Prototyping a free5GC-based Testbed with 3GPP 5G NR and Non-3GPP WiFi Global Connectivity

Markus Haukås
Dept. of Information and Communication Technology
University of Agder (UiA), N-4898 Grimstad, Norway
markus.haukas@gmail.com

Muaz Al-Karrad
Dept. of Information and Communication Technology
University of Agder (UiA), N-4898 Grimstad, Norway
moazakrad1@gmail.com

Geir Jevne
Dept. of Information and Communication Technology
University of Agder (UiA), N-4898 Grimstad, Norway
geir.jevne@uia.no

Frank Y. Li
Dept. of Information and Communication Technology
University of Agder (UiA), N-4898 Grimstad, Norway
frank.li@uia.no

***Abstract*—Open-source 5G networks enable flexible, standards-aligned implementation and experimentation without relying on any commercial network infrastructure. In this paper, we present the design, implementation, and evaluation of a hardware-based end-to-end prototype system that simultaneously supports 3GPP 5G NR access via an SDR-enabled gNB and non-3GPP WiFi access via an SDR-based WiFi access point, both connected through a common 5G core network, *free5GC*. To support two distinct radio access technologies, the core network is integrated with OAI RAN for 5G NR access through a USRP B210 running as the radio unit and with OpenWiFi using an AntSDR E200 as the WiFi access point, with its connection to free5GC enabled by the N3IWF interface. For end users, COTS Android smartphones serve as 5G UEs and N3IWUE-enabled devices serve as WiFi stations. Experimental results demonstrate the operability of our prototype system with end-to-end global Internet connectivity on both access paths, a peak UE downlink throughput of approximately 127 Mbps with 0% packet loss, and successful 4K (2160p) video streaming. With concurrent 5G NR and WiFi traffic, the achieved non-3GPP path throughput reaches 13.8–18.5 Mbps, versus 14.4–24.5 Mbps when running only WiFi traffic.**

***Index Terms*—5G, free5GC, N3IWF, 3GPP/non-3GPP access, open-source implementation, COTS UE, real-life experiments.**

I. Introduction

The fifth generation (5G) mobile network standard features a fundamental architectural paradigm shift, represented by network softwarization and radio access technology (RAT)-agnostic connectivity [1]. Along the path towards beyond 5G evolution, the flexibility provided by the service-based architecture (SBA) and the ability to unify diverse RATs under a single core network (CN) supporting both third generation partnership project (3GPP) and non-3GPP access are gaining popularity among vendors, network operators, service providers, and end users. This capability is especially valuable for use cases in which wireless fidelity (WiFi)-connected devices need to access a mobile CN seamlessly, without the assumption that the WiFi infrastructure is operator-controlled.

To facilitate inter-RAT connectivity, a *non-3GPP interworking function (N3IWF)* has been defined by 3GPP, serving as the interface between non-3GPP networks and the 5G SBA [1]. In accordance with the trend of network softwarization, various open-source 5G core (5GC) platforms have emerged. Among them, free5GC, Open5GS, and OpenAirInterface (OAI) CN software suites are the most eminent ones. To build a fully functional end-to-end 5G system, these CN platforms are typically integrated with open-source radio access network (RAN) implementations with a software-defined radio (SDR) serving as the radio unit (RU) of the 5G next generation NodeB (gNB) [2].

Despite vast efforts on prototyping, emulation, and real-life deployment of open-source based 5G systems [3], few published studies combine SDR-based 5G RAN with SDR-based WiFi access sharing a common open-source 5GC, particularly under the non-3GPP model that applies the N3IWF rather than the *trusted non-3GPP gateway function (TNGF)*. In [4], the control plane performance of multiple open-source 5GC solutions was evaluated from both user registration and lower-level system operation perspectives, without considering possible support of non-3GPP RATs. On the other hand, several open-source platforms that support WiFi network prototyping exist [5], [6], however, without considering joint cellular and WiFi network operation. As the most closely related prior work to our study, the system developed in [7], [8] combines free5GC with 5G new radio (NR), WiFi and visible light communication (VLC) through a common core. However, their work focuses on a modified free5GC-based multi-RAT architecture that supports multiple N3IWF instances and access traffic steering, switching, and splitting (ATSSS)-based traffic steering.

In this paper, we present a prototype operational testbed that we have developed based on the free5GC open-source CN platform [9], which supports both 3GPP 5G NR and N3IWF-based WiFi access applicable to public hotspot scenarios, providing global end-to-end connectivity to end users. More specifically, our system is composed of free5GC, OAI RAN, OpenWiFi software suites running on a single personal computer (PC), universal software radio peripheral (USRP) B210 and AntSDR E200 running as the RU and access point (AP) respectively, as well as commercial off-the-shelf (COTS) user equipment (UE). To demonstrate the operability

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

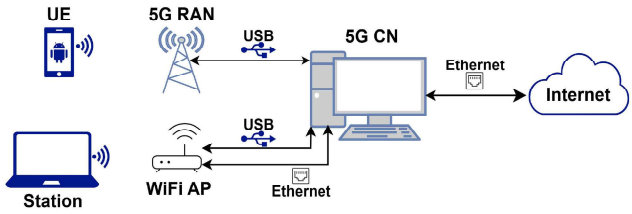


Fig. 1: A high-level overview of the system architecture.

of our testbed and assess its performance, extensive real-life experiments have been performed with respect to global connectivity, network performance, and quality of experience (QoE) for video streaming.

