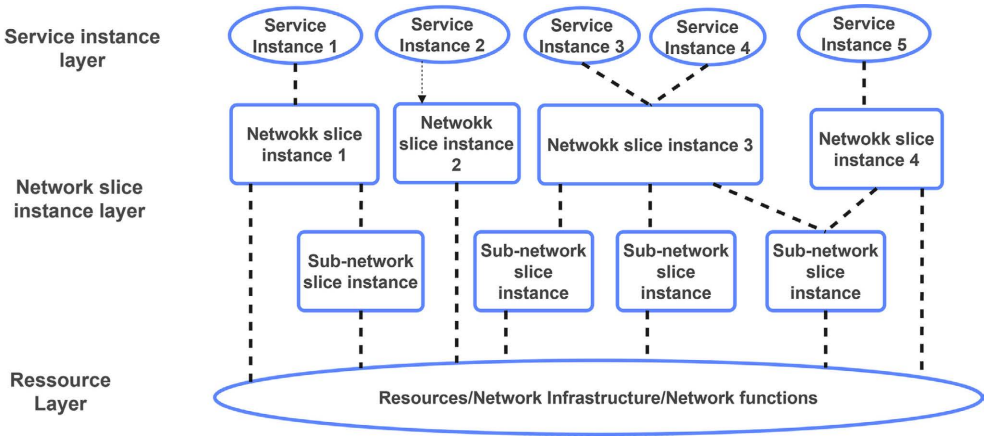


Figure 1. Conceptual NGMN diagram of 5G NS.

3.2. 5G Slicing

3.2.1. Architecture

The need to make the mobile CN more flexible, elastic, and scalable has led 3GPP to introduce in 3GPP TS 23.501 release 15 a new CN architecture, also known as the Next Generation (NG) core network or the 5G network architecture. This new architecture aims to decompose the functions of the current EPC into more specific and modular functions. The most important network functions, as defined in 3GPP TS 23.501 release 15 and illustrated in Figure 2, are as follows:

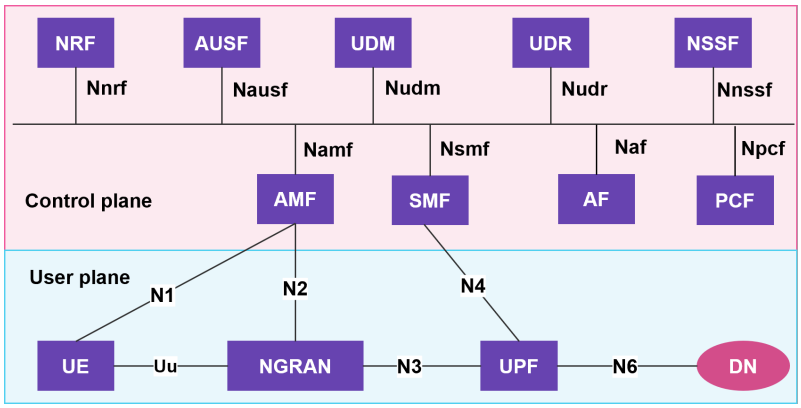


Figure 2. 5G architecture: service-oriented model according to 3GPP TS 23.501.

- The registration, authentication, and mobility management function (AMF): The AMF receives session and connection data from the UE via the N1 and N2 interfaces and handles connection management and mobility tasks. The AMF is described in 3GPP TS 29.518.
- The packet forwarding, traffic routing, and Quality of Service (QoS) enforcement function (UPF): The UPF enables low-latency edge computing. and the PDU session within the UPF ensures UE-DN connectivity with different QoS, depending on the service [24].

- The session management and control of user data function (SMF): The SMF mainly sets up, modifies and releases sessions. It performs the DHCP protocol and ARP proxy for PDUs and many other functionalities. It is connected to the UPF via the N4 interface. The SMF is described in 3GPP TS 29.502.
- The authentication and security functions (AUSF): The AUSF handles the security procedure for UE authentication using the 5G-AKA authentication method, which includes mutual authentication between the device and the network. The AUSF is described in 3GPP TS 29.509.
- The unified data management (UDM): The UDM handles user subscription and identification services. The UDM is described in 3GPP TS 29.503.
- The network repository function (NRF): The NFR supports the discovery of network functions and their supported services. The NRF is described in 3GPP TS 29.510.
- The policy control function: The PCF unifies the network policies and provides policy rules to control plane functions. The PCF is described in 3GPP TS 29 514.

Transmitting and receiving data is dedicated to the NG-RAN. The NG-RAN is described in 3GPP TS 38.300.

The architecture of the NG core evolved from a model-based representation with point-to-point links to a service-oriented approach with different control planes, mutual discovery and interaction of NFs through the NRF. The architecture of a service-oriented NG core can be deployed in a software or virtualised environment such as virtual machines. It also enables network slicing through a new function known as the Network Slice Selection Function (NSSF). The NSSF is responsible for redirecting traffic to a specific network slice based on the data characteristics. The NSSF is described in 3GPP TS 29 531.

Beside customising the composition of NFs, a major advantage of fine granularity for network slicing lies in the ability to share NFs among slices to simplify their management and reduce signalling. It is therefore essential to identify NFs which can be shared to leverage radio access and core network usage. Three groups, A, B and C are identified in 3GPP TR 23.799 release 14.

These 3 groups give a set of shared NFs among several slices. The group (A) is characterised by common RAN network functions and fully dedicated CN slices. A UE can access services from various slices and different instances of the CN. In this configuration, dedicated core NFs may encompass functions such as subscription, mobility, or session management, namely AUSF, AMF, SMF, UDM and UDR. This approach offers advantages in terms of isolation between core NFs, however signalling load may increase. The group (B) is also characterised by a common RAN. AUSF, UDM, UDR, and AMF are shared between all slices and other functions such as session management and user plane function (SMF and UPF) reside in individual slices. This helps reduce signalling load while ensuring high isolation. Finally, the group (C) envisions a fully shared RAN and common control plane management among network slices, while user planes (UPF) are managed as separate network slices. In this case, there is not very high isolation

between slices, but signalling load is significantly reduced.

3.2.2. Discovering and Selecting Network Slice

The NSSAI (network slice selection assistance information) helps the network to select the appropriate slice. Users transmit NSSAIs during registration procedures while RRC and NAS messages are exchanged. NSSAIs consist of a vector containing up to eight unique S-NSSAI (Single NSSAI) values identifying network slices that applies across the 5G Core (5GC), the 5G-RAN, and the UE. Each S-NSSAI comprises two main elements: the Slice/Service Type (SST) ID and an optional information called the Slice Differentiator (SD). The SST ID (8 bits length) describes the characteristics of a slice and the SD (ID (24 bits length) is used to distinguish slices sharing the same SST. S-NSSAIs can be standardised or tied to the PLMN. Standardised S-NSSAIs only include the SST ID, without SD, while non-standardised S-NSSAIs can be defined with SST ID alone or SST ID accompanied by SD. For global interoperability in network slicing, 3GPP has reserved standardised SST values from 0 to 127. Non-standardised values, ranging from 128 to 255, allow operators to introduce network slices tailored to their specific service needs. For instance, a SST of 1 is for eMBB type slice, a SST of 2 is for the uRLLC type slice and SST 3 corresponds to an mIoT type slice. There are different categories of NSSAI.

- Subscribed S-NSSAI: These values are stored in the subscriber registry (UDM). The 5GC uses them by default in case no NSSAI is received from the UE.
- Configured NSSAI: The PLMN can configure a UE with one or more configured NSSAIs. The UE uses them as the default NSSAI. These NSSAIs may be identical to the subscribed S-NSSAIs. Up to 16 S-NSSAIs can be included in a configured NSSAI. The AMF uses NAS Registration Accept messages or NAS Configuration Update Command to inform the UE of the configured NSSAI.
- Allowed NSSAI: These values are assigned by the 5GC and are valid in a registration area or for the PLMN on a given access type. Each allowed NSSAI can include up to 8 network slices, meaning a UE can be served by a maximum of 8 network slices. The use of these 8 network slices requires the UE to establish 8 PDU sessions. The AMF also signals the allowed NSSAI to the gNB through the NGAP Initial Context Setup Request message, which the base station uses for UE resource allocation.
- Requested NSSAI: This category may encompass a configured and allowed NSSAI. The UE transmits the requested NSSAI during completed RRC configuration and NAS registration request. The gNB uses this information for AMF selection and for preliminary processing before obtaining NSSAI authorisation. The 5GC uses the requested NSSAI for slice selection, validation, etc., and then returns the allowed NSSAI.
- Rejected NSSAI: A NSSAI can be rejected in a specific registration area or throughout the PLMN. In such a case, the UE must not attempt to use this NSSAI during this registration procedure or within the PLMN.

