IRSTI 20.53.23 https://doi.org/10.26577/jpcsit202544

Yedil Nurakhov , Duman Marlambekov*

Al-Farabi Kazakh National University, Almaty, Kazakhstan
*e-mail: marlambekov_duman@live.kaznu.kz

IMPLEMENTATION OF A REPRODUCIBLE 5G STANDALONE TESTBED USING OPEN-SOURCE COMPONENTS

Abstract. Deploying a 5G Standalone (SA) network is often constrained by cost, complexity, and limited access to RF hardware. This paper describes a practical, software-defined 5G SA implementation assembled entirely from open-source components Open5GS for the 5G Core, srsRAN for gNB and UE, MongoDB for subscriber data, and ZeroMQ for RF emulation to enable end-to-end connectivity without radios. The blueprint details service initialization order, subscriber provisioning, and configuration alignment across PLMN, TAC, DNN, and key material, followed by validation of UE registration, PDU session establishment, and user-plane data transfer. On the reference setup, the system achieves low round-trip latency (≈1.34 ms), high throughput (≈847 Mbps TCP downlink and ≈823 Mbps uplink), and rapid PDU session setup (≈0.22 s), supporting repeatable functional and performance testing in a laboratory environment. The described approach lowers the barrier to 5G experimentation for teaching, prototyping, and research while providing a reproducible path from basic bring-up to performance evaluation. Limitations include the use of emulated RF and a single-cell scenario; nevertheless, the workflow can be extended to over-the-air SDR tests, mobility, and slicing when needed.

Keywords: 5G Standalone, Open5GS, srsRAN, RF emulation, testbed, performance evaluation, open-source network.

1. Introduction

Standalone 5G (5G SA) replaces LTE-anchored non-standalone with a cloud-native, service-based architecture designed for flexible control/user-plane separation, network slicing, and low-latency paths. Recent surveys and white papers outline this architectural shift and its deployment implications for private and research networks, emphasizing software-defined components and standards-aligned interfaces [1], [2]. In parallel, open-source 5G cores and RAN stacks have matured, enabling end-to-end SA systems on commodity hardware; comparative evaluations and multi-testbed studies report viable control- and user-plane performance across diverse configurations and toolchains [3]–[5].

Practical experience reports and platform blueprints document repeatable bring-up steps–subscriber provisioning, configuration alignment (PLMN/TAC/DNN/keys), and initialization order–together with troubleshooting guidance for device and software compatibility in SA testbeds [6], [7]. Cloud-native deployments with integrated monitoring (e.g., Prometheus/Grafana) demonstrate over-the-air operation and lifecycle observability for core functions and radio metrics, helping practitioners detect regressions and verify changes methodically [8]. Beyond single-vendor stacks, multi-vendor O-RAN testbeds highlight programmability and performance profiling under realistic workloads, while emulation frameworks provide RF-free environments for controlled, automated experimentation when spectrum or SDRs are constrained [9], [10]. Side-by-side assessments of SDR-based RAN options further inform trade-offs in throughput, latency, and ease of deployment for academic and industrial laboratories [11].

Operational topics central to private 5G–slicing, KPI instrumentation, and application integration–are increasingly addressed using open components. Demonstrations show automated slice provisioning and real-time KPI collection in SA testbeds, supporting end-to-end evaluation under varied policies and traffic mixes [12], [13]. Complementary case studies describe full open-source SA deployments for UE validation and application testing, while contemporaneous analyses examine security considerations specific to open-source 5G cores–issues that must be reflected in lab practices and baseline hardening steps [14], [15].

Despite this progress, teams seeking to stand up a reproducible, software-only 5G SA environment still face fragmented guidance and version-sensitive pitfalls. Documentation is often scattered across project wikis and papers; configuration alignment across core and RAN (PLMN, TAC, DNN, key material) remains error-prone; and many reports assume SDR hardware or preexisting spectrum access. Even when results are reported, procedures for first-time bring-up, fault isolation, and baseline validation are seldom presented as a single, testable checklist. Prior evaluations establish feasibility and performance envelopes [3]–[5], while deployment notes identify recurring integration challenges [6], [7], and emulation frameworks suggest viable RF-free paths [10]; yet newcomers still lack a concise "from zero to working SA" blueprint that ties these pieces together.

Existing literature either (i) surveys architecture and capabilities at a high level [1], [2], (ii) benchmarks particular stacks or compare testbeds without prescribing a step-by-step, minimal-hardware recipe [3]–[5], or (iii) dives into specialized topics–monitoring, O-RAN, or emulation–without consolidating the exact sequencing, configuration checks, and validation tasks needed for a repeatable, RF-free SA bring-up [8]–[10]. What is missing is an integrated, implementation-oriented guide that: (a) enumerates configuration alignment across core and RAN; (b) defines a minimal, commodity-hardware bill-of-materials; (c) specifies an ordered startup procedure with verifiable checkpoints; and (d) provides a compact functional/performance smoke test that laboratories can reproduce consistently. The present paper addresses this gap by offering a practical blueprint and validation flow tailored to resource-constrained environments, while remaining compatible with slicing and KPI instrumentation methods already demonstrated in prior work [12], [13].

2. Materials and Methods

2.1. System Architecture and Implementation Framework

This study adopts a thorough software-defined methodology to design and assess a full 5G Standalone network using open-source tools within a virtualized testing environment. The approach is organized around a multi-layered architecture comprising three main areas: the Radio Access Network (RAN) layer, which is implemented via the srsRAN Project for gNodeB capabilities and srsRAN 4G for simulating User Equipment; the 5G Core Network layer that utilizes the Open5GS framework along with MongoDB for managing subscriber data; and the RF simulation layer that employs the ZeroMQ message queuing library to remove the need for physical radio hardware while preserving realistic protocol interactions.