The rest of this paper is organized as follows. After a brief introduction of the 5GC architecture and the building blocks of the prototype system in Sec. II, we explain the design and implementation of our testbed in Sec. III. In Sec. IV, we present the performed experiments and demonstrate the performance of our system. Then, we share our experience with readers in Sec. V, before concluding the paper in Sec. VI.

II. Preliminaries and Building Blocks

In this section, we present preliminaries on the 5GC architecture and the main building blocks of our testbed.

A. 5GC Architecture and Non-3GPP Access

1) 5G CN: The 5GC is built based on the SBA where network functions expose services through service-based interfaces. The key functions in the 5GC SBA include the access and mobility management function (AMF), session management function (SMF), user plane function (UPF), unified data management (UDM), and authentication server function (AUSF). Relying on the SBA, 5G networks facilitate flexible and efficient service provisioning.

2) Enabling non-3GPP access: The N3IWF is defined in 3GPP Rel. 16 to enable non-3GPP UEs or devices to connect to the 5GC via Internet protocol security (IPsec) tunneling [1]. Based on the N3IWF, a non-3GPP device for instance a *WiFi station* (STA) can establish the required communication path towards the 5GC through which to reach the global Internet.

B. Building Blocks

The most important component in our system is an open-source CN platform that supports both 3GPP and non-3GPP RATs. As concluded in [10], among the four popular open-source CN implementation software suites, free5GC, OAI CN, Open5GS, and SD-Core, free5GC is the only one that supports non-3GPP connectivity. In addition, two software suites are selected to support 3GPP/non-3GPP radio access respectively.

1) CN–free5GC: free5GC is a *Linux Foundation* project dedicated to implementing 3GPP Rel. 17 and beyond CN [9]. It is an SBA-based 5GC implementation written primarily in Go, supporting both 5GC functionalities and non-3GPP access.

2) 3GPP RAT – OAI RAN: It offers both RAN and CN implementations, featured by its seamless integration with SDR-based platforms and its support for end-to-end 5G experimentation. However, deployment and configuration complexity are generally higher compared to other open-source solutions [11].

TABLE I: Hardware components, roles, and specifications

Component	Role	Key Specifications
Dell XPS 9320	CN + gNB host	i7-1260P, 16 GB
USRP B210	5G SDR gNB	2 × 2 MIMO, USB3
COTS UEs	5G UEs	Android 14/15
sysmoISIM-SJA5-S17	ISIM card	SUCI, OPc support
AntSDR E200	OpenWiFi AP	FPGA-based 802.11
Station laptop (N3IWUE)	WiFi station	Windows 11 + VirtualBox

‡Four COTS UE models that are successfully connected to our testbed: Samsung Galaxy S21 FE 5G; OnePlus 12; OnePlus Nord 5; Motorola Edge 30 Fusion.

3) Non-3GPP RAT – OpenWiFi: It is a Linux mac80211 implementation primarily written in C, supporting full-stack IEEE802.11n/WiFi 4 compatibility [12]. To enable WiFi connectivity, the OpenWiFi software needs to run on an SDR.

III. System Design and Implementation

In this section, we first give an overview of our system design and then present the implementation of the components.

A. System Overview

Fig. 1 presents a sketch of our system architecture including the main hardware components and the overall connections among them. In a nutshell, the system consists of one or multiple UEs connected to the 5G RAN, one or multiple stations connected to the WiFi AP, and one PC which hosts 5GC supporting both 5G and WiFi RATs. The PC is further connected to the global Internet via Ethernet offered by the university local network.

More specifically, our prototype system provides two parallel radio access paths through a shared free5GC core, i.e., 1) a 5G NR path via an OAI gNB (centralized unit (CU)–distributed unit (DU)) and a USRP B210 which functions as the RU, and (2) a non-3GPP path via OpenWiFi/AntSDR and the N3IWF for WiFi connection. Both paths are served by a single free5GC virtual machine (VM) hosted by a laptop running on a Ubuntu 24.04 long term support (LTS) operating system (OS).

B. Hardware and Software Components

1) Hardware components: Tab. I summarizes the hardware composition of our testbed from COTS UE and WiFi station to RAN and CN. In what follows, we present the three major constituent components in our testbed in more details.