4. Our System

The objective of this work is to improve our system introduced in [25] by building a testbed aimed at optimizing resource management and enhancing the energy efficiency of drones within a deployed drone network, for example, for monitoring a dense forest following a wildfire. To avoid overload and bottlenecks, and to ensure quality and long lifespan of the system, we implemented three network slices on the same physical infrastructure. One slice has access to a Multi-Access Edge Computing (MEC) system while the other two are dedicated to the data network. The first slice is designed to handle high throughput (eMBB) to transmit data collected by onboard cameras, the second slice enables connection and data exchange between onboard sensors for IoT service, and the third slice is used for real-time communication (URLLC) with base stations. The schematic diagram of the considered scenario is depicted in **Figure 3**.

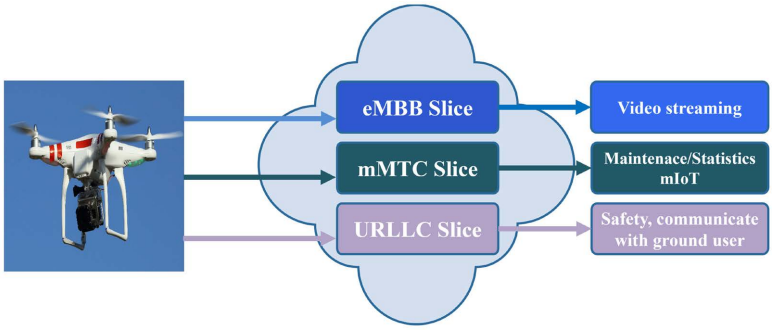


Figure 3. Schematic diagram of the scenario studied.

4.1. Architecture

To implement our testbed, we utilised OpenAirInterface, an open-source platform freely available. The source code and documentation are accessible on their platform [26]. However, it is worth mentioning that network slicing, in particular, is still under development in OpenAirInterface (OAI).

Our system is shown in **Figure 4**, and its total cost is less than $5000. It is deployed on computers with the following specifications: Operating System: Linux Lowlatency kernel Ubuntu 20.04 LTS; Processors: Intel(R) Core (TM) i5-10210U CPU @ 1.60 GHz 2.11 GHz; RAM: 8 GB; UHD Driver. Two USRP B210 SDRs are used for this setup, one acting as a radio access point at the gNB level, and the other emulating a UE. Each USRP is linked to a machine via Universal Serial Bus (USB) 3.0 interface. We used the n78 band, a 3.6 GHz frequency and a 40 MHz bandwidth. The antenna gain is 18 dB, and the measurement distance ranges up to 10 meters depending on the UE.

OAI offers two solutions for UPF implementation: oai-spgwu-UPF and VPP-UPF. The slicing in OAI is done using the latter. However, we chose the oai-spgwu-UPF solution and successfully implemented it. Indeed, VPP-UPF is not suited for environments with limited computing resources as revealed by our experiments in [27] which gave almost zero throughput with VPP-UPF while Spettel

et al. [28], whom work on high performance EURECOM servers obtained throughput not exceeding 1.5 Mbit/s. The oai-spgwu-UPF option proved to be more efficient in our specific context.

Figure 4. The testbed.

To design the network slicing architecture, we relied on 3GPP standards. As stated in Section 0.0.1, group (A) suffers from network overload while group (C) has a slice isolation issue, we chose group B for our system implementation, as it offers a good compromise between isolation and reducing signalling overhead. In a first step, we deployed two distinct slices, each having its own set of components, including UPF, SMF, while sharing the same AMF, NSSF, UDR, UDM, and AUSF. We then deployed a third slice with specific UPF, SMF, and NRF, while continuing to share the same AMF, NSSF, UDR, UDM, and AUSF with the two previously established slices. This architecture is depicted in **Figure 5**. The purpose of this third slice is to enable the system to provide services to real-time applications. Through this slice, we plan to offer the benefits of MEC to our system. The PDU session was established in each of slice, as illustrated in **Figure 6**, and we proceeded with performance testing to evaluate the operation of the system.

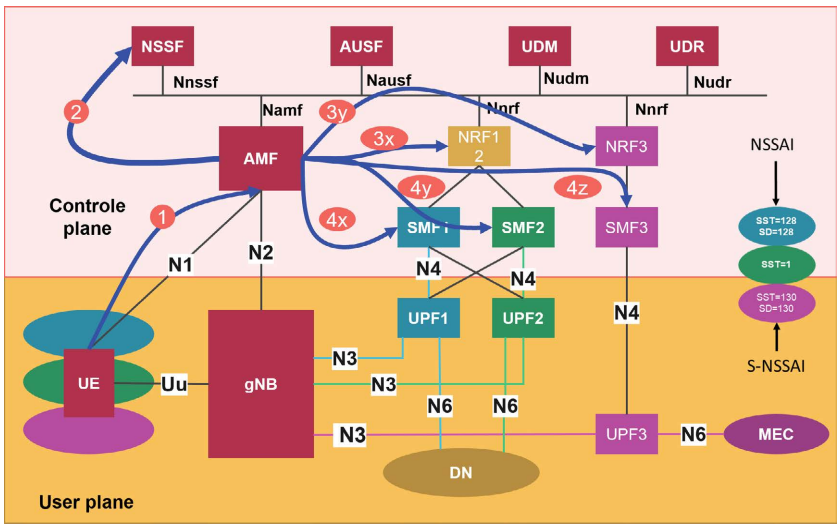


Figure 5. Architecture of 5G NS with 2 NRFs and 3 slices.

Figure 6. Image illustrating the establishment of the PDU session for each of the 3 slices.

4.2. Operating Principle

As depicted in Figure 5 and Figure 7, when a UE registers with a PLMN, it is required to inform the network by providing a requested NSSAI corresponding to the slice type the UE wishes to access, provided that the UE has a configured or allowed NSSAI. During the connection establishment steps, different UEs may indicate NSSAIs with different SST values, depending on the specific user requirements.

The RAN extract the NSSAI information when it receives the UE's RRC connection request and forwards the NAS messages to the AMF which initiates the configuration process. To do so, the AMF retrieves the user subscription information, including the subscribed S-NSSAI, through the UDM. As several slices may be approved for different users, the AMF needs to use the NSSF to obtain the allowed NSSAI. In fact, the NSSF selects an appropriate slice instance and determines the set of target AMFs to serve the UE. Since we have a single AMF instance, the NSSF thus responds to the AMF by providing the allowed NSSAI, which is then transmitted to the UE. Subsequently, the AMF interacts with the NRF to locate the UPF and assigns the most suitable SMF to establish the PDU session. The discovery and selection of the SMF are based on the information provided by the NRF or the S-NSSAI, in conjunction with the DNN. Slice selection rules associate applications with one or more slices corresponding to the S-

NSSAIs subscribed by the UE. When an application associated with a specific S-NSSAI requests traffic transmission, the UE initiates the PDU session establishment process with the control plane functions of the slice. Afterward, the chosen SMF establishes a PDU session using the S-NSSAI and the DNN. Finally, the user data traffic is handled by the dedicated user plane functions of the relevant network slice.

