Use of Open5Gs for 5G simulation and ETSI MEC API testing

1st Shobhit Gupta
BTECH CSE
IIIT Vadodara
202151149

2nd Archit Verma
BTECH CSE
IIIT Vadodara
202151192

3rd Dipean Dasgupta
BTECH CSE
IIIT Vadodara
202151188

4th Rahul Rathore
BTECH CSE
IIIT Vadodara
202151126

***Abstract*—Using Open5GS and UERANSIM, this project simulates 5G networks and assesses connectivity, protocol exchanges, and the functionality of 5G core networks. The 5G core network is implemented compliantly by Open5GS, and gNodeB and user equipment are emulated by UERANSIM. The study also investigates the ETSI MEC Sandbox for testing Multi-access Edge Computing (MEC) APIs, such as RNIS, AMS, and MEC011, in a variety of network settings. Advances in edge computing, low-latency service delivery, and smooth 5G connectivity are highlighted by the integration and experimentation, which leads to successful real-time data processing and service continuity certification.**

***Index Terms*—Open5GS, UERANSIM, 5G Simulation, ETSI MEC Sandbox, 5G Core Network, gNodeB, User Equipment (UE), RNIS, AMS.**

I. Introduction

The rise of 5G has revolutionized wireless communication with ultra-low latency, high data rates, and massive connectivity, driving advancements in IoT, smart cities, and autonomous systems. This project utilizes Open5GS and UERANSIM for 5G network simulation and MEC API testing in the ETSI MEC Sandbox.

Open5GS enables cost-effective simulation of 4G/5G core networks, while UERANSIM emulates User Equipment (UE) and gNodeB functionalities for evaluating network interactions. The ETSI MEC Sandbox extends this by enabling low-latency service testing through APIs like MEC011 and RNIS across various network scenarios, highlighting edge computing's role in 5G innovation.

A. *Open5Gs*

Open5GS is an open-source implementation of the core network for 4G LTE and 5G systems. It is designed to be compliant with the 3GPP specifications and provides all the core network functions needed to build a fully functional 4G or 5G network. Open5GS supports both Evolved Packet Core (EPC) for LTE and 5G Core (5GC) for standalone 5G deployments. The key components of Open5GS include:

- MME (Mobility Management Entity): Handles signaling between the mobile devices and the network.
- SGW (Serving Gateway) and PGW (PDN Gateway): Provide data routing and forwarding.
- AMF (Access and Mobility Management Function), SMF (Session Management Function), and other functions for 5G.

Open5GS is widely used in research, educational environments, and testing labs to simulate 4G/5G networks. It allows developers and engineers to test and deploy end-to-end mobile networks without needing expensive commercial solutions.

B. *UERANSIM*

UERANSIM (User Equipment and Radio Access Network Simulator) is an open-source simulator for 5G networks. It allows users to simulate both the UE (User Equipment) and RAN (Radio Access Network) aspects of a 5G system. UERANSIM connects with core networks like Open5GS to simulate the behavior of mobile devices and the radio access interface in a 5G environment.
Key features of UERANSIM include:

- UE Simulation: Simulates a 5G mobile device (phone or IoT device) that can perform registration, session management, and data transfer.
- gNodeB Simulation: Simulates the 5G RAN (gNodeB), which is responsible for communication between the UE and the core network.

UERANSIM is used for testing and experimenting with 5G features, helping researchers and developers understand how UEs interact with 5G core networks, all in a simulated environment.

Use Cases
Open5GS and UERANSIM are often used together to simulate complete end-to-end 5G networks. Open5GS acts as the core network, while UERANSIM simulates the user devices and the radio access interface, making them ideal for testing, research, and development of 5G technology.

C. *MEC ETSI API*

*1) **Overview of Edge Computing**:* Edge computing is a distributed computing paradigm that reduces latency and bandwidth consumption by moving computation and data storage closer to the point of demand. The concept of edge computing is particularly relevant in 5G networks, where low-latency connectivity is necessary for autonomous driving, gaming, and real-time video processing. By processing data closer to the end user, at the network's edge, MEC enables faster decision-making and more efficient use of network resources.

2) MEC Architecture: The foundation of MEC architecture is the idea of placing computing power close to user equipment (UE) at the network's edge. The following are the main elements of MEC architecture:

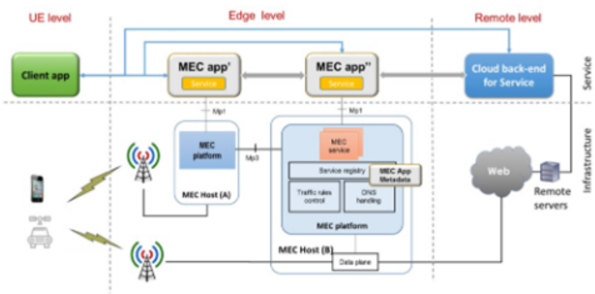

Fig. 1. New application development paradigm introduced by MEC.