- **CN and RAN:** A Dell XPS 9320 PC hosts both CN and gNB (CU–RU). While the free5GC CN runs inside a VM, the OAI RAN suite runs directly on the bare metal Ubuntu system. Additionally, a USRP B210 which functions as the RU is connected to the PC via a universal serial bus (USB) cable.
- **3GPP RAT:** The 5G NR path is provided through the OAI RAN (gNB (CU–RU)) and the USRP B210 (RU). Four COTS UE models are subscribed to the system through the Sysmocom `sysmoISIM-SJA5-S17` subscriber identity module (SIM) cards.
- **Non-3GPP RAT:** The WiFi path is provided through the OpenWiFi AP and the non-3GPP interworking UE (N3IWUE) VM. The OpenWiFi AP is connected to the

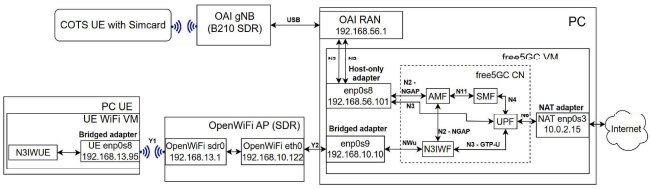


Fig. 2: Implementation details: Components and IP addresses.

same PC using Ethernet for data traffic and USB for access and control.

For providing global Internet connectivity to our testbed, the PC is connected the on-campus local computer network via Ethernet (10 Gbps) or WiFi with a university-owned gateway performing network address translation (NAT). In Fig. 2, we illustrate the main hardware components of our testbed as well as the allocated IP version 4 (IPv4) addresses.

2) Software components: The software stack adopted in our testbed contains the following major components. A complete list of these software components are given in Tab. II.

TABLE II: Software components, versions, and roles

Software	Version	Function
Oracle VirtualBox	7.0.16	Host and manage the VMs in the setup
free5GC	4.2.0	5G CN implementation
OAI RAN	2026.w11-3-gfdb6b44b05	5G RAN gNB implementation
OpenWiFi	1.5.0	WiFi AP implementation
Kuiper GNU	Linux 11.2 (bullseye)	OS for AntSDR E200
Tio	2.7	Terminal access AntSDR E200 via USB
N3IWUE	1.0.1-37-g0283256	Non-3GPP UE function
MongoDB	8.0.18	Subscriber database used by free5GC
UHD	4.8.0	Driver support for the USRP B210

- **Host system:** The host laptop runs Ubuntu 24.04 LTS as the host OS and Oracle VirtualBox for VM setup.
- **CN:** The free5GC v4.2.0 software suite runs inside a VirtualBox 7.0 VM on Ubuntu 24.04 with MongoDB 8.0.18 as the database for subscriber management. Access for both 3GPP and non-3GPP is provided through free5GC.
- **3GPP RAT:** The OAI RAN software suite version `2026.w11-3-gfdb6b44b05` is executed natively on the host PC, without relying on any VM.
- **Non-3GPP RAT:** The AntSDR runs `OpenWiFi 1.5.0` firmware, exposing an SDR-based IEEE 802.11 AP. Another laptop runs `N3IWUE v1.0.1` inside a VirtualBox VM bridged to the laptop's WiFi interface.

C. Implementation

The implementations of the CN, 3GPP RAT, and non-3GPP RAT, as well as UEs are summarized below.

1) free5GC CN: The `free5GC v4.2.0` repository provided by [9] is built inside the CN VM and the network functions are configured through its `WebConsole`. To provide both 3GPP/non-3GPP RAT access, the free5GC VM is configured with two network adapters, as explained below:

- **3GPP RAT:** The `enp0s8 (192.168.56.101)` adapter is allocated to the N2/N3 AMF interface, connecting OAI RAN via `vboxnet0 (192.168.56.1)`.
- **Non-3GPP RAT:** The `enp0s9 (192.168.10.10)` adapter is assigned for the N3IWF interface, listening on the `enp0s9` address.
- **Other interfaces:** All free5GC internal service-based interfaces adopt loopback addresses `(127.0.0.x)` registered against the network repository function (NRF) at `127.0.0.10:8000`. The public land mobile network (PLMN) is configured as mobile country codes `(MCC) 999` and mobile network code `(MNC) 70`. For the connected UEs, a pool of IPv4 addresses in the range of `10.60.0.x` is dynamically allocated through free5GC.

2) OAI RAN and USRP B210: To enable 3GPP RAT connectivity, OAI RAN is operated in the NR standalone (SA) deployment mode. The gNB (CU-DU) connects to the AMF at `192.168.56.101:38412` via the stream control transmission protocol (SCTP). After installing the required OAI dependencies, the OAI gNB is built directly on the Ubuntu host system using the `gnb.free5gc.b210.conf` configuration file. Fig. 3 illustrates the successful connection between OAI RAN gNB (CU-DU) and free5GC CN. To connect the RU with OAI RAN, a USRP hardware driver (UHD) is installed.

```
Response (RAN UE NGAP ID: 0)
2026-05-13T15:49:31.395718861+02:00 [INFO][AMF][Ngap] [AMF] SCTP Accept from: 192.168.56.1:37829
2026-05-13T15:49:31.395917221+02:00 [INFO][AMF][Ngap] Create a new NG connection for: 192.168.56.1:37829
2026-05-13T15:49:31.396022313+02:00 [INFO][AMF][Ngap][ran_addr:192.168.56.1:37829] Handle NGSetupRequest
2026-05-13T15:49:31.396073899+02:00 [INFO][AMF][Ngap][ran_addr:192.168.56.1:37829] Send NG-Setup response
```

Fig. 3: RAN connecting to the free5GC CN.