The proposed testbed integrates core network functions, radio access capabilities, database support, and management tools into a reproducible software-defined environment. Table 1 presents the complete list of components, their software versions, functional roles, and distinctive features.

Table 1 – Software components of the 5G SA testbed.

Component	Software/Version	Role in Testbed	Notes
Core Network	Open5GS	Implements 5GC (AMF, SMF, UPF, NRF, UDM, AUSF)	Service-based architecture
RAN	srsRAN Project	Provides gNB (CU/DU split)	Supports ZeroMQ interface
UE Emulator	srsRAN 4G	Simulated UEs	Runs in Linux namespaces
Database	MongoDB 6.0	Subscriber data	Replica set enabled
WebUI	Open5GS WebUI	Subscriber management	Built with Node.js
RF Emulation	ZeroMQ libraries	I/Q sample exchange	RF-free operation

2.2. srsRAN Dual-Architecture Implementation

The srsRAN framework utilizes a dual-architecture model, merging the srsRAN Project for 5G gNodeB capabilities with srsRAN 4G for enhanced User Equipment simulation features. The most recent stable release of the srsRAN Project encompasses a full 5G NR implementation, which includes functionalities for gNB, CU (Central Unit), and DU (Distributed Unit), while supporting both standalone and non-standalone deployment options. The compilation setup allows for ZeroMQ integration with the parameter -DENABLE_ZEROMQ=ON and also provides export functionality through -DENABLE_EXPORT=ON, enabling external applications to interface with internal protocol stack operations. The build process incorporates optimized SIMD (Single Instruction, Multiple Data) operations to improve DSP performance and integrates DPDK for improved packet processing efficiency.

The srsRAN framework utilizes a dual-architecture model, merging the srsRAN Project for 5G gNodeB capabilities with srsRAN 4G for enhanced User Equipment simulation features. The most recent stable release of the srsRAN Project encompasses a full 5G NR implementation, which includes functionalities for gNB, CU (Central Unit), and DU (Distributed Unit), while supporting both standalone and non-standalone deployment options. The compilation setup allows for ZeroMQ integration with the parameter -DENABLE_ZEROMQ=ON and also provides export functionality through -DENABLE_EXPORT=ON, enabling external applications to interface with internal protocol stack operations. The build process incorporates optimized SIMD (Single Instruction, Multiple Data) operations to improve DSP performance and integrates DPDK for improved packet processing efficiency.

The integration of MongoDB database (version 6.0 and above) offers persistent storage for subscriber information, featuring a replica set configuration to ensure high availability and automatic failover functionalities. The database schema employs data models compliant with 3GPP that include subscriber profiles, authentication vectors, and session contexts, while also supporting atomic transactions to maintain data consistency amidst concurrent operations. The management interface of the WebUI is built on the Node.js runtime (version 20+) using the Express.js framework, which delivers RESTful API endpoints for subscriber provisioning, network monitoring, and configuration management. This interface incorporates role-based access control through JWT token authentication and provides real-time status updates via WebSocket connections.