Fig. 1. MEC Architecture

- **Client Applications:** Client applications refer to software running on user devices such as smartphones, tablets, or IoT devices. These applications utilize MEC services to access low-latency functionalities, perform real-time data processing, and offload computationally intensive tasks to edge nodes. By leveraging MEC capabilities, client applications achieve superior performance in latency-sensitive use cases like gaming, video streaming, and autonomous driving.
- **Device Applications:** Device applications are software components installed on edge devices such as base stations, gateways, or local servers. These applications act as intermediaries, managing the flow of data between client applications and the MEC platform. By ensuring smooth communication and minimal processing latency, device applications play a vital role in optimizing user experiences and maintaining robust connectivity.
- **MEC Applications:** MEC applications are hosted at the network's edge, providing diverse services to client applications. These services range from content delivery and augmented reality to video processing and analytics. The strategic placement of MEC applications ensures high-performance computing with reduced latency, enabling seamless operation in bandwidth-intensive scenarios.
- **Cloud-Based Backend Services:** MEC applications extend their capabilities through cloud-based backend services installed at the network edge. These services encompass tasks such as analytics, content delivery, and augmented reality. By processing data locally and reducing dependency on centralized cloud infrastructure, MEC architecture supports high-quality, latency-sensitive services tailored to user needs.

While meeting the demanding performance requirements of new 5G use cases, this modular approach in MEC architecture enables a wide range of applications.

II. System Design and Implementation

A. *Open5Gs and UERANSIM*

To set up the AMF (Access and Mobility Management Function) part in Open5GS and integrate it with UERANSIM, you are correct that the AMF needs to communicate with the gNodeB via NGAP (over SCTP) and the gNodeB connects to the UE (User Equipment) via RRC. The step-by-step guide for setting up the AMF in Open5GS and linking it to UERANSIM is shared below:

1) Setup Open5Gs(AMF): In Open5GS, the AMF handles UE registration, mobility, and authentication procedures. The AMF communicates with the gNodeB over SCTP through NGAP protocol. Here's how to configure the AMF part of Open5GS:

Configuration of AMF

- Locate AMF Configuration File: The AMF configuration file is typically found in open5gs/config/amf.yaml. This file configures the IP addresses, ports, and other settings for the AMF.
- Set AMF IP and SCTP Port: The AMF listens on IP 127.0.0.5 (as per your setup) and on port 38412 for NGAP over SCTP communication.

Restart Open5GS Components: After configuring AMF, restart the Open5GS services to apply the configuration.

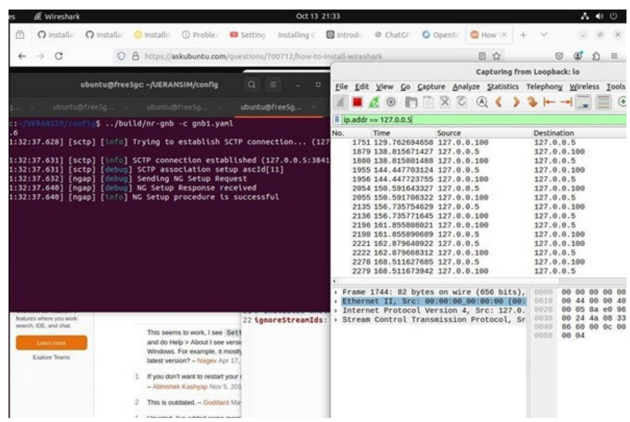

Fig. 2. Open5Gs Setup

2) Setup UERANSIM (gNodeB and UE): The UERANSIM tool is utilized to simulate both the gNodeB and User Equipment (UE) in a 5G network environment. The following steps outline the procedure for building and configuring these components:

- **Build UERANSIM:** To begin, clone the UERANSIM repository from its official source and build the project on your system. This step ensures that all necessary components of the simulator are compiled and ready for deployment.
- **Configure gNodeB**: The gNodeB in UERANSIM is configured to establish a connection with the AMF (Access and Mobility Management Function) using the NGAP protocol over SCTP. This involves setting up the

appropriate parameters in the gNodeB configuration file (gnb.yaml).