The USRP B210, which functions as the RU of the gNB, is identified by its serial number in the OAI configuration file. Following the standard OAI SA configuration, null encryption (NEA0) is adopted for data radio bearers (DRB) ciphering with advanced encryption standard (AES)-based integrity (NIA2) for integrity [13].

3) OpenWiFi: To facilitate non-3GPP RAT connectivity, the OpenWiFi [5] AntSDR AP serves the `192.168.13.0/24` subnet via the dynamic host configuration protocol (DHCP) on its SDR radio interface `(sdr0: 192.168.13.1)`. A host-side NAT rule forwards traffic from the AP to the free5GC VM's N3IWF interface `(192.168.10.10)`. The laptop which functions as a WiFi station connects to the AntSDR's service set identifier (SSID) and bridges its WiFi interface to the N3IWUE VM via a VirtualBox bridged adapter.

Moreover, the N3IWUE establishes an Internet key exchange version 2 (IKEv2) SA to the N3IWF at `192.168.10.10`, authenticates via the extensible authentication protocol–authentication and key agreement (prime) (EAP-AKA'), and creates a packet data unit (PDU) session. A generic routing encapsulation (GRE) tunnel `(gretun-id-2-1)` carries user-plane traffic inside a virtual routing and forwarding (VRF) `(vrf-pdu-1)` interface. Static routing rules inside the N3IWUE VM direct 5GC-bound traffic through the GRE tunnel.

4) COTS UEs: In general, it requires manual configuration to force COTS UEs to be operated *exclusively* in the `NR SA-only` mode. For the two **OnePlus** phones, we apply the `Set 5G Only` application (App) to force the NR-only mode and `Developer Options` to select SA+NSA priority. For

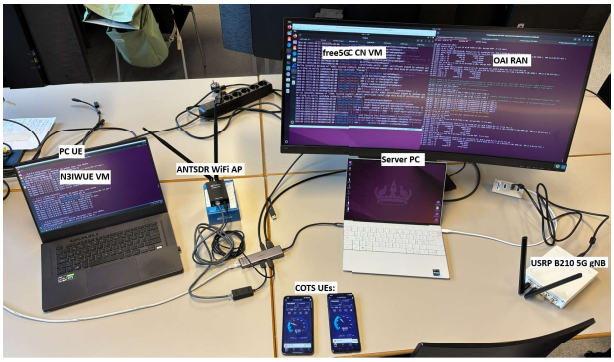


Fig. 4: The implemented system: Hardware and software components.

Motorola Edge 30 Fusion, an Android secret code to access hidden network settings works. For **Samsung Galaxy S21 FE,** a third-party application, `5G Switch`, is adopted.

Additionally, it is worth mentioning that the tested **Google Pixel 8** running Android 16 could not be connected to our testbed, since it does not expose the MNC/MCC and non-standalone (NSA)/SA configuration fields required for the custom PLMN.

D. Illustration of the Implemented System

Fig. 4 illustrates the implemented hardware-based testbed where the main network functions are distributed across separate components. In addition to the components presented above, the 5G NR and OpenWiFi SDRs are connected to the laptop through a USB, achieved by using two USB-C docking stations respectively.

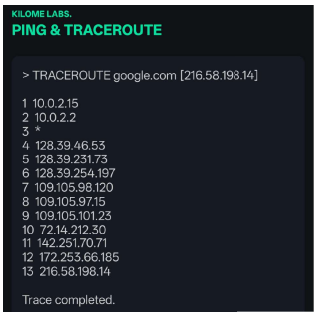
IV. Experiments and Numerical Results

Based on the implemented testbed, we have performed extensive real-life experiments in a lab environment located on UiA's campus in Grimstad, Norway. In this section, we present a few selected sets of experimental results with respect to global connectivity, network performance (throughput, latency, and packet loss), and application-level QoE.

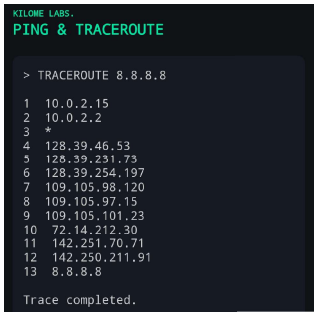
A. Verification of Global Connectivity

1) 3GPP connectivity: Through the PDU session establishment procedure towards free5GC, the connected UEs obtain IPv4 addresses in the `10.60.0.x` range. In Fig. 5, we illustrate the hop-by-hop path from two COTS UEs (OnePlus Nord 5 (`10.60.0.1`) and OnePlus 12 (`10.60.0.2`)) towards `google.com` respectively, obtained by `traceroute`.