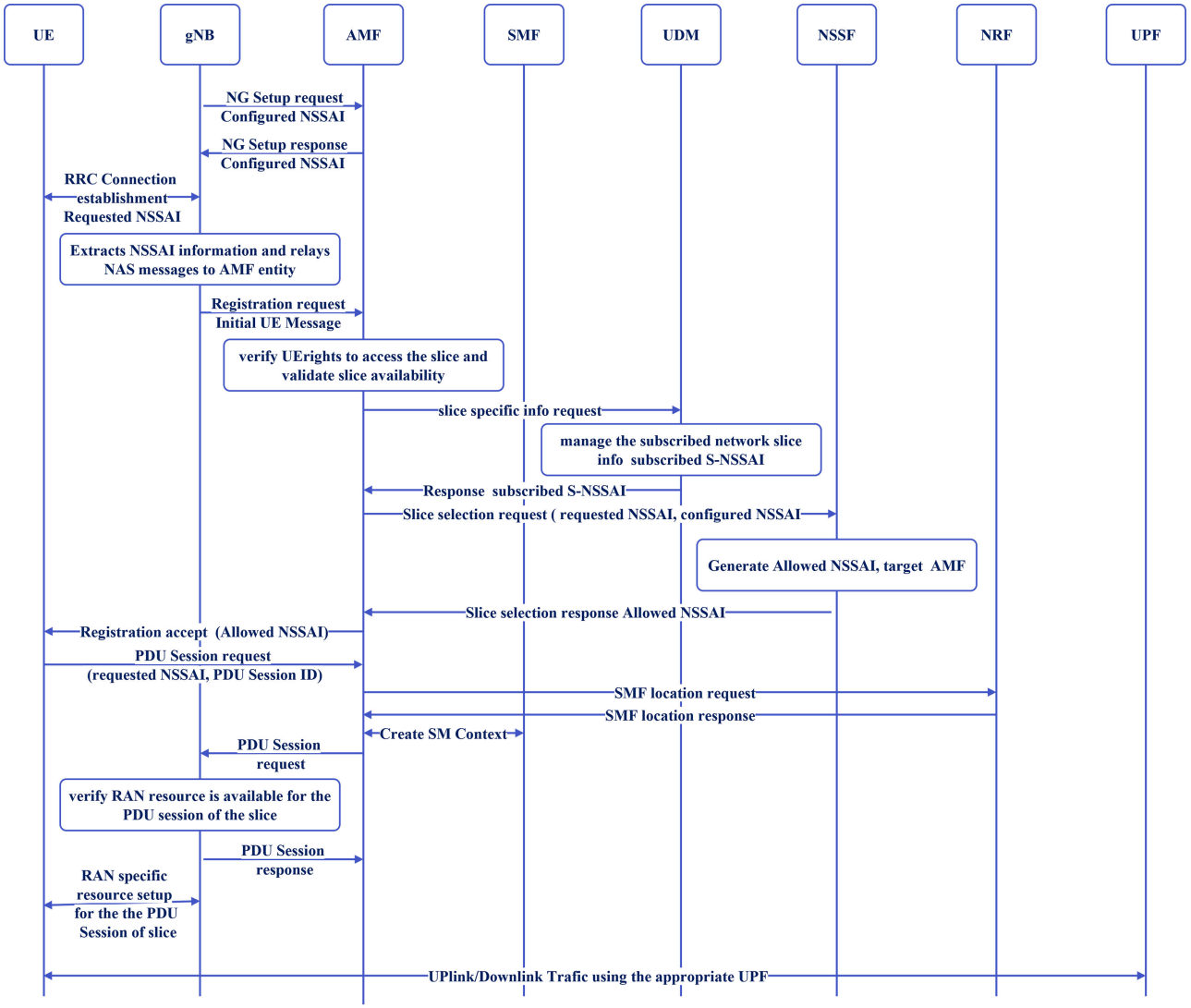


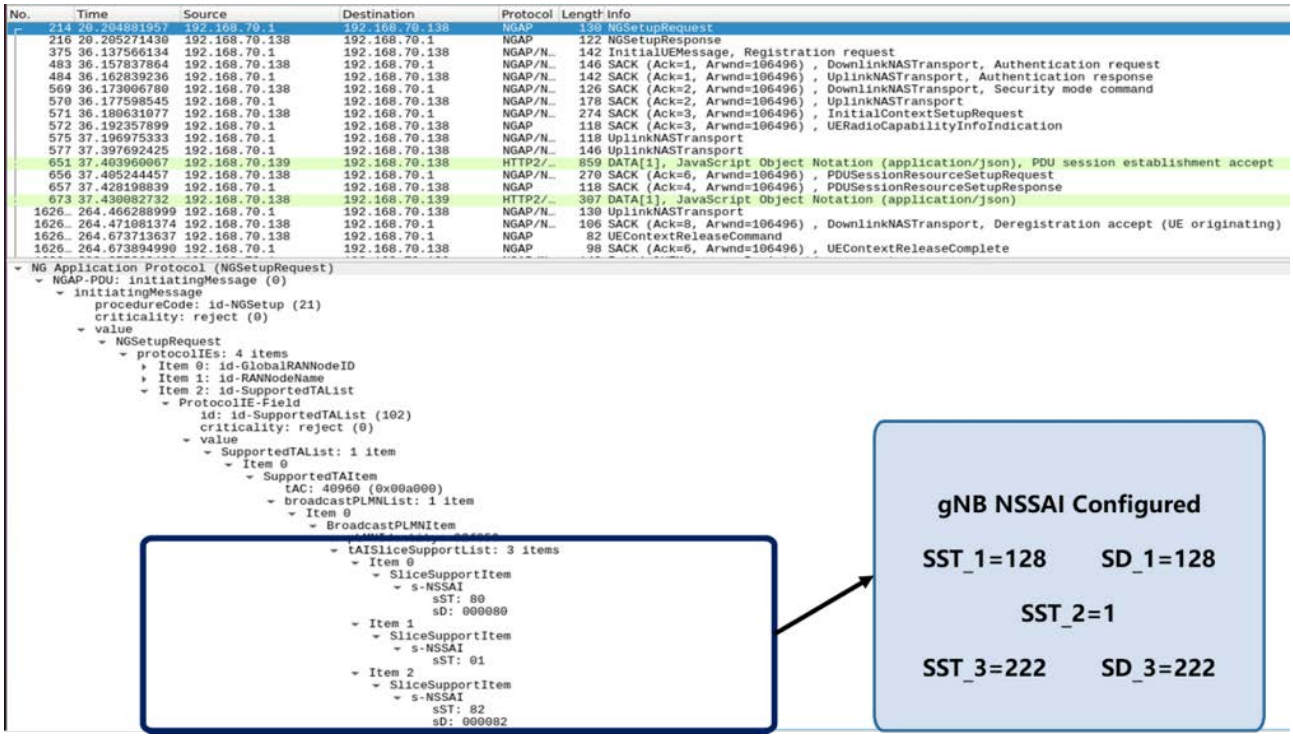
Figure 7. Principle of establishing a PDU session.

5. Performance Evaluation

We captured various NGAP signalling messages for the slices using Wireshark subsequently to the network deployment. During the gNB registration in the network, we can observe *Configured NSSAI* and Supported NSSAI using the *NGSetupRequest* and *NGSetupResponse* messages during the exchange between the AMF and the gNB as illustrated in Figure 8. Similarly, when the AMF checks the UE's access conditions, it informs the gNB of the *Allowed NSSAI* with the *Initial-*

ContextSetupRequest message, see Figure 8. During the establishment of the PDU session, *Requested NSSAI* can be observed with the *PDUSessionResource-SetupRequest*message, as illustrated in Figure 8.

Figure 6 depicts the SMF log, where it is seen that the PDU session was successfully established, and an IP address is assigned to each connected UE.