- **Configure UE**: Similarly, the UE is simulated by configuring its corresponding file (ue.yaml).
- **Wireshark Integration**: To capture and analyze packets exchanged during the simulation, a Wireshark extension is installed. Since the UE is simulated, it cannot directly catch packets. The RLS Wireshark Dissector extension is employed to overcome this limitation, enabling packet analysis and enhancing visibility into the NGAP and SCTP protocol exchanges.

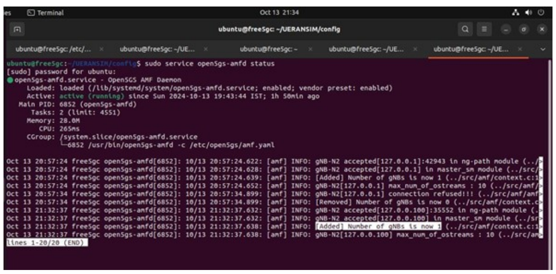

Fig. 3. gnodeB connection

In the initial screenshot, the destination IP address is shown as 127.0.0.5 for the first request. This port is predefined; however, if modifications are required, it can be configured in the AMF section. Please note that any changes to this configuration will necessitate corresponding updates in the linked and associated files to ensure proper connectivity.
Now, lets observe what happens when a UE is created:

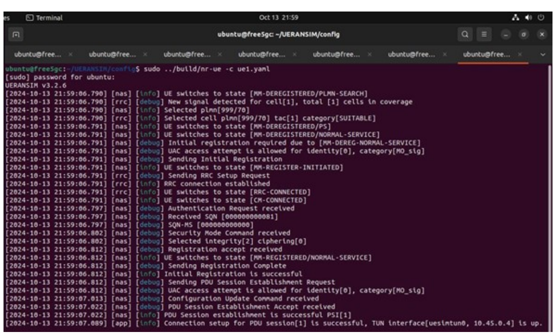

Fig. 4. UE creation

The image depicts the simulation of a 5G User Equipment (UE) using the UERANSIM software. The terminal logs depict the process of establishing a connection with the 5G core network. Key steps include PLMN selection, RRC connection setup, registration procedure, authentication, and PDU session establishment. The final output confirms a successful session setup with the TUN interface active, indicating proper UE functionality in the simulated 5G environment.
Now as UE is successfully created, all the packets and protocols captured by Wireshark can be seen.

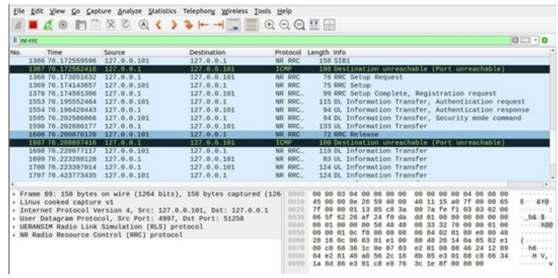

Fig. 5. Reflection of UE creation in Wireshark

Now, lets observe the call flow after second request: So,

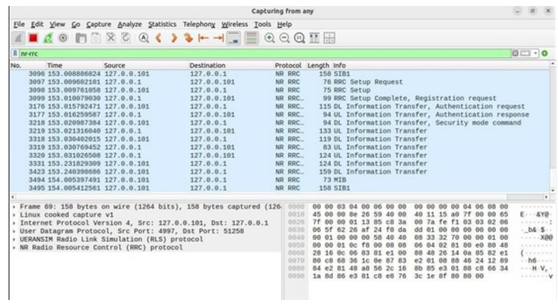

Fig. 6. 5G device connection to gnodeB

from the 2nd image, it can be observed that the simulation of 5G device connection to gnodeB has been successful and capturing packets and protocols.

B. ETSI MEC Testing

*1) **ETSI MEC Sandbox Setup:**:* A virtual environment called the ETSI MEC Sandbox is intended for testing MEC services and applications. Devel0opers may install and test their apps in a variety of network settings within its regulated environment. The sandbox enables thorough testing and assessment of MEC applications by providing pre-configured network settings, terminal devices, and MEC services.
The ETSI MEC Sandbox environment, which is modelled after the city of Monaco has been used for the test purposes. This setting offered a realistic and intricate network situation including WiFi access points, 4G and 5G macro cells, and V2X (Vehicle-to-Everything) networks. Attempts to install MEC apps, setup and oversee the network infrastructure, and track real-time performance of those apps were successful.

*2) **Network Scenarios Deployment:**:* Several network scenarios were deployed in the sandbox environment as part of the experimental phase. The purpose of these scenarios was to evaluate the scalability and performance of MEC applications in various circumstances.

