Traffic analysis and forecasting for adaptive network resource management in 5G/6G networks

Adaptability and Latency in Network Reconfigurations of Virtualized Network Functions in 5G Networks

Georgescu Calin-Stefan

Supervisors: Nitinder Mohan, Marco Colocrese

EEMCS Faculty
Delft University of Technology
The Netherlands

A thesis submitted in partial fulfilment of the requirements
for the Bachelor of Computer Science and Engineering

Delft, June 22, 2025

Name of student: Georgescu Calin-Stefan
Course: CSE3000 Research Project
Thesis committee: Nitinder Mohan, Marco Colocrese (Supervisor), Guohao Lan (Examiner)

An electronic version is available at
`http://repository.tudelft.nl/`

Abstract

This paper investigates the latency and resilience of user-plane anchor reconfiguration in a fully virtualized 5G core environment using Open5GS and UERANSIM. The experiment spans five VirtualBox virtual machines, each hosting a key component of the 5G core or radio stack: 5G-core gNB, UPF1, UPF2, and a single UE. All nodes communicate over a shared internal network, ensuring control- and user-plane traffic remains isolated from external variability.

The UE is initially anchored to UPF 1 via DNN "internet." After the initial tunnel is established and validated, a re-anchoring procedure is triggered by calling the SMF's REST API. Although the endpoint is intended to perform a PFCP Session Modification, Open5GS tears down the session and creates a new one on UPF 2 instead. By analyzing timestamped UE logs—capturing tunnel setup, session release, and re-establishment—we measure the latency of user-plane reattachment.

Our results reveal high variability in recovery times, ranging from sub-second to over 50 seconds. These inconsistencies are attributed to limitations in Open5GS's session handling, the lack of true migration support, and hardware limitations of the used machine. Despite these challenges, the study offers insights into the practical behavior of PFCP-driven anchor reconfiguration and the operational gaps that remain in open-source 5G core implementations.

1 Introduction

In the rapidly evolving domain of telecommunications, 5G networks serve as the backbone for an unprecedented variety of applications. These factors demand adaptable, efficient, and continuous connectivity. With 6G on the horizon, these networks are set to become even more intricate, bolstering their capabilities to meet fluctuating user demands and a broader spectrum of services, by being fully cloud native. Essential to this adaptability is the deployment of Virtualized Network Functions (VNFs), which can be rapidly instantiated or migrated across diverse environments like cloud services or local networks [1]. The ability of VNFs, such as User Plane Functions (UPFs), to respond dynamically to changing network conditions is critical for maintaining optimal service performance and quality.

However, this potential for flexibility is frequently accompanied by challenges regarding latency and its impact on user experience during network reconfigurations. Specifically, issues arise in terms of how swiftly and seamlessly VNFs can adapt to new traffic demands and workloads, especially when migrating services mid-session. Existing frameworks claim to uphold session continuity during such transitions, yet practical assessments of their effectiveness and efficiency remain inadequately explored. A crucial aspect of preserving service reliability is latency [2], but it is unclear how launching or migrating UPFs affects the ongoing user sessions.

Previous research in the field defined a Session Reassignment Cost [3], but an actual measurement of this cost was never calculated. Other papers focus on the performance of the UPF regarding resource consumption and throughput [4], rather than latency during migrations.

This research aims to precisely examine the adaptability and latency associated with dynamic VNF reconfigurations in 5G networks, focusing on how quickly these virtualized services can respond to shifting demands, as well as offering a measurement for the Session Reassignment Cost. This research question centers on the speed of adaptation in virtual network services within a 5G environment, with sub-questions addressing the specific factors contributing to latency in UPF launches or migrations, and the behavior and performance impact on user sessions during these transitions.

This study encompasses a detailed analysis of UPF deployment and migration times under various conditions. This also offers both quantitative measures and visualizations that show network behavior and user impact during reconfigurations.

2 5G Core Architecture

In order to get a clear picture of the setup, it is important to understand the architecture of a 5G core network.