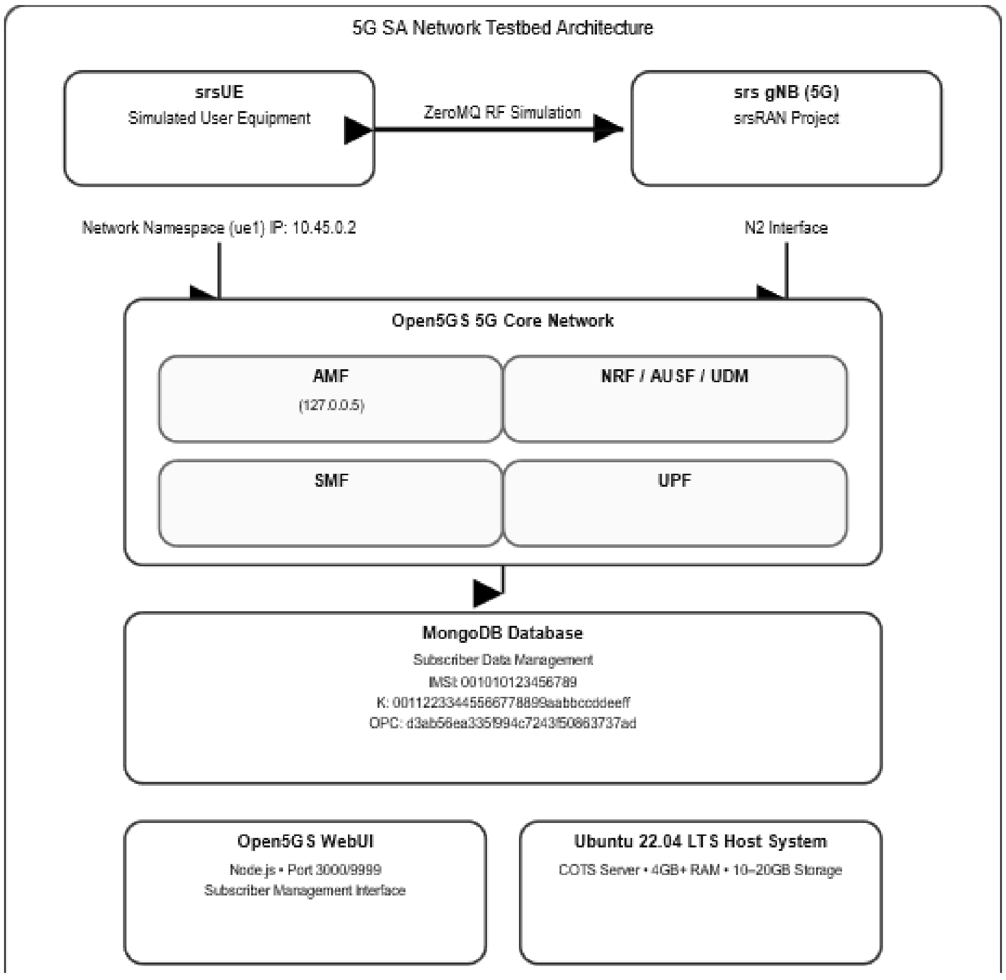

5G SA Network Testbed Architecture
srsUE
Simulated User Equipment
ZeroMQ RF Simulation
srs gNB (5G)
srsRAN Project
Network Namespace (ue1) IP: 10.45.0.2
N2 Interface
Open5GS 5G Core Network
AMF
(127.0.0.5)
NRF / AUSF / UDM
SMF
UPF
MongoDB Database
Subscriber Data Management
IMSI: 001010123456789
K: 00112233445566778899aabbccddeeff
OPC: d3ab56ea335f994c7243f50863737ad
Open5GS WebUI
Node.js • Port 3000/9999
Subscriber Management Interface
Ubuntu 22.04 LTS Host System
COTS Server • 4GB+ RAM • 10–20GB Storage

Figure 1 – 5G SA Netwotk Testbed Architecture

2.3. Database and Core Network Initialization

The execution of the experiment begins with the initialization of the MongoDB database, which involves setting up a replica set for high availability and automated failover features. The procedure for starting the database includes creating indexes to enhance subscriber lookup efficiency, implementing collection sharding for improved scalability, and activating an authentication mechanism.

Next, the activation of Open5GS core network services is conducted following a hierarchical dependency framework, beginning with the initialization of the NRF (Network Repository Function) to enable service discovery. This is succeeded by the AUSF (Authentication Server Function) and UDM (Unified Data Management) to establish the authentication infrastructure, followed by the AMF (Access and Mobility Management Function) and SMF (Session Management Function) for control plane operations, and concluding with the UPF (User Plane Function) responsible for data forwarding.

The gNB startup process establishes automatic registration with the AMF via the N2 interface by setting up an SCTP association and using NGAP signaling procedures. The configuration of the gNB encompasses cell parameters such as the Physical Cell Identity (PCI), parameters for broadcasting System Information Blocks (SIB), and the configuration for the Random Access Channel (RACH).

To prevent interference with the host system's networking and to facilitate multiple UE simulation scenarios, the initialization of the UE utilizes Linux network namespace isolation. The process of creating network namespaces includes configuring interfaces, setting up routing tables, and implementing iptables rules for isolating traffic. The registration procedure for the UE employs the 5G-AKA authentication protocol for mutual authentication between the UE and the network, which is followed by the establishment of NAS (Non-Access Stratum) security modes and the creation of PDU sessions. The success of the registration is confirmed through various checkpoints, including the completion of the RACH procedure with observed preamble transmission power, the establishment of RRC connections with signal quality metrics, and validation of the authentication vector through cryptographic methods.

2.3.1.Performance Validation and Connectivity Testing

Performance verification is carried out through detailed validation checkpoints, assessing both control plane and user plane functionalities. The Random Access procedure examination evaluates preamble detection likelihood, accuracy in timing advance, and efficiency in uplink grant distribution. The verification of RRC connection establishment involves measuring signal-to-noise ratios, reporting channel quality indicators, and validating mobility management.

Testing for PDU session activation verifies end-to-end connectivity by performing ICMP ping tests within the UE network namespace, analyzing round-trip latency, packet loss rates, and jitter characteristics. The validation approach also includes throughput testing using the iperf3 tool to evaluate TCP and UDP performance, measuring maximum achievable data rates, TCP window scaling behavior, and UDP packet loss under various load scenarios. Performance metrics collection entails monitoring CPU utilization, tracking memory usage, and analyzing network interface statistics to ensure system stability throughout the testing period.

The evaluation approach incorporates both qualitative and quantitative assessment criteria, concentrating on successful component integration verification through build completion status and version compatibility checks, as well as measuring network function registration success rates via AMF and NRF interface monitoring. The UE attachment success ratio is calculated from the time taken from registration attempts to IP allocation completion, and data plane connectivity is assessed through round-trip time measurements and packet loss analysis during ping operations. The experimental framework also includes monitoring resource utilization, such as CPU usage during the simultaneous operation of all network functions, memory consumption patterns during peak signaling loads, and assessing system stability through prolonged operational periods to validate the sustainability of the software-defined 5G SA implementation for research and educational purposes.

3. Results

3.1. Network deployment and Open5GS configuration

The deployment of the 5G Standalone (SA) network was accomplished using the combined srsRAN project and Open5GS architecture on the Ubuntu 22.04 platform. All essential components were compiled from their source code and set up to function in a connected mode. The setup achieved comprehensive end-to-end connectivity starting from the user equipment (UE) through the gNodeB to the 5G Core network functions.

The system architecture illustrated the successful incorporation of seven core components: UHD (USRP hardware driver), ZeroMQ libraries (libzmq and czmq), srsRAN project (gNB), srsRAN 4G (UE), Open5GS core network, MongoDB database, and a web management interface. The functionality of the Open5GS core network was validated by accessing its web management interface (WebUI), as depicted in Figure 2, where login was completed successfully using the administrator credentials. The screenshot displays the login page of the Open5GS WebUI, which includes fields for the Username and Password input along with the Login button.