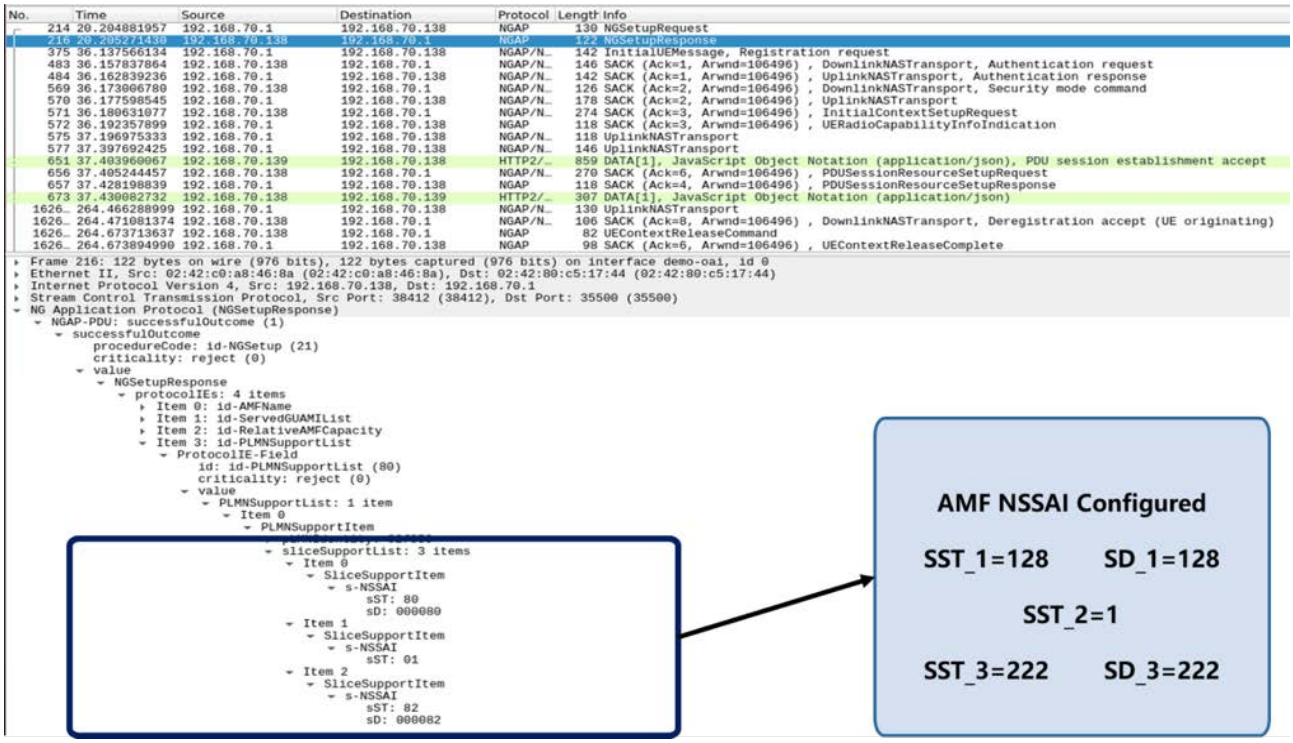


Figure 8. Wireshark analysis: gNB NSSAI configuration, AMF NSSAI configuration, gNB NSSAI allowed, UE NSSAI request.

We then evaluated the UE connectivity to the Internet via the three slices. To do so, we successfully sent pings to Google's DNS server using the IP addresses assigned to each UE, as illustrated in Figure 9. We observed that the UEs not only have fast

internet access but also exhibit similar access speeds across all slices. This demonstrates that all slices are equally handled when no service is associated. Any behaviour change in a slice could be interpreted as resulting from the associated service.

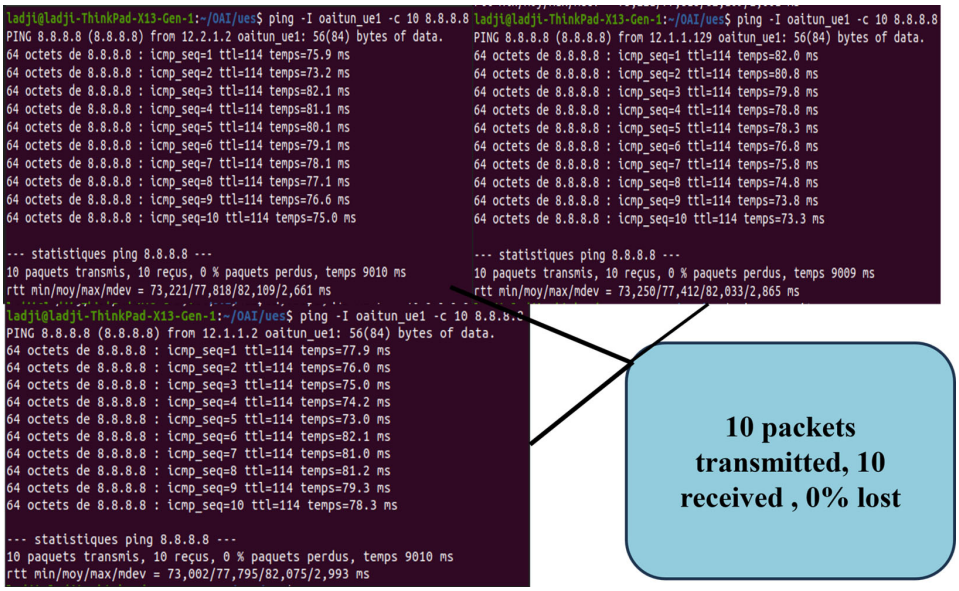


Figure 9. Connectivity test of the 3 slices.

We also measured the latency, the uplink and downlink throughputs and the jitter of the three slices using Iperf3 and Ping tools. The UDP protocol, being connectionless, transmits data without establishing a prior connection (UDP packets are sent without acknowledgment or delivery guarantee), making it ideal for applications where speed and latency outweight reliability, such as real-time multimedia streaming, online gaming, and VoIP. Regarding UDP throughput measurements, they are obtained by evaluating the bandwidth with a target throughput. By default, this target throughput is set to 1 Mbit. In our study, we measured the uplink and downlink throughputs with target throughputs of 20 Mbits and 60 Mbits, respectively. These target values were chosen to ensure consistent throughputs without being hindered by hardware limitations. In fact, this choice stems from several trials with significantly higher target throughputs but experienced network congestions because of the limited performance of our equipment.

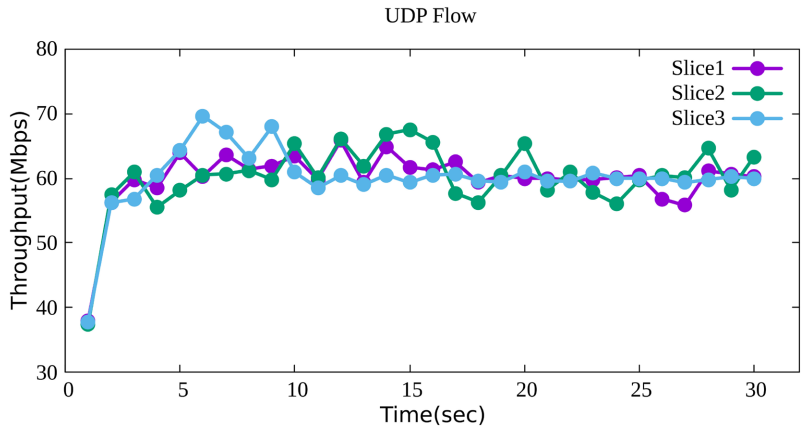
The UDP throughput and jitter results in both uplink and downlink are illustrated in Figure 10. The observed mean values are 20 Mbps and 60 Mbps for uplink and downlink throughputs, respectively. As for the jitter average values of 1 ms and 0.25 ms were observed in the uplink and downlink channels, respectively.

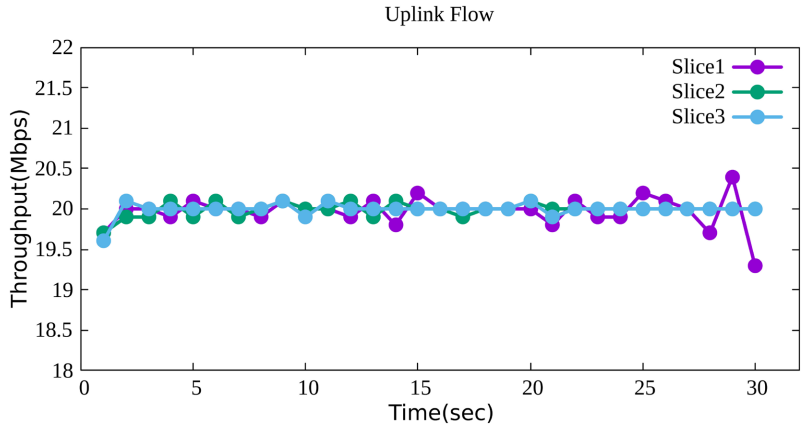
We can see in Figure 10 that the three slices exhibit similar behaviour in concordance with the concept of network slicing. As expected, the throughput and jitter in both uplink and downlink are nearly identical for the three slices. Subsequently, we evaluated the system's maximum bandwidth by measuring the TCP throughput of the three slices. The results showed a TCP throughput of approximately 60 Mbits across the three slices, as illustrated in Figure 11.