Figure 1 depicts the architectural layout of a 5G core network. The service-based interfaces represent the Control Plane of the 5G network, while the non service based interfaces represent the User Plane of the network. The UE represents the user equipment used (such as a phone or laptop) and the RAN represents the antenna to which it is connected. Each Network Function (NF) works as an independent microservice, each with its own purpose and capabilities. The key functions of a 5G core network are:

- **Access and Mobility Management Function (AMF)**: Manages registration, connection, reachability, and mobility for UEs. It terminates NAS signaling and interfaces with RAN via NGAP.
- **Session Management Function (SMF)**: Controls session establishment, modification, and release. Selects and configures UPFs and exposes RESTful APIs for managing user sessions.
- **User Plane Function (UPF)**: Handles packet routing and forwarding, selects anchor points for inter-RAT mobility, and implements traffic detection. UPFs are placed at the edge for low-latency paths.
- **Network Repository Function (NRF)**: Maintains NF profiles and supports service discovery. This acts like a lookup table for all the other NFs.
- **Unified Data Repository (UDR)**: Stores subscriber data (e.g., subscription, policy control data) in a unified, scalable database. It provides RESTful interfaces for data access.
- **Authentication Server Function (AUSF)**: Performs UE authentication and security key agreement, interacting with UDM for subscriber credentials.
- **Unified Data Management (UDM)**: Manages subscriber information (e.g., profiles, user identifiers) and interacts with UDR and AUSF.

Inter-NF communication uses standardized service-based interfaces (SBI) over HTTP/2; the NG-CF reference point connects RAN and AMF. By virtualizing all NFs on a single host, we can flexibly reconfigure or scale individual components and directly measure control- and user-plane performance under varying loads.

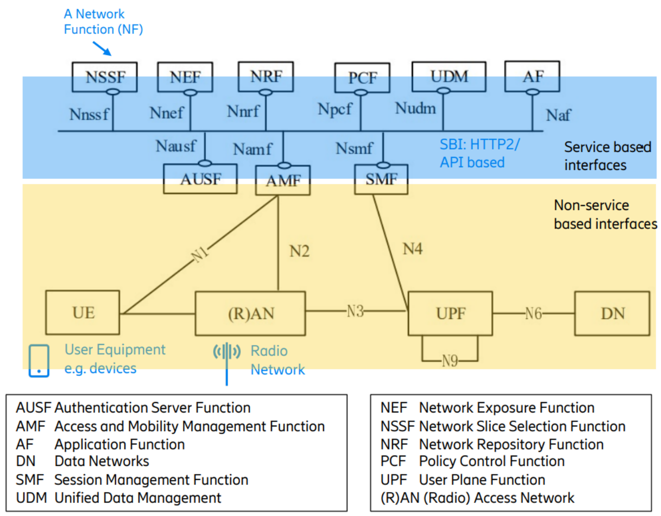

Figure 1: 5G Core Network Architecture

3 Methodology

To evaluate how rapidly virtualized 5G core functions can adapt to changes in user-plane infrastructure, we deploy Open5GS and UERANSIM across five separate Ubuntu 24.04 virtual machines, each hosting a distinct role in the 5G architecture.

All network functions are compiled from source and executed natively on each VM. An internal network named "5g-net" is created between the virtual machines to allow isolated communication between the networks.

At the beginning of each trial, the UE is configured with the DNN of "internet", the same DNN the UPF's have, and establishes a PDU session via UPF1. This process is confirmed by inspecting UERANSIM logs, where a successful tunnel creation message marks the session start. The same DNN was chosen for both UPF's to allow migration of traffic between them.

To simulate user-plane failure and trigger migration, we forcibly terminate the UPF1 process. This causes the SMF to detect a lost PFCP heartbeat and eventually release the PDU session. The UE reacts by automatically initiating a new session request. Because both UPFs have the same DNN, SMF is capable of easily choosing UPF2 to host the new session of the user.

We do not rely on ICMP traffic or packet capture, because previous studies have shown that they are not truly suitable if the protocol is not TCP/UDP [5]. Instead, we analyze timestamps from the UE's internal logs to determine:

- The time of initial tunnel creation
- The time the old session was released
- The time the new session was successfully established

This log-based method allows precise measurement of the session downtime during anchor reallocation. By repeating the experiment across multiple runs with varying background load, we quantify the delay introduced by this disruption and assess its variability.

4 Experimental Framework for Measuring Adaptability in Virtualized 5G Networks

To ensure both repeatability and environmental isolation, we construct our entire 5G core and RAN testbed using five interconnected VirtualBox virtual machines running Ubuntu 24.04 LTS. Each VM hosts a distinct network component, and all machines are connected via a VirtualBox internal network with static addressing for deterministic routing and communication. No container orchestration (e.g., Docker or Kubernetes) is used.