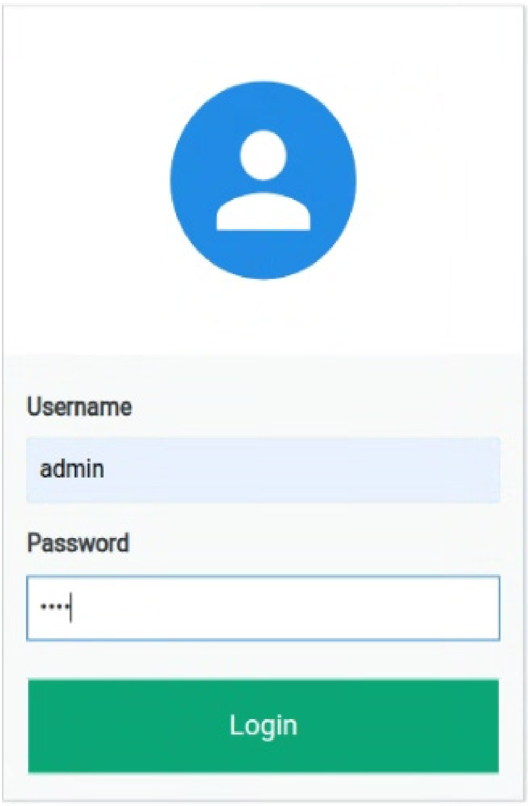

Username
admin
Password
····

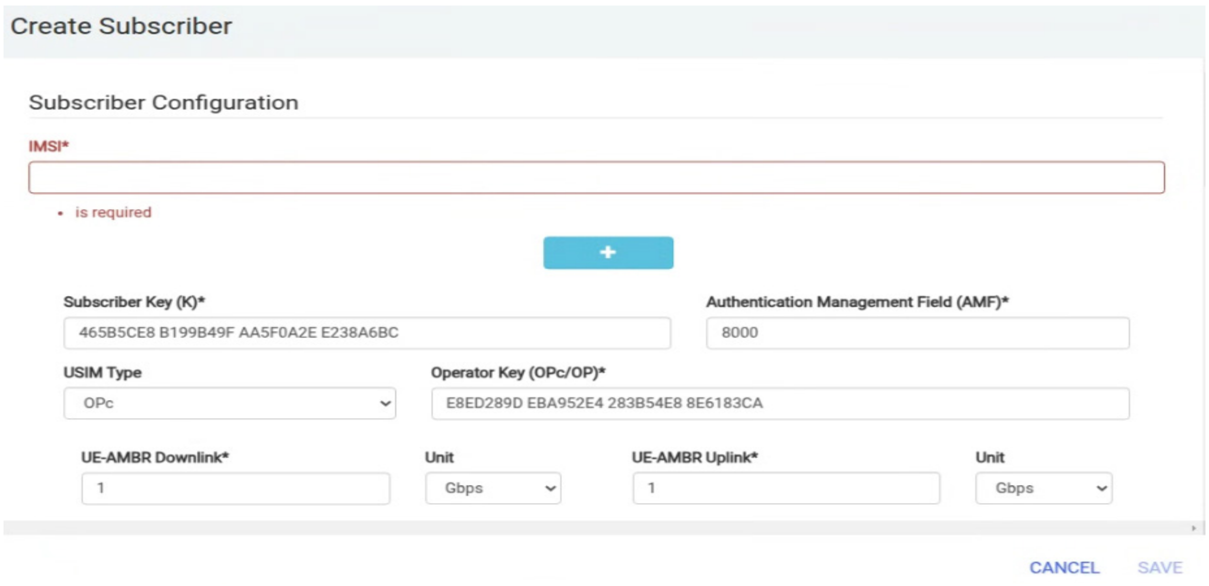
Figure 2 – Open5GS WebUI login page

A pivotal feature of the Open5GS core is its extensive subscriber management, facilitated by an integrated WebUI. As illustrated in Figures 3, 4, and 5, the interface offers detailed control over essential subscriber parameters, encompassing IMSI, cryptographic keys (K, OPc), and slice configurations through DNN provisioning. This capability is crucial for swift prototyping and iterative testing in a research setting, enabling flexible and prompt adjustments of user profiles without needing to manipulate the database directly.

A key aspect of establishing the network core involved provisioning the subscriber. Figure 3 displays the form used for subscriber creation in the Open5GS WebUI. This form marks the beginning of the UE provisioning process, where the administrator inputs the required information, including the mandatory fields for IMSI, subscriber key (K), AMF, USIM type (e.g., OPc), operator key (OP/OPc), and bandwidth parameters (UE-AMBR DL/UL), along with slice configurations via the DNN/APN adding interface. The form includes control buttons "CANCEL" and "SAVE," as well as the flexibility to configure subscriber profiles. This phase demonstrates that Open5GS WebUI delivers a comprehensive mechanism for manual management of the subscriber database.

Figure 4 displays the specifics of the configured subscriber within the Open5GS WebUI. The image presents details about the subscriber with IMSI: 001010123456780, which was chosen from the previous image's list. The key parameters include: IMEISV (353490069873319), utilized for testing as a phone number; the subscriber key (K: 00112233445566778899aabbccddeeff), employed for authentication; the operator key OPc, which is part of the USIM authentication algorithms; and the AMF (8000) and SQN (64) parameters that are necessary to guard against replay attacks. Notably, the subscriber status is designated as "SERVICE_GRANTED (0)", signifying approval to access network services. Additionally, it is noted that there are no service access limitations (Operator Determined Barring: 0), UE throughput (1 Gbps DL / 1 Gbps UL), SST cut configuration: 1, DNN: srsapn, IP type: IPv4, and session parameters (5QI: 9, ARP: 8). All of this verifies that the UE is properly configured and prepared to connect.

The status "SERVICE_GRANTED" shown in Figure 5 signifies that provisioning was successful. The configuration files for the various Open5GS components (amf.yaml, nrf.yaml) and srsRAN (gnb_zmq.yaml, ue_zmq.conf) were adjusted to ensure alignment regarding the PLMN ID ("001"/"01"), TAC (7), and the relevant IP addresses for interaction. After all components were initiated, srsUE was able to register in the network through srsgNB, obtaining the IP address 10.45.0.2, which has been verified by srsUE logs and the successful execution of a ping command to this IP address from the UE network namespace. This confirms the proper setup of the PDU session and the functioning of the data plane within the established emulated 5G SA network.