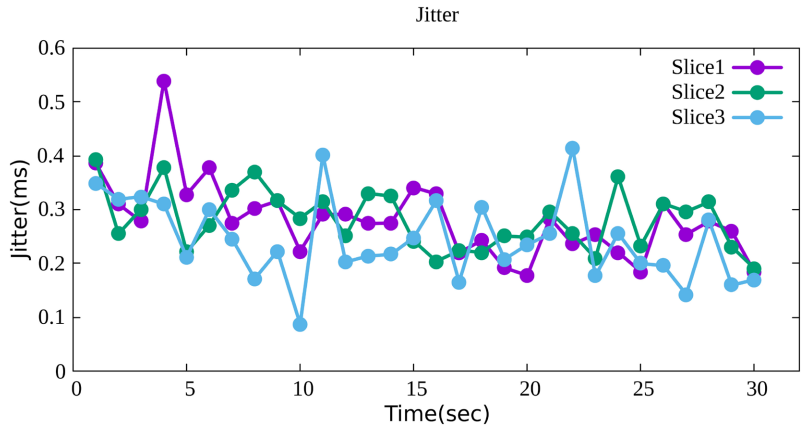
Moreover, we measured the latency of our system with the *ping* tool, successively transmitted 10 packets to the UE. The latency measurements, as given in Table 1 indicate a steady value for all slices.

Table 1. Latency measurements for all 3 slices (ms).

Slice	Minimum	Average	Maximum
#1	5.95	10.18	14.43
#2	5.5	9.65	14.02
#3	5.42	9.82	14.48







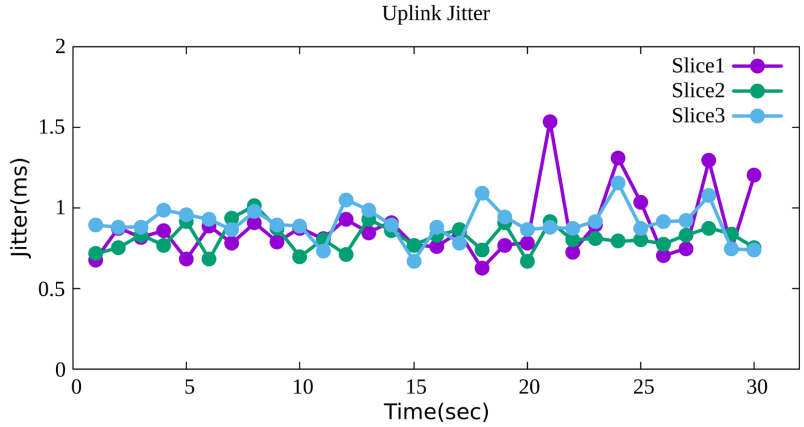


Figure 10. Throughput [Downlink , Uplink] and Jitter [Downlink, Uplink] of the 3 slices.

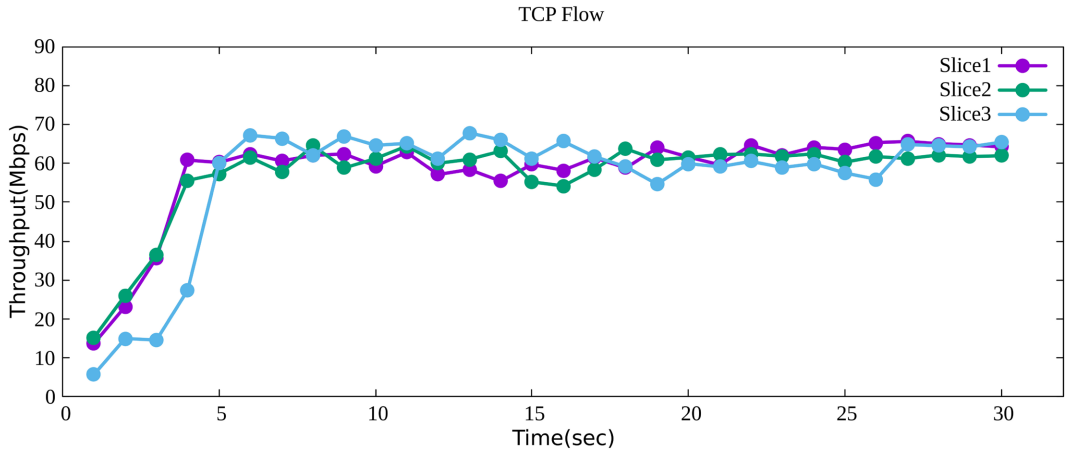


Figure 11. TCP throughput of the 3 slices.

Latency and throughput measured values remained steady when two slices were simultaneously loaded.

The measured throughputs are slightly below the 5G user-experienced data rate of 100 Mbit/s. Several factors could explain this limitation, especially both performance of the used computers and the USRP B210 devices. In order to determine the minimum and maximum theoretical throughputs that we can achieve with the frequency band, the parameters of our testbed, and the USRP B210, we used the formula given in the 3GPP TS 38.306 V15.2.0 release 15 [29]. The theoretical throughputs range between 17.40 and 72.91 Mbit/s on the downlink and between 12.41 and 52 Mbit/s on the uplink. It should be noted that our experimental results fall within these theoretical ranges. Regarding the latency, we note that the observed results show significantly lower values than the typical latency of LTE networks, which is usually around 100 ms [30]. As for the jitter, values are well below the generally accepted standards for VoIP, where a jitter lower than 40 ms is optimal and a jitter lower than 75 ms is acceptable [31].

6. Energy Consumption

We conducted a comprehensive evaluation of our system's energy consumption,

measuring it before and after implementing network slicing. Before presenting the results, it is essential to outline the methodology employed. Initially, the overall network energy efficiency was assessed without slicing, and this assessment was repeated post-slicing. By comparing these two states, we analyzed the impact of slicing on energy efficiency, particularly focusing on the gNB. The tools Scaphandre (developed by Hubblo), Prometheus, and Grafana were used to monitor energy consumption metrics.

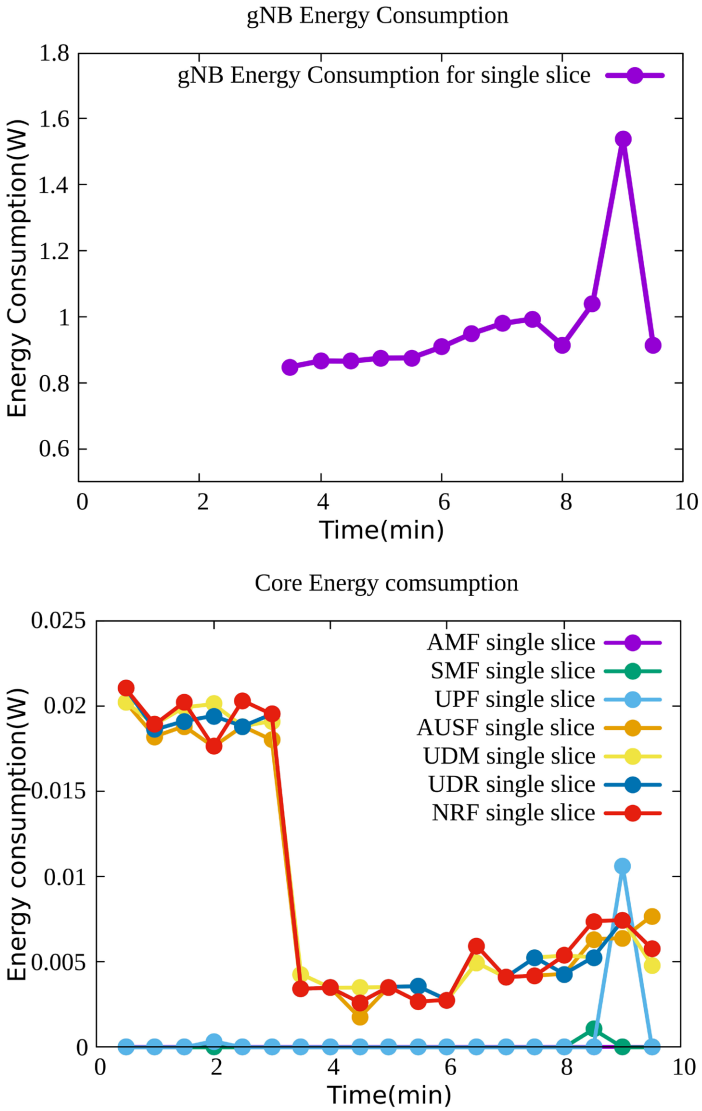


Figure 12. Energy consumption for a single slice.