The components are distributed across the following VMs:

- **VM-CORE** (192.168.100.1): Hosts the AMF, SMF, MongoDB, and the Open5GS web UI.
- **VM-UPF1** (192.168.100.5): Runs UPF1, the initial anchor point for the UE's PDU session.
- **VM-UPF2** (192.168.100.6): Runs UPF2, which is used as the failover target during anchor migration.
- **VM-GNB** (192.168.100.4): Hosts the UERANSIM gNB component.
- **VM-UE** (192.168.100.3): Runs a single UE simulated via UERANSIM.

All components are installed according to their official documentation. We use:

- **Open5GS v2.7.5** for the core network components (AMF, SMF, UPFs)
- **UERANSIM v3.2.7** for the gNB and UE
- **MongoDB** as the backend data store for Open5GS, it's part of the Open5GS installation.

In order for the network to accept user sessions, it needs to be configured with a subscriber. This needs to be added to the database, using the WebUI. Then, this subscriber ID will be used as part of the UE's configuration, so the network will accept this UE and create sessions for it.

To accommodate multiple UPFs, we disable the default "open5gs-upfd" service and manually launch two separate instances of the UPF binary—each with its own configuration file and binding to separate IP addresses and ports. Each UPF announces its PFCP and GTP-U endpoints to the SMF via PFCP association.

Figure 2 illustrates the logical connections and traffic flows among the five VMs. All communications occur within the internal 192.168.100.0/24 network, with traffic routing and migration behavior strictly confined to this environment.

All services are managed via background shells or custom scripts rather than init systems or orchestrators, providing fine-grained control over timing, logging, and failure injection (e.g., stopping UPF1 to force migration).

5 Experimental Results

This section outlines the experimental procedures and observations used to evaluate the behavior of a 5G core network deployment based on Open5GS and UERANSIM. Then it outlines the results generated using this procedure.

5.1 Experimental Process

Each experiment proceeds in two structured phases:

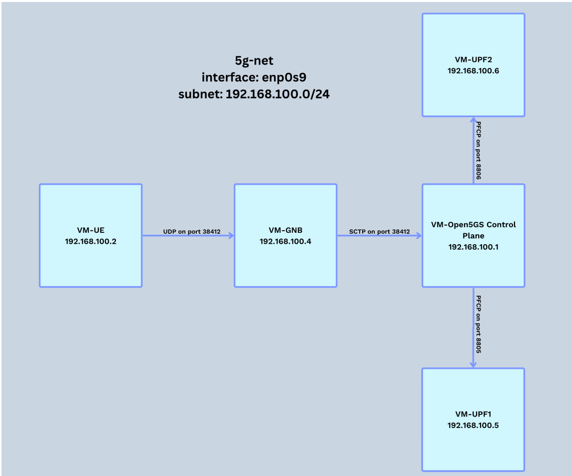

Figure 2: Traffic flow in Virtualized 5G deployment

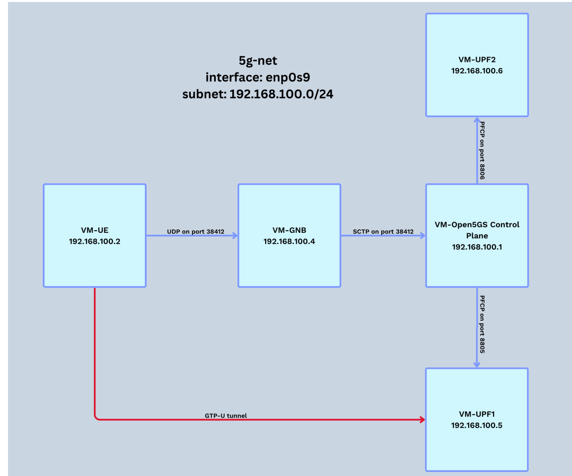

Figure 3: Traffic flow in Virtualized 5G deployment, after PDU session is established

1. **Baseline Session Establishment and Tunnel Observation** We begin by launching all five virtual machines. The control-plane functions are started first on `VM-CORE` (192.168.100.1). We then sequentially activate the user-plane and RAN components:
   - **UPF 1** is started on `VM-UPF1` (192.168.100.5), registering via PFCP to the SMF.
   - **The gNB** is brought online on `VM-GNB` (192.168.100.4), followed by **the UE** on `VM-UE` (192.168.100.2).