As shown in the figure, along the path, **Hop 1 –** `10.0.2.15` is the NAT address of the free5GC VM. Thereafter **Hop 2 –** `10.0.2.2` is the VirtualBox NAT gateway. **Hop 3** is shown as *, meaning that no response is returned from that hop. In this setup, this corresponds to the server-side private communication path. The following hops show the traffic path through external networks. The `128.39.x.x` addresses belong to **SIKT,** the *Norwegian Agency for Shared Services in Education and Research*, and the `109.105.x.x` addresses belong to **NORDUnet**, *the Research and Education Networks in the Nordic countries*. The final destinations,



(a) Traceroute from OnePlus Nord 5.



(b) Traceroute from OnePlus 12.

Fig. 5: Traceroute from two COTS 5G UEs: From UE to google.com.

`216.58.198.14` (Amsterdam) and `8.8.8.8` (Mountain View, CA), belong to Google.

2) Non-3GPP connectivity: Fig. 6 illustrates the configuration of the N3IWUE WiFi STA. In our system, the derived operator code (OPc) value from the `Sysmocom SIM` information is adopted as the input to obtain the operator variant value (OP) value required by N3IWUE.

```
IMSI:
    PLMNID:
        MCC: 999
        MNC: 70
    MSIN: 0000119540
```

(a) N3IWUE UE IMSI.

```
Security:
    K: b73a90cbcf3afb622dba83c58a8415df
    RAND: b120f1c1a0102a2f507dd543de68281f
    SQN: 16f3b3f71005
    AMF: 8000
    OP: b672047e003bb952dca6cb8af0e5b779
    OPC: df0c67868fa25f748b7044c6e7c245b8
```

(b) N3IWUE UE security parameters.

Fig. 6: Configuration file for N3IWUE UE.

Fig. 7 depicts the `Wireshark` capture for the WiFi station connection establishment with free5GC. The screenshot includes IKE and encapsulating security payload (ESP) message exchanges between the WiFi `STA (enp0s9: 192.168.13.95)` and the bridge adapter `(enp0s9: 192.168.10.10)` which is directly connected with the N3IWF within free5GC, demonstrating that the secure communication path for N3IWUE is established. This screenshot confirms that *the non-3GPP access path is able to connect to the free5GC CN through the N3IWF*.

When executing `ping` or `traceroute` on the WiFi `STA` which has a source address `(enp0s9: 192.168.13.95)`, it reaches `(NAT enp0s3: 10.0.2.15)` first. Thereafter,

No.	Tin	Source	Destination	Protocol	Lengt	Info
1	…	192.168.13.95	192.168.10.10	ISAKMP	698	IKE_SA_INIT MID=00 Initiator Request
2	…	192.168.10.10	192.168.13.95	ISAKMP	698	IKE_SA_INIT MID=00 Responder Response
3	…	192.168.13.95	192.168.10.10	ISAKMP	214	IKE_AUTH MID=01 Initiator Request
4	…	192.168.10.10	192.168.13.95	ISAKMP	1446	IKE_AUTH MID=01 Responder Response
5	…	192.168.13.95	192.168.10.10	ISAKMP	182	IKE_AUTH MID=02 Initiator Request
6	…	192.168.10.10	192.168.13.95	ISAKMP	166	IKE_AUTH MID=02 Responder Response
7	…	192.168.13.95	192.168.10.10	ISAKMP	150	IKE_AUTH MID=03 Initiator Request
8	…	192.168.10.10	192.168.13.95	ISAKMP	150	IKE_AUTH MID=03 Responder Response
9	…	192.168.13.95	192.168.10.10	ISAKMP	182	IKE_AUTH MID=04 Initiator Request
10	…	192.168.10.10	192.168.13.95	ISAKMP	118	IKE_AUTH MID=04 Responder Response
11	…	192.168.13.95	192.168.10.10	ISAKMP	150	IKE_AUTH MID=05 Initiator Request
12	…	192.168.10.10	192.168.13.95	ISAKMP	278	IKE_AUTH MID=05 Responder Response
13	…	192.168.13.95	192.168.10.10	ESP	134	ESP (SPI=0x7f5e7ae9)
14	…	192.168.10.10	192.168.13.95	ESP	134	ESP (SPI=0x00000002)
15	…	192.168.13.95	192.168.10.10	ESP	134	ESP (SPI=0x7f5e7ae9)
16	…	192.168.10.10	192.168.13.95	ESP	198	ESP (SPI=0x00000002)
17	…	192.168.13.95	192.168.10.10	ESP	134	ESP (SPI=0x7f5e7ae9)
18	…	192.168.13.95	192.168.10.10	ESP	150	ESP (SPI=0x7f5e7ae9)
19	…	192.168.10.10	192.168.13.95	ESP	134	ESP (SPI=0x00000002)
20	…	192.168.13.95	192.168.10.10	ESP	182	ESP (SPI=0x7f5e7ae9)
21	…	192.168.10.10	192.168.13.95	ESP	134	ESP (SPI=0x00000002)
22	…	192.168.10.10	192.168.13.95	ESP	182	ESP (SPI=0x00000002)
23	…	192.168.13.95	192.168.10.10	ESP	134	ESP (SPI=0x7f5e7ae9)
24	…	192.168.10.10	192.168.13.95	ISAKMP	486	CREATE_CHILD_SA MID=00 Responder Request
25	…	192.168.13.95	192.168.10.10	ISAKMP	470	CREATE_CHILD_SA MID=00 Initiator Response
26	…	192.168.10.10	192.168.13.95	ESP	230	ESP (SPI=0x00000002)
27	…	192.168.13.95	192.168.10.10	ESP	134	ESP (SPI=0x7f5e7ae9)
28	…	192.168.13.95	192.168.10.10	ESP	166	ESP (SPI=0xd0984d58)
29	…	192.168.13.95	192.168.10.10	ESP	166	ESP (SPI=0xd0984d58)
30	…	192.168.13.95	192.168.10.10	ESP	198	ESP (SPI=0xd0984d58)