During the first phase, illustrated in **Figure 12**, we observed the network's energy consumption before slicing. Measurements were recorded over a 10-minute period, during which several actions were performed, such as initializing the core network, deploying the gNB, and generating 20 Mbps traffic on both uplink and downlink channels. Initially, the core network was deployed. At the 3-minute mark, the gNB was brought online, followed by the connection of a UE at 6

minutes. At 8 minutes, 20 Mbps traffic was generated on the uplink and downlink channels. Finally, at 10 minutes, the system deployment was rolled back. We note a reduction in the core network components' energy consumption after deploying the gNB, with a slight increase in UPF energy consumption due to its role in traffic management. Overall, the core network's energy consumption remained negligible. In contrast, the gNB exhibited an average energy consumption of approximately 0.9 W in the absence of traffic, gradually increasing to nearly 1.8 W under 20 Mbps traffic on both channels.

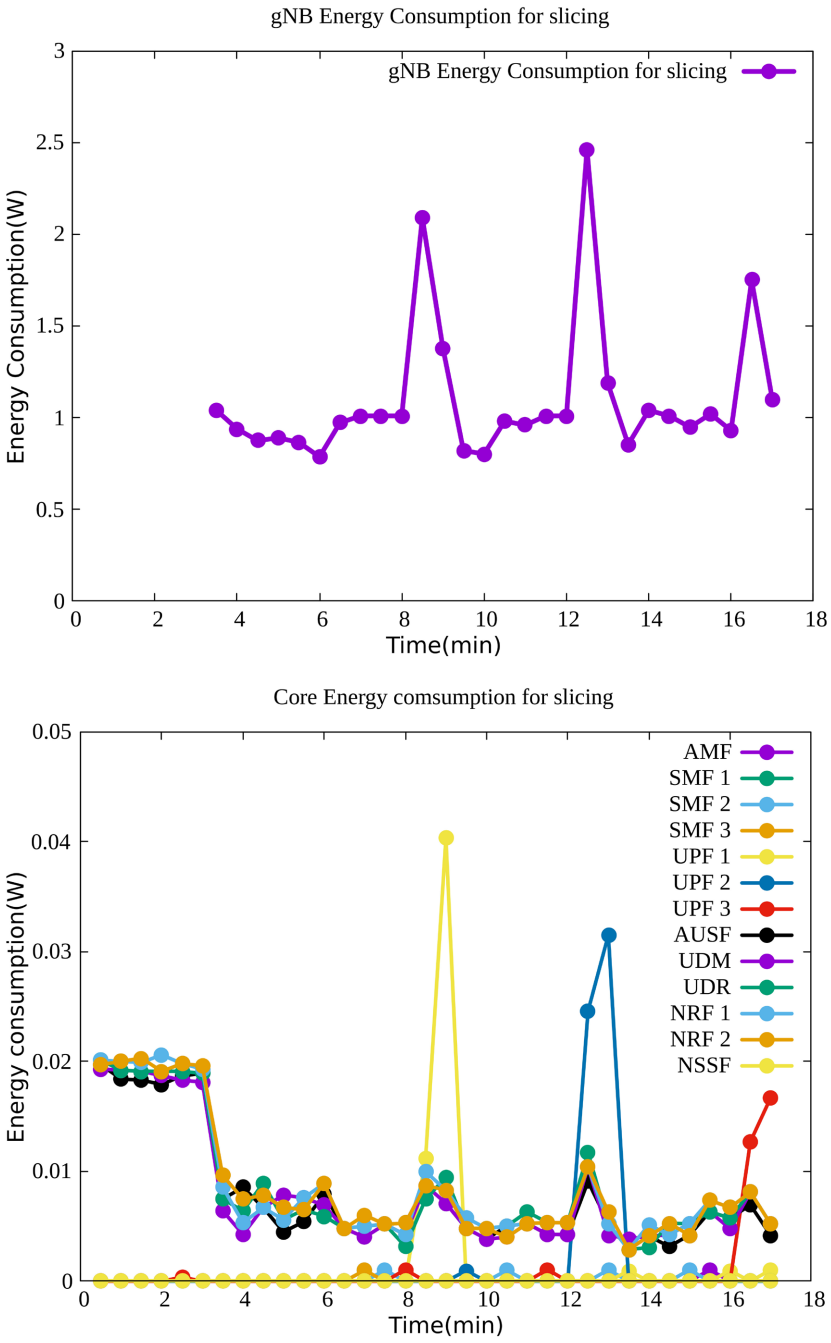


Figure 13. Energy consumption for network slicing.

During the second phase, depicted in Figure 13, we evaluated energy consumption after slicing the network into three distinct slices. Measurements spanned an 18-minute period, involving similar actions to those in the first phase, with the added utilization of slices for UE connectivity. Initially, the core network was deployed. At the 3-minute mark, the gNB was deployed. At 6 minutes, a UE was connected to slice 1. At 8 minutes, 20 Mbps traffic was initiated on the uplink and downlink channels. At 10 minutes, a second UE connected to slice 2, followed at 12 minutes by 20 Mbps traffic on both channels. At 14 minutes, a third UE connected to slice 3, with traffic on both channels beginning at 16 minutes. Finally, at 18 minutes, the system deployment was rolled back. As in the first phase, we observed a reduction in the core network components' energy consumption after the gNB deployment, alongside a slight increase in UPF energy consumption in the presence of traffic. The overall core network energy consumption remained negleagable. For the gNB, average energy consumption was approximately 1W in the absence of traffic, increasing to nearly 2.5 W under 20 Mbps traffic, consistent with observations from the first phase.

By comparing the network's pre- and post-slicing states, we can conclude that network slicing does not induce additional energy consumption in the core network components. Regarding the gNB, Figure 14 compares energy consumption before and after network slicing, revealing nearly identical energy consumption levels. This highlights the energy efficiency advantages of network slicing. Adopting a three-slice network reduces energy consumption by up to three times compared to deploying three separate networks. Additionally, the energy consumption difference between a network where multiple UEs use a single slice and one where UEs access different slices for distinct services underscores the benefits of effective segmentation in optimizing energy usage. Given that the core network consumes more energy under traffic load, it can be inferred that networks where multiple UEs share a single slice exhibit significantly higher energy consumption than networks where UEs access separate slices tailored to specific services.

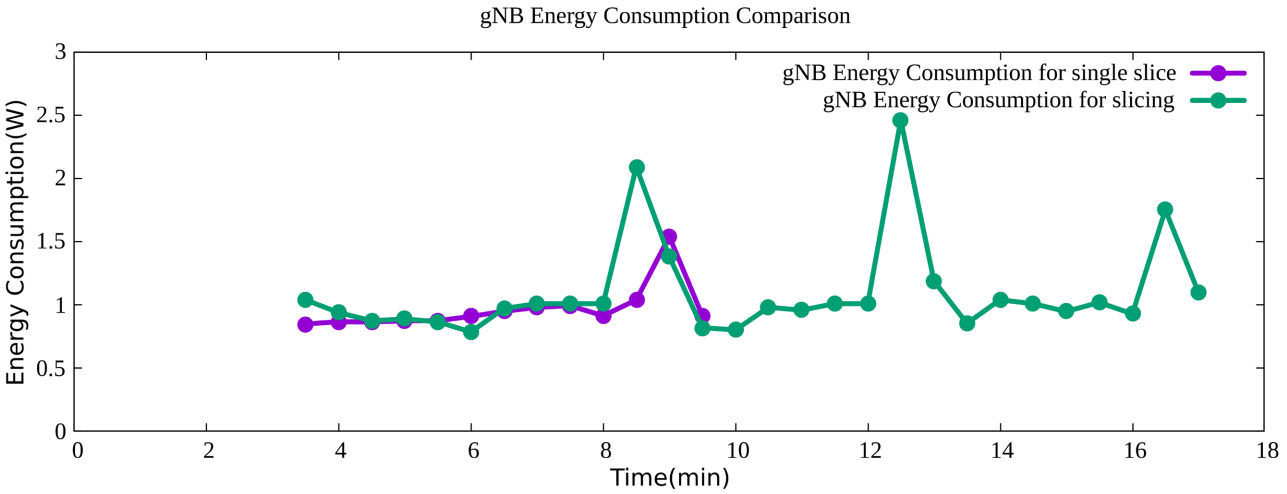
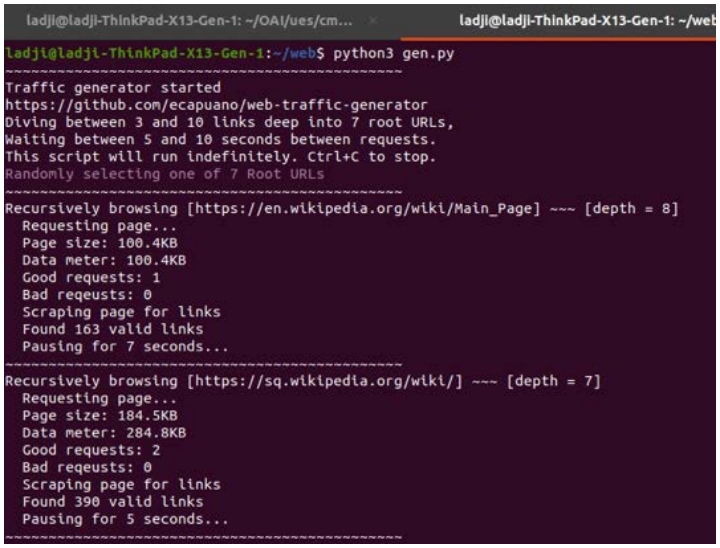