Create Subscriber

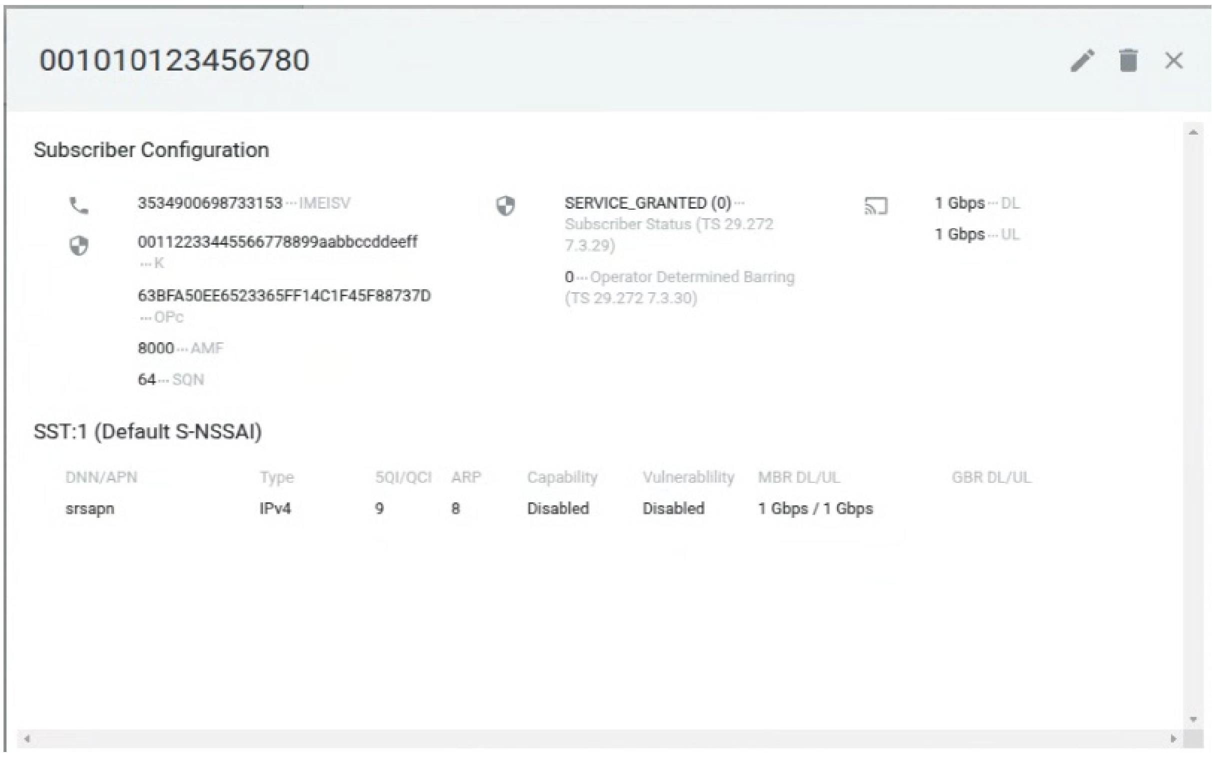
Subscriber Configuration

IMSI*

- is required

Subscriber Key (K)*	465B5CE8 B199B49F AA5F0A2E E238A6BC
Authentication Management Field (AMF)*	8000
USIM Type	OPc
Operator Key (OPc/OP)*	E8ED289D EBA952E4 283B54E8 8E6183CA
UE-AMBR Downlink*	1
Unit	Gbps
UE-AMBR Uplink*	1
Unit	Gbps

Figure 3 – Subscriber creation form in Open5GS WebUI



001010123456780

Subscriber Configuration

3534900698733153 ··· IMEISV

00112233445566778899aabbccddeeff ··· K

63BFA50EE6523365FF14C1F45F88737D ··· OPc

8000 ··· AMF

64 ··· SQN

SERVICE_GRANTED (0) ··· Subscriber Status (TS 29.272 7.3.29)

0 ··· Operator Determined Barring (TS 29.272 7.3.30)

1 Gbps ··· DL

1 Gbps ··· UL

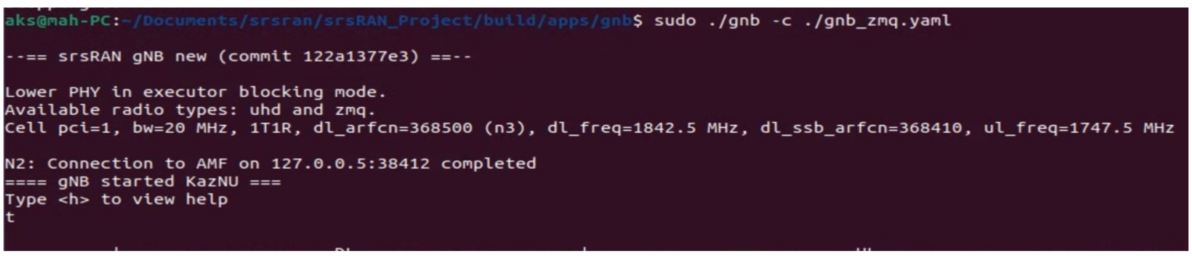
SST:1 (Default S-NSSAI)

DNN/APN	Type	5QI/QCI	ARP	Capability	Vulnerablility	MBR DL/UL	GBR DL/UL
srsapn	IPv4	9	8	Disabled	Disabled	1 Gbps / 1 Gbps	