Fig. 7: Wireshark: Connection establishment WiFi⇔free5GC N3IWF.

TABLE III: COTS UE Performance by `Ookla Speedtest`

UE Model	DL [Mbps]	UL [Mbps]	Ping [ms]	Loss [%]
OnePlus 12	125.0	13.54	24.4	0
OnePlus Nord 5	127.0	15.10	25.8	0
Motorola Edge 30 Fusion	127.2	14.56	25.4	0

the path for global connectivity beyond free5GC CN (`NAT enp0s3: 10.0.2.15`) is the same as illustrated in Fig. 5. This result reveals that *both 3GPP and non-3GPP traffic goes to/from the Internet via the common free5GC CN.*

B. VM Resource Allocation for free5GC VM

The VM that hosts free5GC on Dell XPS 9320 can be equipped with various virtual central processing unit (vCPU) and random access memory (RAM) sizes. By sweeping vCPU cores (2, 4, 6, 8, 10, 12) and RAM (2048, 4096, 8192 MB) in both `Performance` and `Balanced` laptop power modes, we perform a set of experiments when connecting two COTS UEs to the gNB.

As a result, the `Performance` mode with 8 vCPU cores and 8192 MB RAM is selected for the subsequent experiments presented in the rest of this section. This VM configuration is selected because it provides the most stable CN behavior under high traffic load conditions, while also providing a consistent and repeatable operational environment for our experiments.

C. COTS 5G UEs: Data Rate, Latency, and Packet Loss

Based on the aforementioned configuration for the free5GC VM, we have conducted a set of performance assessments using the `Ookla Speedtest App`. These experiments are carried out by using *individually* three different COTS UE models and the results summarized in Tab. III are the average values obtained across 5 runs for each UE.

As shown in the table, the UEs achieve approximately 125–127 Mbps downlink (DL), 13–15 Mbps uplink (UL) data rates, `ping` latency 24–26 ms, and 0% packet loss. These results are reasonable given that the gNB is operated as a single cell in an indoor lab environment (band n78, 40 MHz channel bandwidth, 30 kHz subcarrier spacing (SCS) (numerology 1), and 106 physical resource blocks (PRBs).

D. Intra- and Inter-RAT Operability and Performance

To assess the interoperability and performance of our prototype system for both 3GPP and non-3GPP RATs, a set of experiments are conducted based on the *transmission control protocol (TCP) reverse mode* using `iperf3`. For these measurements, both single- and dual-UE 5G NR connections are involved, *with and without concurrent non-3GPP traffic*. Tab. IV and Tab. V illustrate the experiment configuration and the obtained experimental results respectively. Note that, due to dynamic IP address allocation in the free5GC CN, the IP addresses of the two UEs shown in Tab. IV are different from the ones reported in Subsec. IV-A.

In the single-UE case, UE 1 achieves 117 Mbps DL throughput on average (on the UE-side receiver, across 128 parallel streams). When a simultaneous N3IWUE `iperf3` session is active, the 5G UE throughput reduces to 100 Mbps, and the N3IWUE WiFi path achieves 18.5 Mbps (on the receiver side).