   Once the UE completes its registration and PDU session setup, the SMF establishes a GTP-U tunnel between the UE and UPF 1. Figure 3 shows this new tunnel used to send the traffic to the network. This association is visible in the UE logs, confirming successful connectivity. At this point, the data plane is fully functional and the UE's traffic is anchored on UPF 1.
   We then launch **UPF 2** on `VM-UPF2` (192.168.100.6), which also registers to the SMF but remains idle until migration. After that, we patch the MTU of the newly created interface, `ogstun2`. It is automatically created with an MTU of 1500, but the SMF is configured with an MTU of 1400, so UPF2 wouldn't be able to hold the session without this change.
2. **UPF Failure and Anchor Reallocation** To simulate a migration event, we forcibly terminate the UPF 1 process. The SMF detects the PFCP de-association and responds by establishing a new session on UPF 2. This leads to the teardown of the existing tunnel and the setup of a new one anchored on UPF 2. The UE receives a `PDU Session Release Command`, followed by re-initiation of the session and creation of a new tunnel—confirmed by the UE logs and tunnel device reset. The new traffic flow can be seen in Figure 4
   Each reconnection event is timestamped to measure downtime between the initial session's release and the new tunnel's activation. This duration constitutes the service disruption interval caused by the migration.

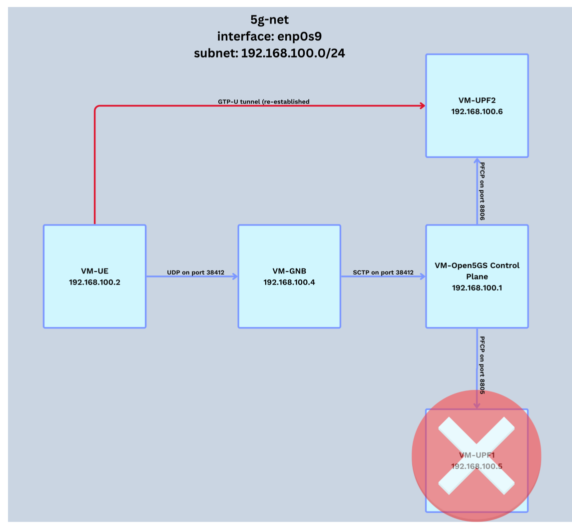

Figure 4: Traffic flow in Virtualized 5G deployment, after UPF1 is torn down

All UE logs and SMF-UPF traces are recorded locally on each VM. Repeating the experiment multiple times under identical conditions yields varying reconnection delays, highlighting non-determinism in the recovery process. Results are presented in the next section with a focus on:

- Time taken for the SMF to detect and respond to the UPF 1 failure
- Time between session release and re-establishment at the UE
- Variability across trials, ranging from near-instant recovery to reconnection delays exceeding 40 seconds

After manually running this experiment, a Python script was created to automate the process and run multiple experiments without further intervention. The script was executed inside the first VM containing the control plane, because the envinronment was completely isolated and the host machine had no access to the machines. The Paramiko Python library was used to create SSH connections and interactive terminals, allowing us to run the services and execute the experiments.

5.2 Results

Our experiments reveal that PFCP-triggered user-plane migration in Open5GS does not preserve the original session context. Instead of migrating the existing PDU session between UPFs, the SMF initiates a full teardown of the initial session and establishes a new one anchored on the target UPF. This results in a temporary loss of connectivity, observable as either ICMP ping disruption or session reinitialization delays in the UE logs.

We conducted multiple migration trials and recorded the time elapsed between the release of the initial session and the recreation of the tunnel towards UPF2. This latency captures both the teardown-to-setup transition and any SMF/UE signaling delays.

A total of 50 latency samples were collected from our automated experiments. The results are summarized as follows:

- **Minimum latency:** 0.259 seconds
- **Maximum latency:** 55.009 seconds
- **Average (mean) latency:** 18.040 seconds
- **Sample standard deviation:** 13.810 seconds

These results highlight **inconsistency and volatility** in the duration of user-plane handover. Although some migrations completed in less than a second, others experienced latencies exceeding 50 seconds. This variability is attributed to a combination of factors:

- The inability to maintain the existing tunnel during migration, requiring full UE session reestablishment.
- Delays in SMF-to-UPF PFCP reassociation.
- Kernel-level interface (e.g., `ogstun2`) creation and MTU configuration delays on the receiving UPF.
- Variable UE behavior upon receiving a `PDU Session Release Command`.