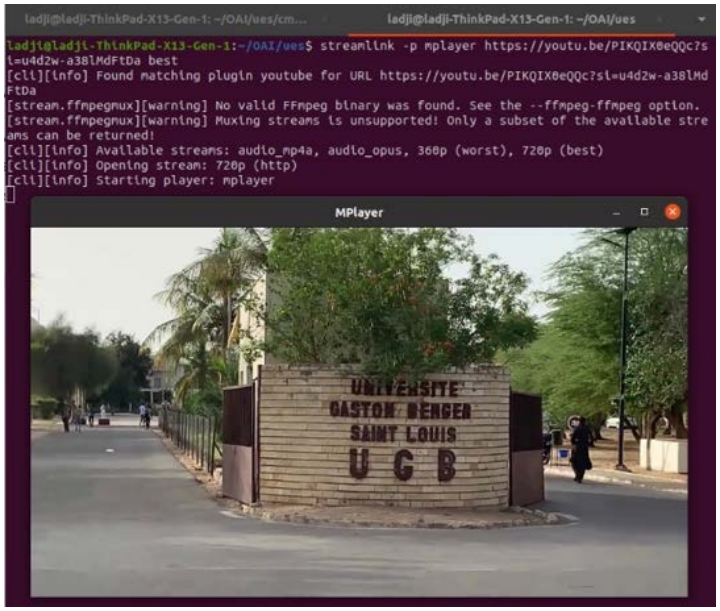
Figure 14. gNB energy consumption comparison.

7. System Validation

We conducted tests on several applications, including web browsing, video streaming, and VoNR, to validate it. To assess the web application, we set up a web traffic generator module [32] which simulates real user navigation on the internet, as illustrated in **Figure 15(a)**. Similarly, we successfully implemented the video streaming application using the StreamLink Python library [33], as illustrated in **Figure 15(b)**. StreamLink is a command-line utility that enables users to access and stream video feeds from various streaming platforms through their preferred media player, rather than via a web browser. It supports various streaming services, including livestreams, Twitch, YouTube, and many others.



(a)



(b)

Figure 15. (a) Web traffic application; (b) Video streaming application.

As for VoNR, we deployed Asterisk V18 [34] as the IMS server to communicate with our 5G system. Asterisk is an open-source software for developing communication applications supporting text, voice, and video. We adopted the gsm and G.711 a-law codecs as alternative audio coding methods. Figure 16 illustrates the flow exchanges between UEs, IMS, and UPF, while Figure 17 presents the audio samples obtained with all slices.

Figure 16. Flow exchange graph between UEs, IMS, and UPF.

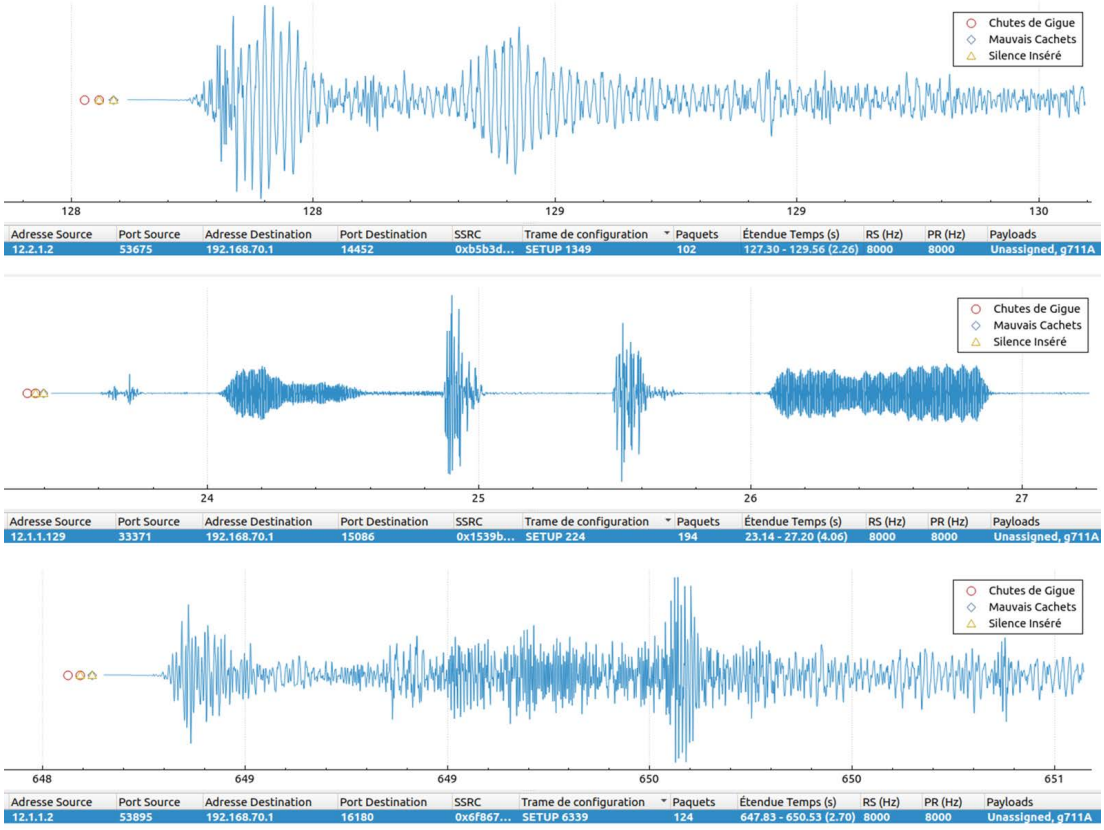


Figure 17. Images of recorded audio samples.

8. Conclusion

Network slicing emerged as an essential solution to address the complex challenges associated with 5G, including efficient resource management, flexibility, and scalability. We implemented a 5G end-to-end network slicing testbed with three slices dedicated to eMBB, URLLC and mMTC use cases type applications. This design is particularly suitable for drone network in IoT applications needing VoNR and video streaming capabilities. In such context, energy consumption is of great importance to preserve system's life time. Accordingly, this study illustrates that network slicing does not increase the overall energy consumption of the 5G system. First, a literature overview on 5G NS was presented, followed by a detailed explanation of the key principles and concepts of this technology. Then, we outlined our system along with the scenario on which we worked to set up the testbed. Once the testbed was set up, we evaluated the performance and energy consumption of our system, validating it with applications such as web browsing, video streaming, and VoNR. The results obtained with regard to latency and jitter are very satisfactory, and they meet the generally accepted standards for most IoT applications. The observed latency (around 6 ms) exhibited significantly better performance than the typical latency of LTE networks. As for jitter, it was below the generally accepted standards for VoIP. However, the obtained throughput is low, mainly due to the performance of the equipment and hardware used for this testbed, thus limiting its use to cases where throughput is not crucial. These preliminary results show a very promising path in addressing resource management, connectivity, and energy usage issues for drone network and IoT applications.