TABLE IV: Test configuration: two UEs + one WiFi STA

Item	OnePlus Nord 5	OnePlus 12	N3IWUE WiFi STA
Command host	`free5gc`		
Routing context	`vrf-pdu-1`		
PDU/IPv4 address	`10.60.0.12`	`10.60.0.13`	`10.60.0.7`
Remote iPerf3 host	`192.168.56.101`		
iPerf3 port	`5010`	`5020`	`7000`
Mode	TCP reverse mode, -R		
Parallel streams	`10, -P 10`		
Test duration	Approximately 10 s		

TABLE V: DL TCP throughput with intra-/inter-RAT connections

RAT: 5G only versus 5G + WiFi	UE 1 (OnePlus Nord 5)	UE 2 (OnePlus 12)	N3IWUE WiFi STA
1×5G UE only	117 [Mbps]	N.A.	N.A.
1×5G + WiFi	100 [Mbps]	N.A.	18.5
2×5G UE only	65.9 [Mbps]	65.8 [Mbps]	N.A.
2×5G + WiFi	57.1 [Mbps]	59.4 [Mbps]	13.8 [Mbps]

When two 5G NR sessions are undergoing in parallel (intra-RAT traffic), each UE achieves approximately 65.9 Mbps. With two 5G UEs and the N3IWUE path all active simultaneously (inter-RAT traffic), the obtained throughput falls down respectively to 57–59 Mbps per 5G UE and 13.8 Mbps on the WiFi path. These results reveal that 1) inter-RAT operability has been achieved by our prototype system, and 2) network capacity is shared across the active parallel sessions, however, unbalanced between 5G NR and WiFi connections.

E. Non-3GPP Path Segment Bottleneck Exploration

To understand the reasons why the WiFi connection achieves much lower throughput than the 5G NR connection, we explore whether the observed limitation originates from the CN, the RAN components, or shared host-system resources through another set of experiments.

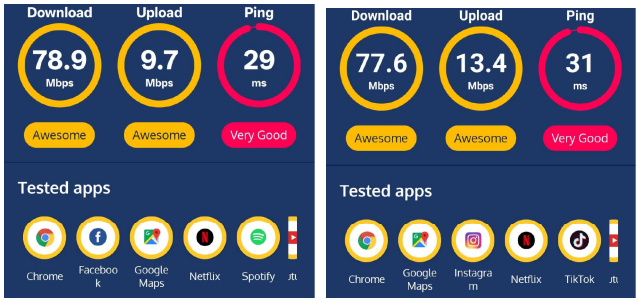
Tab. VI presents the *isolated* `iperf3` throughput for each segment of the non-3GPP communication path (averaged across 10 parallel TCP streams, without any simultaneous 5G traffic). The STA-to-AntSDR sole WiFi segment achieves the highest throughput (24.5 Mbps, on the receiver side). Meanwhile, both paths toward the free5GC VM are limited to approximately 14.4 Mbps on the receiver side, and the AntSDR-to-free5GC path experiences substantially higher number of retransmissions during the experiment. This result indicates that the N3IWF-side processing and the wired forwarding path constitute the primary bottleneck along the non-3GPP path towards the global Internet.

TABLE VI: Non-3GPP path segment bottleneck analysis

Segment	Sender	Receiver	Retrans.
WiFi STA → AntSDR	26.5 [Mbps]	24.5 [Mbps]	53
WiFi STA → free5GC VM	15.8 [Mbps]	14.4 [Mbps]	442
ANTSDR → free5GC VM	16.0 [Mbps]	14.4 [Mbps]	4664

F. QoE: Video Streaming and App Performance

Finally, we perform a set of application- and multimedia-oriented experiments from a user perspective based on the prototype system. The experiments are performed along the



(a) OnePlus Nord 5. (b) OnePlus 12.

Fig. 8: The results for streaming and App performance assessment.

5G OAI RAN – free5GC CN path using the `Ookla Video Test` benchmark and `Meteor App`.

For these experiments, two UEs (one OnePlus 12 and one OnePlus Nord 5) are connected to the system simultaneously, and the experimental results are illustrated in Fig. 8. As evident from the figure, both UEs achieve the maximum supported video resolution of 2160p (4K) with 0% buffering under simultaneous dual-device load. `Meteor` rates all the tested applications (streaming, gaming, social media, video calls) as `Awesome` on both UEs. These results demonstrate that the implemented 5G network is able to support popular multimedia applications with premium user QoE satisfaction.

V. Further Discussions

In this section, we share thoughts on the behavior of the inter-RAT 5G–WiFi testbed based on our experience.

A. Bottleneck Analysis: RAN versus CN

As shown in Tab. III, the 5G UE DL throughput (125–127 Mbps) is constrained by the 5G radio access path rather than the free5GC CN. When performing `Ookla Speedtest` directly from the free5GC VM, we obtain 936 Mbps for DL and 829 Mbps for UL respectively. As such, the most likely limiting factors are the OAI gNB processing load on the host CPU, the USB3-based USRP B210 data path, and the selected OAI radio configuration (106 PRBs). These components share CPU resources with the free5GC VM, creating a system-level constraint that cannot be attributed to any single component without independent variation.