Although the results confirm that Open5GS can recover from UPF failure and resume connectivity through another UPF, the process is not seamless. There is no true live migration, only disconnection and reattachment.

6 Discussion

6.1 Real-World Expectations vs Implementation Behavior

In commercial 5G deployments, user-plane relocation is ideally handled through SSC Mode 3[6], where ongoing sessions persist across anchor migrations. However, current technologies are not prepared to handle SSC Mode 3, so the current 5G network tear down the session and create a new one.

Although Open5GS exposes a REST API endpoint for modifying a session's UPF anchor, which would allow SSC mode 3, our experiments confirmed that this path is not fully implemented in the SMF logic. Internally, no PFCP Session Modification messages are issued, and instead the SMF proceeds with PFCP Session Deletion followed by a new Session Establishment on the new UPF. This indicates that Open5GS does not yet support true user-plane migration as defined in 3GPP TS 29.244.

6.2 Inconsistencies and Code-Level Fragility

We observed erratic behavior in several areas of the Open5GS stack:

- The PFCP heartbeat logic sometimes fails to re-establish sessions after an anchor switch, particularly when the second UPF is started post-registration.
- UPFs fail silently if they cannot write to the log directory, resulting in misleadingly successful startup messages.
- If an old UPF process remains active in the background, SMF may associate to it instead of the intended instance, causing confusion and inconsistent session state.
- Some of the events that the AMF sends to SMF are not recognized by the SMF, despite this being the internal code logic.

These code-level fragilities—along with the missing session modification support—significantly limit Open5GS's suitability for controlled migration testing without patching or extending the core.

6.3 Testbed Constraints and Latency Variation

Despite careful configuration, the limitations of our testbed—five VirtualBox VMs sharing a single laptop CPU—introduced a layer of unpredictability that was difficult to control. We observed that even small variations in VM boot order or background I/O activity could significantly alter experiment outcomes. Some trials completed flawlessly, with the UE re-establishing its session on the second UPF within 200 milliseconds. Other times, recovery dragged on for tens of seconds, or failed entirely as the second UPF lost PFCP state due to missed heartbeats.

These discrepancies were not due to protocol flaws, but rather systemic instability under load. The migration logic, PFCP signaling, and GTP-U tunnel recreation mechanisms remained identical across trials. What changed was the responsiveness of the virtualized environment—its ability to process state transitions and keep timers in sync under CPU contention.

In light of this, we interpret the minimum observed recovery time as a valid representation of the protocol's performance under idealized conditions—free from other bottlenecks. It reflects the best-case latency achievable with the current implementation, had the infrastructure not been strained. Rather than dismissing slower trials as outliers, we treat them as evidence of the broader challenge: reliable migration requires not just correct logic, but also stable execution environments.

6.4 Implications

Despite these challenges, the experiment confirms that under certain ideal timing conditions, a UE can successfully reconnect after an anchor change with minimal delay. However, the lack of proper migration handling in Open5GS and the fragility of PFCP management underscore the gap between research prototypes and production-grade 5G cores. Robust session migration remains an open engineering challenge.

7 Responsible Research

We conducted all experiments in a fully controlled simulated 5G environment using Open5GS and UERANSIM. All traffic was synthetically generated—no real user data were involved, so privacy and compliance concerns are eliminated.

In line with responsible research practices, the test environment was fully isolated from any public or production networks. All components operated on virtual machines within an internal, non-routable network (5g-net), ensuring that no external entities were affected during testing.

The entire setup was created according to the official documentation of the tools, with minimal changes to allow it to be easily reproduced. The only changes made were the addition of an extra UPF, and the change of the IP addresses and ports to match the IP's assigned to the virtual machines. Furthermore, the UE and gNB configuration had to match the subscriber data that had to be manually added using the WEBUI of Open5GS.

Public AI tools were also used to help configure all of the components, as well as creating the python script required to automate the experiments.