Acknowledgements

This work has been supported by WAEMU through *Programme de Bourses d'Excellence.*

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Talwar, S., Choudhury, D., Dimou, K., Aryafar, E., Bangerter, B. and Stewart, K. (2014) Enabling Technologies and Architectures for 5G Wireless. 2014 *IEEE MTT-S International Microwave Symposium* (*IMS*2014), Tampa, 1-6 June 2014, 1-4. https://doi.org/10.1109/mwsym.2014.6848639

[2] Fuentes, M., Carcel, J.L., Dietrich, C., Yu, L., Garro, E., Pauli, V., *et al.* (2020) 5G New Radio Evaluation against IMT-2020 Key Performance Indicators. *IEEE Access*, **8**, 110880-110896. https://doi.org/10.1109/access.2020.3001641

[3] Cisco (2019) Cisco Visual Networking Index: Forecast and Trends, 2017-2022. https://www.futuretimeline.net/data-trends/pdfs/cisco-2017-2022.pdf

[4] Cisco Annual Internet Report (2018-2023). https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[5] Azimi, Y., Yousefi, S., Kalbkhani, H. and Kunz, T. (2022) Applications of Machine Learning in Resource Management for Ran-Slicing in 5G and Beyond Networks: A Survey. *IEEE Access*, **10**, 106581-106612. https://doi.org/10.1109/access.2022.3210254

[6] Wu, Y., Dai, H., Wang, H., Xiong, Z. and Guo, S. (2022) A Survey of Intelligent Network Slicing Management for Industrial IoT: Integrated Approaches for Smart Transportation, Smart Energy, and Smart Factory. *IEEE Communications Surveys & Tutorials*, **24**, 1175-1211. https://doi.org/10.1109/comst.2022.3158270

[7] Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A. and Flinck, H. (2018) Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions. *IEEE Communications Surveys & Tutorials*, **20**, 2429-2453. https://doi.org/10.1109/comst.2018.2815638

[8] GSMA (2018) Network Slicing Use Case Requirements-XED. https://www.gsma.com/futurenetworks/wp-content/uploads/2018/07/Net-work-Slicing-Use-Case-Requirements-_xed.pdf

[9] Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J.J., Lorca, J. and Folgueira, J. (2017) Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. *IEEE Communications Magazine*, **55**, 80-87. https://doi.org/10.1109/mcom.2017.1600935

[10] Debbabi, F., Jmal, R. and Chaari Fourati, L. (2021) 5G Network Slicing: Fundamental Concepts, Architectures, Algorithmics, Projects Practices, and Open Issues. *Concurrency and Computation*: *Practice and Experience*, **33**, e6352. https://doi.org/10.1002/cpe.6352

[11] Barakabitze, A.A., Ahmad, A., Mijumbi, R. and Hines, A. (2020) 5G Network Slicing Using SDN and NFV: A Survey of Taxonomy, Architectures and Future Challenges. *Computer Networks*, **167**, Article ID: 106984. https://doi.org/10.1016/j.comnet.2019.106984

[12] Kaloxylos, A. (2018) A Survey and an Analysis of Network Slicing in 5G Networks. *IEEE Communications Standards Magazine*, **2**, 60-65. https://doi.org/10.1109/mcomstd.2018.1700072

[13] Zhang, S. (2019) An Overview of Network Slicing for 5G. *IEEE Wireless Communications*, **26**, 111-117. https://doi.org/10.1109/mwc.2019.1800234

[14] Sajjad, M.M., Bernardos, C.J., Jayalath, D. and Tian, Y. (2022) Inter-slice Mobility Management in 5G: Motivations, Standard Principles, Challenges, and Research Directions. *IEEE Communications Standards Magazine*, **6**, 93-100. https://doi.org/10.1109/mcomstd.0001.2000025

[15] Chahbar, M., Diaz, G., Dandoush, A., Cerin, C. and Ghoumid, K. (2021) A Comprehensive Survey on the E2E 5G Network Slicing Model. *IEEE Transactions on Network and Service Management*, **18**, 49-62. https://doi.org/10.1109/tnsm.2020.3044626

[16] Salahdine, F., Liu, Q. and Han, T. (2022) Towards Secure and Intelligent Network Slicing for 5G Networks. *IEEE Open Journal of the Computer Society*, **3**, 23-38. https://doi.org/10.1109/ojcs.2022.3161933

[17] Khan, L.U., Yaqoob, I., Tran, N.H., Han, Z. and Hong, C.S. (2020) Network Slicing: Recent Advances, Taxonomy, Requirements, and Open Research Challenges. *IEEE Access*, **8**, 36009-36028. https://doi.org/10.1109/access.2020.2975072

[18] Su, R., Zhang, D., Venkatesan, R., Gong, Z., Li, C., Ding, F., *et al.* (2019) Resource Allocation for Network Slicing in 5G Telecommunication Networks: A Survey of Principles and Models. *IEEE Network*, **33**, 172-179.

https://doi.org/10.1109/mnet.2019.1900024

[19] Campolo, C., Molinaro, A., Iera, A. and Menichella, F. (2017) 5G Network Slicing for Vehicle-To-Everything Services. *IEEE Wireless Communications*, **24**, 38-45. https://doi.org/10.1109/mwc.2017.1600408

[20] Habibi, M.A., Han, B. and Schotten, H.D. (2017) Network Slicing in 5G Mobile Communication Architecture, Profit Modeling, and Challenges. arXiv: 1707.00852.

[21] Trivisonno, R., An, X. and Wei, Q. (2017) Network Slicing for 5G Systems: A Review from an Architecture and Standardization Perspective. 2017 *IEEE Conference on Standards for Communications and Networking* (*CSCN*), Helsinki, 18-20 September 2017, 36-41. https://doi.org/10.1109/cscn.2017.8088595

[22] Deibert, P. (2015) NGMN 5G White Paper. NGMN Alliance.

[23] Alliance, N. (2016) Description of Network Slicing Concept. https://ngmn.org/wp-content/uploads/160113_NGMN_Network_Slicing_v1_0.pdf

[24] Rommer, S., Hedman, P., Olsson, M., Frid, L., Sultana, S. and Mulligan, C. (2019) 5G Core Networks: Powering Digitalization. Academic Press.

[25] Diawara, M. and Faye, A. (2025) An Experimental 5G Standalone Testbed for Rural Connectivity. In: Cheikh, M.F., Kebe, K., Gueye, A., Ndiaye, A., Sene, N.A. and Maiga, A.S., Eds., *Innovations and Interdisciplinary Solutions for Underserved Areas*, Springer, 100-114. https://doi.org/10.1007/978-3-031-86493-3_9

[26] OpenAirInterface 5G Software Alliance for Democratising Wireless Innovation. https://openairinterface.org/

[27] Mahamadou, D. and Andre, F. (2024) 5G Standalone Network Slicing Enabled on Top of a Multi-Access Edge Computing Environment. 2024 *IEEE Multi-Conference on Natural and Engineering Sciences for Sahel's Sustainable Development* (*MNE3SD*), Ouagadougou, 28-30 November 2024, 1-7. https://doi.org/10.1109/mne3sd63831.2024.10812156

[28] Spettel, S. (2023) Robotics and Edge Computing in 5G: A Prototype for the OpenAirInterface 5G System. https://repositum.tuwien.at/bitstream/20.500.12708/158270/1/Spettel%20Stefan%20-%202023%20-%20Robotics%20and%20Edge%20Computing%20in%205G%20A%20Prototype%20for%20the...pdf

[29] 3GPP (2018) NR; User Equipment (UE) Radio Access Capabilities. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3193

[30] Zhu, Y., Halpern, M. and Reddi, V.J. (2015) The Role of the CPU in Energy-Efficient Mobile Web Browsing. *IEEE Micro*, **35**, 26-33. https://doi.org/10.1109/mm.2015.8

[31] Bassil, C. (2005) SVSP (Secure Voice over IP Simple Protocol) une solution pour la scurisation de la voix sur IP. Ph.D. Thesis, Télécom Paris Tech.

[32] GitHub—ReconInfoSec/Web-Trac-Generator: A Quick and Dirty HTTP/S "Organic" Trac Generator. https://github.com/ReconInfoSec/web-traffic-generator

[33] Streamlink 6.7.2 Documentation. https://streamlink.github.io/

[34] Asterisk. https://www.asterisk.org/