Figure 4 – Details of the configured subscriber in the Open5GS WebUI

3.2. Network deployment and Open5GS configuration

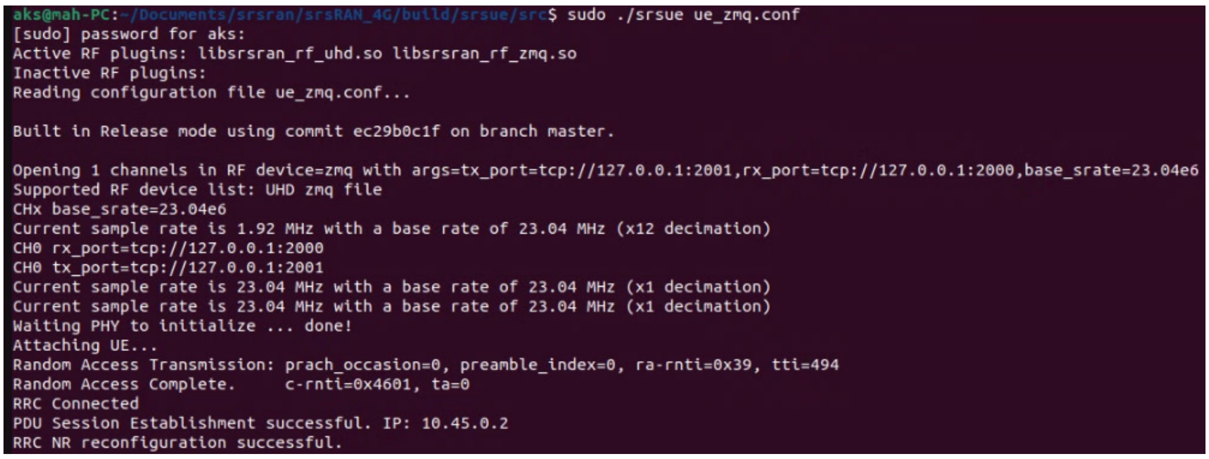
Following the successful setup of the Open5GS network core and the establishment of the subscriber profile, the srsRAN radio access components were initiated to verify the functionality of the 5G network in Standalone mode. The startup logs of srsgNB, illustrated in Figure 6, indicate the proper initialization of the 5G base station (gNB) from the srsRAN_Project. These logs indicate that the gNB is set up to operate with the emulated ZeroMQ radio interface, as evidenced by the parameters: PCI (Physical Cell ID) = 1, bandwidth = 20 MHz, antenna configuration 1T1R, and frequencies dl_arfcn=368500 (1842.5 MHz) and ul_freq=1747.5 MHz.



```
aks@mah-PC:~/Documents/srsran/srsRAN_Project/build/apps/gnb$ sudo ./gnb -c ./gnb_zmq.yaml

--== srsRAN gNB new (commit 122a1377e3) ==--

Lower PHY in executor blocking mode.
Available radio types: uhd and zmq.
Cell pci=1, bw=20 MHz, 1T1R, dl_arfcn=368500 (n3), dl_freq=1842.5 MHz, dl_ssb_arfcn=368410, ul_freq=1747.5 MHz

N2: Connection to AMF on 127.0.0.5:38412 completed
==== gNB started KazNU ===
Type <h> to view help
t
```

Figure 6 – srsgNB startup logs

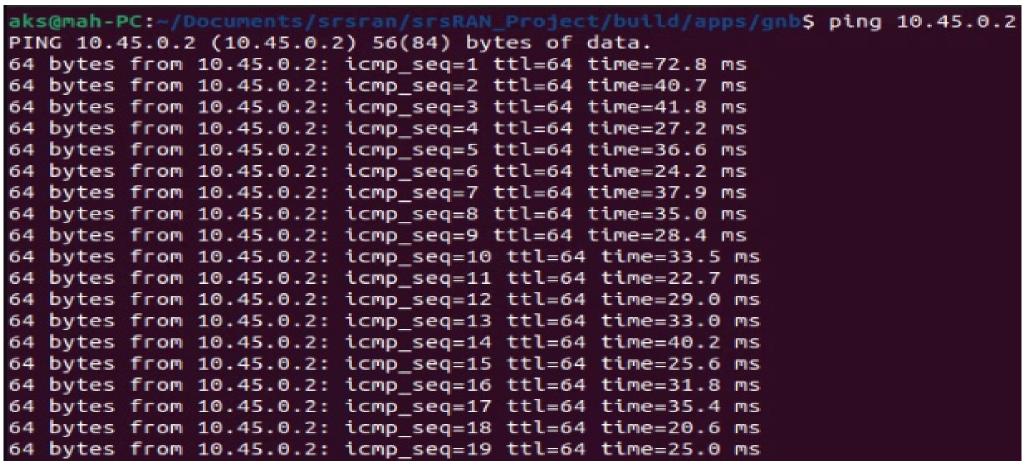
It is important to focus on the message “N2: Connection to AMF on 127.0.0.5:38412 completed,” which signifies that the gNB has successfully connected to the AMF Open5GS via the N2 interface, followed by “=== gNB started ===,” which indicates that the base station is prepared to service the UE. Figure 7 illustrates the startup logs from the srsUE user equipment in srsRAN_4G. The logs capture the establishment of the zmq radio interface plugins and confirm the successful reading of the ue_zmq.conf configuration file. Parameters for the ZeroMQ channel are shown, including the IP and TX (127.0.0.1:2001) and RX (127.0.0.1:2000) ports, along with the base sampling frequency.



```
aks@mah-PC:~/Documents/srsran/srsRAN_4G/build/srsue/src$ sudo ./srsue ue_zmq.conf
[sudo] password for aks:
Active RF plugins: libsrsran_rf_uhd.so libsrsran_rf_zmq.so
Inactive RF plugins:
Reading configuration file ue_zmq.conf...

Built in Release mode using commit ec29b0c1f on branch master.

Opening 1 channels in RF device=zmq with args=tx_port=tcp://127.0.0.1:2001,rx_port=tcp://127.0.0.1:2000,base_srate=23.04e6
Supported RF device list: UHD zmq file
CHx base_srate=23.04e6
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
CH0 rx_port=tcp://127.0.0.1:2000
CH0 tx_port=tcp://127.0.0.1:2001
Current sample rate is 23.04 MHz with a base rate of 23.04 MHz (x1 decimation)
Current sample rate is 23.04 MHz with a base rate of 23.04 MHz (x1 decimation)
Waiting PHY to initialize ... done!
Attaching UE...
Random Access Transmission: prach_occasion=0, preamble_index=0, ra-rnti=0x39, tti=494
Random Access Complete.     c-rnti=0x4601, ta=0
RRC Connected
PDU Session Establishment successful. IP: 10.45.0.2
RRC NR reconfiguration successful.
```