B. Non-3GPP Path Limitations

As observed, the non-3GPP path throughput (14.4 Mbps to the free5GC VM) is significantly lower than the 5G NR DL throughput. The retransmission count escalation from 53 (for pure WiFi) to 4664 (for AntSDR→free5GC VM) suggests that the N3IWF-side forwarding path may contribute significantly to the experienced throughput limitation, instead of the SDR WiFi link itself.

Domain name system (DNS) resolution failures on the N3IWUE path (e.g., `Speedtest` does not work due to VRF) limit the applicability of N3IWUE in comparison with use cases with direct IP connection. Resolving this would require configuring DNS inside the VRF context or deploying a DNS proxy reachable from within the PDU session address space.

VI. Concluding Remarks

This paper presents the design, implementation, and evaluation of a fully operational free5GC-based end-to-end 5G prototype system that supports both 3GPP and non-3GPP RAT connections through one common 5GC, free5GC. The operation of our system is enabled by a small-scale testbed composed of free5GC, OAI RAN, OpenWiFi, SDR, COTS UEs, and N3IWUE hardware and software components. Using multiple networking tools including `Wireshark`, `ping`, and `traceroute`, we demonstrate that both 3GPP 5G NR access and non-3GPP WiFi access have been successfully established through the same 5GC. Through `speedtest`, and `iperf3`, we demonstrate that premium service satisfaction has been achieved. Our experiments also reveal that network performance and stability are affected by the available hardware resources, the use of VM-based components, and thermal behavior on the host platform. For further enhancement, one may consider to allocate dedicated hardware per component (separate machines for CN, RAN, and the non-3GPP gateway).

References

[1] *Access to the 3GPP 5G core network via non-3GPP access networks,* Technical Specification TS 24.502, R19, v19.4.0, 3GPP, Jun. 2026.

[2] A. Martian, Razvan F. Trifan, Teodora C. Stoian, M.-C. Vochin, and F. Y. Li, "Towards open RAN in beyond 5G networks: Evolution, architectures, deployments, spectrum, prototypes, and performance assessment," *Comput. Netw.*, vol. 259, art. no. 111087, pp. 1–19, Mar. 2025.

[3] J. L. Herrera, S. Montebugnoli, D. Scotece, L. Foschini, and P. Bellavista, "A tutorial on O-RAN deployment solutions for 5G: From simulation to emulated and real testbeds," *IEEE Commun. Surveys Tuts.*, vol. 28, pp. 1709–1748, Jun. 2026.

[4] T. Mukute, L. Mamushiane, A. A. Lysko, E.-R. Modroiu, T. Magedanz, and J. Mwangama, "Control plane performance benchmarking and feature analysis of popular open-source 5G core networks: OpenAir-Interface, Open5GS, and free5GC," *IEEE Access*, vol. 12, pp. 113336–113360, Aug. 2024.

[5] X. Jiao, W. Liu, M. Mehari, M. Aslam, and I. Moerman, "openwifi: a free and open-source IEEE802.11 SDR implementation on SoC," in *Proc. IEEE Vec. Technol. Conf. (VTC) Spring*, May 2020, pp. 1–2.

[6] D. Ergenç and F. Dressler, "An open source implementation of Wi-Fi 7 multi-Link operation in OMNeT++," in *Proc. 20th Wireless On-Demand Network Systems and Services Conference (WONS)*, Jan. 2025, pp. 131–134.

[7] K. Schneider, E. Lyczkowski, D. Rittershofer, W. Kiess, S. Treib, M. Dietrich, and S. Hoeltken, "The industrial 6G network of networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM) Workshops*, Dec. 2024, pp. 1–6.

[8] K. Schneider, S. Treib, M. Dietrich, W. Kiess, and E. Lyczkowski, "MultiNet6G: 5G NR, WiFi and visible light communication working as single network," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Sep. 2025, pp. 1–7.

[9] The free5GC Team, "free5GC: Open source 5G core network based on 3GPP Release 17," GitHub Repository. 2026. [Online]. Available: https://github.com/free5gc/free5gc

[10] M. Barbosa, M. Silva, E. Cavalcanti, and K. Dias, "Open-source 5G core platforms: A low-cost solution and performance evaluation," in *Proc. Int. Conf. Informat. Netw. (ICOIN)*, Jan. 2025, pp. 99–104.

[11] A. Markan, G. Kumar, and C. Markan, "Private 5G lab as a service (P5G-LaaS): Next generation communication platform for simulation and performance evaluation." In: *Proc. IEEE Microw., Antennas, Propag. Conf. (MAPCON)*, Dec. 2025, pp. 1–4.

[12] OpenWiFi, "open-source IEEE 802.11 WiFi baseband FPGA (chip) design: driver, software," GitHub Repository. 2026. [Online]. Available: https://github.com/open-sdr/openwifi/tree/master

[13] OpenAirInterface, "Project information: OpenAirInterface 5G wireless implementation," OAI GitLab. 2026. [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g