8 Conclusions and Future Work

8.1 Conclusion

In this study, we deployed Open5GS and UERANSIM across five VirtualBox VMs to evaluate the adaptability of user-plane relocation in a virtualized 5G environment. Rather than performing true session migration, we observed that Open5GS tears down the existing PDU session and creates a new one when the SMF's REST API is used to switch UPF anchors. Despite the exposed endpoint, the SMF does not issue PFCP Session Modification messages, and therefore cannot maintain continuous session state across anchors.

By analyzing UE logs, we measured the latency between the release of the initial tunnel and the establishment of the new one. Results showed considerable variation—from sub-second recoveries to reconnection delays exceeding 50 seconds—due to both Open5GS's internal limitations and hardware constraints imposed by running five VMs on a single laptop. These findings highlight the fragility of current open-source implementations under dynamic reconfiguration, and the need for further robustness in PFCP session management to enable seamless anchor migration.

8.2 Future Work

A natural extension of this work is to revisit a more realistic, fully containerized deployment—specifically, running each Open5GS network function (AMF, SMF, UPF 1, UPF 2) and UERANSIM as separate Kubernetes pods. In that scenario, each NF would live in its own pod, and the SMF's DNN-to-UPF mapping could be exposed via Kubernetes Services, allowing us to evaluate:

- **Container startup and scaling delays:** Measure how quickly new UPF replicas spin up when traffic spikes, and how fast the SMF can detect and reconfigure PFCP associations.
- **Network Orchestration:** Orchestrate all of the microservices with kubernetes, which can more easily help us recreate or restart independent NF's, as well as storing sessions details.
- **Network-policy impact on GTP-U:** Investigate how Kubernetes NetworkPolicies (or CNI plugins) affect GTP-U throughput and latency during migration events.
- **Resilience and failover:** Simulate pod failures (e.g., evicting UPF 1) and measure how rapidly the SMF automatically switches to a newly scheduled UPF 2.

To accomplish this, we will:

(1) Formalize Kubernetes manifests for each NF (AMF, SMF, UPF1, UPF2) with appropriate `Deployment`, `Service`, and SCTP/CNI configurations.
(2) Automate the creation of `ConfigMaps` for each JSON/YAML configuration, enabling on-the-fly changes to DNN mappings.
(3) Implement a grooming script (or Helm chart) that orchestrates pod startup order (e.g., MongoDB → AMF → SMF → UPFs → UERANSIM) and verifies readiness via liveness probes.
(4) Repeat the migration experiment at scale—e.g., 5, 10, 20 concurrent UEs—to observe how Kubernetes scheduling latency and CNI overhead affect PFCP handover times.

Ultimately, moving to a Kubernetes-based deployment will allow us to assess how container orchestration layers influence the responsiveness and resilience of a virtualized 5G core in production-like settings.

References

[1] Peter Rost, Albert Banchs, Ignacio Berberana, Markus Breitbach, Mark Doll, Heinz Droste, Christian Mannweiler, Miguel A. Puente, Konstantinos Samdanis, and Bessem Sayadi. Mobile network architecture evolution toward 5g. *IEEE Communications Magazine*, 54(5):84–91, 2016.

[2] K.X. Du, G. Carrozzo, M.S. Siddiqui, O. Carrasco, B. Sayadi, F. Lazarakis, A. Kourtis, J. Sterle, and R. Bruschi. Definition and evaluation of latency in 5g: A framework approach. In *2019 IEEE 2nd 5G World Forum (5GWF)*, pages 135–140, 2019.

[3] Irian Leyva-Pupo, Cristina Cervelló-Pastor, Christos Anagnostopoulos, and Dimitrios P Pezaros. Dynamic upf placement and chaining reconfiguration in 5g networks. *Computer Networks*, 215:109200, 2022.

[4] Whai-En Chen and Chia Hung Liu. High-performance user plane function (upf) for the next generation core networks. *IET Networks*, 9(6):284–289, 2020.

[5] Cristel Pelsser, Luca Cittadini, Stefano Vissicchio, and Randy Bush. From paris to tokyo: On the suitability of ping to measure latency. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 427–432, 2013.

[6] Eungha Kim and Young-il Choi. Session and service continuity visualization monitoring system for 5g network. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 995–1000, 2020.

[7] Open5GS Contributors. Open5gs: Open source 5g core and epc. https://open5gs.org/, 2024. Accessed: 2025-05-23.

[8] UERANSIM Contributors. Ueransim: Open source 5g ue and ran simulator. https://github.com/aligungr/UERANSIM, 2024. Accessed: 2025-05-23.