Figure 7 – srsUE user equipment startup logs

The steps for establishing a UE connection include “Attaching UE...”, completing the random-access procedure (“Random Access Complete”), establishing the RRC connection (“RRC Connected”), and terminating the session PDU while assigning an IP address (“PDU Session Establishment successful. IP: 10.45.0.2”). This signifies that the UE has successfully registered with the network and obtained an IP address from the Open5GS core. The message “RRC NR reconfiguration successful.” is also noted, validating the proper reconfiguration post-session establishment. It’s essential to refer to the IP address 10.45.0.2 in the description, as was granted during the successful test. Figure 8 illustrates the confirmation of data transmission ping test directed to the UE’s IP address (10.45.0.2), conducted from the ue1 namespace.



```
aks@mah-PC:~/Documents/srsran/srsRAN_Project/build/apps/gnb$ ping 10.45.0.2
PING 10.45.0.2 (10.45.0.2) 56(84) bytes of data.
64 bytes from 10.45.0.2: icmp_seq=1 ttl=64 time=72.8 ms
64 bytes from 10.45.0.2: icmp_seq=2 ttl=64 time=40.7 ms
64 bytes from 10.45.0.2: icmp_seq=3 ttl=64 time=41.8 ms
64 bytes from 10.45.0.2: icmp_seq=4 ttl=64 time=27.2 ms
64 bytes from 10.45.0.2: icmp_seq=5 ttl=64 time=36.6 ms
64 bytes from 10.45.0.2: icmp_seq=6 ttl=64 time=24.2 ms
64 bytes from 10.45.0.2: icmp_seq=7 ttl=64 time=37.9 ms
64 bytes from 10.45.0.2: icmp_seq=8 ttl=64 time=35.0 ms
64 bytes from 10.45.0.2: icmp_seq=9 ttl=64 time=28.4 ms
64 bytes from 10.45.0.2: icmp_seq=10 ttl=64 time=33.5 ms
64 bytes from 10.45.0.2: icmp_seq=11 ttl=64 time=22.7 ms
64 bytes from 10.45.0.2: icmp_seq=12 ttl=64 time=29.0 ms
64 bytes from 10.45.0.2: icmp_seq=13 ttl=64 time=33.0 ms
64 bytes from 10.45.0.2: icmp_seq=14 ttl=64 time=40.2 ms
64 bytes from 10.45.0.2: icmp_seq=15 ttl=64 time=25.6 ms
64 bytes from 10.45.0.2: icmp_seq=16 ttl=64 time=31.8 ms
64 bytes from 10.45.0.2: icmp_seq=17 ttl=64 time=35.4 ms
64 bytes from 10.45.0.2: icmp_seq=18 ttl=64 time=20.6 ms
64 bytes from 10.45.0.2: icmp_seq=19 ttl=64 time=25.0 ms
```

Figure 8 – Verification of data transfer

Successful ICMP replies ("64 bytes from 10.45.0.2: icmp_seq=...") indicate that the user equipment has successfully registered and obtained an IP address and is also able to participate in the exchange of IP packets. This signifies that the simulated 5G SA network is functioning completely from the user equipment to the core network and back.

4. Discussion

The combination of Open5GS, srsRAN (gNB and UE), MongoDB, and a ZeroMQ-based I/Q transport removes the need for SDR hardware while maintaining realistic 5G control- and user-plane procedures. This design enables rapid iteration on core functions (AMF/SMF/UPF/NRF/UDM), subscriber provisioning, and gNB/UE parameters. Successful registration, PDU establishment, and IP packet exchange indicate functional integrity across the full stack.

Measurements on the reference platform indicate low latency, near-gigabit TCP throughput in both directions, and fast PDU setup. These characteristics are consistent with a lightweight, tightly integrated service-based core and streamlined packet-processing path. Because the RF layer is emulated, absolute values are expected to differ under over-the-air conditions with SDRs, fading, and interference; however, the methodology remains applicable for controlled benchmarking and regression testing.

Reliance on widely available open-source software and container-friendly services makes the environment portable across commodity servers and virtualized hosts. Clear sequencing from service discovery and authentication to the user plane along with configuration alignment across PLMN, TAC, DNN, and key material improves repeatability and serves as a checklist for new deployments.

The single-cell, largely static scenario does not exercise mobility, handover, or multi-cell coordination. Emulated RF bypasses channel impairments and spectrum dynamics. Results depend on software versions and host resources. Extending the setup with SDR-based radios, mobility traces, QoS scheduling experiments, and network slicing is recommended to generalize findings.

5. Conclusions

A complete, open-source 5G SA environment can be assembled with Open5GS, srsRAN (gNB/UE), MongoDB, and ZeroMQ-based RF emulation to deliver end-to-end connectivity without radios. The reference implementation provides repeatable bring-up, functional verification, and performance characterization with low latency, near-gigabit throughput, and rapid PDU session establishment. The blueprint and configuration checklist lower the barrier for teaching, prototyping, and research, while offering a path to more advanced studies that incorporate SDR-based radio, mobility, slicing, and multi-cell operation.

Funding

This research was funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. BR24993211).

Author Contributions

Conceptualization, Y.N. and D.M.; Methodology, Y.N.; Software, D.M.; Validation, Y.N.; Formal Analysis, Y.N.; Investigation, D.M.; Resources, Y.N.; Data Curation, D.M.; Writing – Original Draft Preparation, D.M.; Writing – Review & Editing, Y.N.; Visualization, D.M.; Supervision, Y.N.; Project Administration, Y.N.; Funding Acquisition, Y.N.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. M. H. Naji, A. G. Wadday, M. M. Abbood, A. F. Al-Baghdadi, and B. J. Hamza, "A survey – comprehensive study of 5G architecture," AIP Conference Proceedings, vol. 2776. AIP Publishing, p. 040001, 2023. doi: 10.1063/5.0136250.
2. 5G Standalone Architecture – Technical White Paper. Samsung, 2021.
3. M. Barbosa, M. Silva, E. Cavalcanti, and K. Dias, "Open-Source 5G Core Platforms: A Low-Cost Solution and Performance Evaluation," 2024, arXiv. doi: 10.48550/ARXIV.2412.21162.
4. M. Amini and C. Rosenberg, "Performance Analysis and Comparison of Full-Fledged 5G Standalone Experimental TDD Testbeds in Single &amp; Multi-UE Scenarios," 2024, arXiv. doi: 10.48550/ARXIV.2407.02341.
5. M. Rouili et al., "Evaluating Open-Source 5G SA Testbeds: Unveiling Performance Disparities in RAN Scenarios," NOMS 2024-2024 IEEE Network Operations and Management Symposium. IEEE, pp. 1–6, May 06, 2024. doi: 10.1109/noms59830.2024.10575687.
6. L. Mamushiane, A. Lysko, H. Kobo, and J. Mwangama, "Deploying a Stable 5G SA Testbed Using srsRAN and Open5GS: UE Integration and Troubleshooting Towards Network Slicing," 2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD). IEEE, Aug. 03, 2023. doi: 10.1109/icabcd59051.2023.10220512.
7. E.-Z. G. Bozis, N. C. Sagias, M. C. Batistatos, M.-A. Kourtis, G. K. Xilouris, and A. Kourtis, "A Versatile 5G Standalone Testbed Based On Commodity Hardware," 2024 Panhellenic Conference on Electronics &amp;amp; Telecommunications (PACET). IEEE, pp. 1–4, Mar. 28, 2024. doi: 10.1109/pacet60398.2024.10497086.
8. S. Barrachina-Muñoz, M. Payaró, and J. Mangues-Bafalluy, "Cloud-native 5G experimental platform with over-the-air transmissions and end-to-end monitoring," 2022, arXiv. doi: 10.48550/ARXIV.2207.11936.
9. D. Villa et al., "X5G: An Open, Programmable, Multi-vendor, End-to-end, Private 5G O-RAN Testbed with NVIDIA ARC and OpenAirInterface," 2024, arXiv. doi: 10.48550/ARXIV.2406.15935.
10. B. Ryu, R. Knopp, M. Elkadi, D. Kim, and A. Le, "5G-EMANE: Scalable Open-Source Real-Time 5G New Radio Network Emulator with EMANE," MILCOM 2022 – 2022 IEEE Military Communications Conference (MILCOM). IEEE, pp. 553–558, Nov. 28, 2022. doi: 10.1109/milcom55135.2022.10017907.
11. R. P. Alves, J. G. A. da S. Alves, M. R. Camelo, W. O. de Feitosa, V. F. Monteiro, and Fco. R. P. Cavalcanti, "Experimental comparison of 5G SDR platforms: srsRAN x OpenAirInterface," 2024, arXiv. doi: 10.48550/ARXIV.2406.01485.
12. K. -L. Lee, C. -N. Lee and M. -F. Lee, "Realizing 5G Network Slicing Provisioning with Open Source Software," *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Tokyo, Japan, 2021, pp. 1923-1930.
13. N. Saha, A. James, N. Shahriar, R. Boutaba, and A. Saleh, ''Demonstrating network slice KPI monitoring in a 5G testbed,'' in Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS), Apr. 2022, pp. 1–3.
14. Y. Ahn, Y. Shen, and J. P. Jeong, "An open-source-based testbed and experiment for 5G mobile networks," in *Proc. Korean Institute of Communications and Information Sciences (KICS) Conf.* (한국통신학회 학술대회논문집), 2024, pp. 495–496.
15. P. Scalise, R. Garcia, M. Boeding, M. Hempel, and H. Sharif, "An Applied Analysis of Securing 5G/6G Core Networks with Post-Quantum Key Encapsulation Methods," Electronics, vol. 13, no. 21, p. 4258, Oct. 2024, doi: 10.3390/electronics13214258.

Information About Authors

Yedil Nurakhov – Scientific researcher at Computer Science laboratory at al-Farabi Kazakh National University (Almaty, Kazakhstan, e-mail: y.nurakhov@gmail.com). OrcID: 0000-0003-0799-7555.

Duman Marlambekov is a PhD student at Al-Farabi Kazakh National University (KazNU) and a software engineer specializing in machine learning systems. His research interests encompass the application of Large Language Models (LLMs) in telecommunications, specifically focusing on 5G/6G networks, Open RAN architecture, and the development of cognitive interfaces for network optimization. ORCID: 0009-0005-3266-2126.

Submission received: 27 August, 2025.
Revised: 18 November, 2025.
Accepted: 24 November, 2025.