- 4G Macro Network: Testing MEC applications in a conventional 4G network environment was the main goal of the 4G macro network scenario. Several base stations, user equipment (UE), and edge nodes were dispersed around the city in this scenario. The MEC apps underwent testing to ensure they could manage mobility, handle heavy traffic loads, and provide low-latency services.

- 4G-WiFi Macro Network: We added WiFi access points to the network in the 4G-WiFi macro network scenario, giving user equipment an extra layer of connection. The MEC apps were put to the test in this scenario to see how well they could maintain service continuity, smoothly transition between 4G and WiFi networks, and optimize resource utilization based on available connection choices.
- 4G-5G-WiFi Macro Network: By combining 4G, 5G, and WiFi networks, the 4G-5G-WiFi macro network scenario produced a heterogeneous environment with a variety of connection choices. Because the MEC applications had to handle various QoS levels, manage several radio access technologies (RATs), and adjust to quickly changing network conditions, this scenario was very difficult.
- 4G-5G Macro V2X Network: Testing MEC applications in a Vehicle-to-Everything (V2X) communication environment was the main goal of the 4G–5G macro V2X network scenario. High-velocity user equipment (like cars) in this situation has to continue having low-latency connectivity with the MEC applications. The scenario evaluated the apps' capacity to maintain high mobility in a V2X environment, provide real-time services, and guarantee safety-critical communications

*3) **Device and Service Management:**:* During the experimental phase, various Multi-access Edge Computing (MEC) platform functions were controlled to evaluate their performance and behavior under diverse conditions. These functions included service lifecycle management, application instance management, and terminal device configuration. The details of these processes are outlined below:

- Configuration of Terminal Devices: Terminal devices were configured within a sandbox environment to simulate different types of user equipment (UE). These included high-mobility UEs such as vehicles and stationary UEs such as smartphones. This setup allowed for the evaluation of MEC applications' performance across various network scenarios and mobility conditions. The configuration enabled testing under a range of simulated use cases to assess application responsiveness and adaptability.
- MEC Application Instance Management: The management of MEC application instances encompassed the deployment, scaling, and termination of applications across multiple edge nodes. Experiments were conducted using both distributed and centralized deployment strategies to analyze their impact on resource utilization and application performance. Real-time monitoring capabilities within the sandbox environment facilitated the tracking of performance metrics, enabling adjustments to optimize resource allocation and evaluate different deployment options.
- Control of MEC Services Lifecycle: The lifecycle management of MEC services included operations such as service upgrades, activations, and terminations. The experiments aimed to understand the dynamic management of MEC services in response to changing network conditions and user demands. Using APIs for service lifecycle management, various configurations were tested to assess their influence on application performance and overall system efficiency. These tests provided insights into the adaptability of MEC services in dynamic operational scenarios.

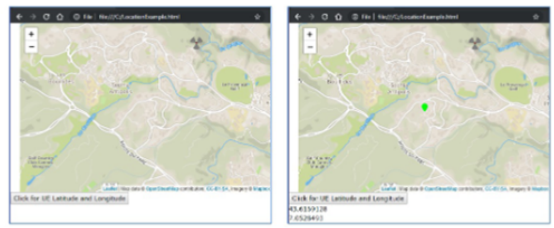

Fig. 7. API testing using ETSI MEC

*4) **Experimentation with MEC Service APIs:**:* To evaluate the functionality and performance of MEC applications, extensive testing of the APIs provided by the ETSI MEC Sandbox were conducted. The experimentation focused on understanding the APIs' features and their impact on application efficiency and responsiveness. The following APIs were utilized and assessed:

- MEC011 - Edge Platform Application Enablement (Mp1): The MEC011 API, also known as the Mp1 API, is a fundamental interface for enabling MEC applications to interact with edge platform services. It provides capabilities such as resource management, communication, and service discovery. Using the MEC011 API, applications were able to interface with the MEC platform, request network resources, and exchange data with other services, thereby enhancing their ability to utilize edge computing capabilities effectively.
- Radio Network Information Service (RNIS): The RNIS API provides real-time radio network information, including signal strength, cell details, and interference levels. This API was integrated into MEC applications to optimize their performance based on the current radio environment. For instance, by leveraging the RNIS API, applications dynamically adjusted their transmission power to maintain connectivity under challenging radio conditions, ensuring uninterrupted performance.
- Location Service API: This API provides precise location data for user equipment (UE), enabling the development of location-based services. By incorporating the Location Service API, applications were enhanced to deliver context-aware services such as location-based advertising and navigation. The real-time location tracking enabled applications to adapt their behavior based on the geographical position of the UE, offering a tailored user experience.

- Application Mobility Service (AMS): The AMS API enables MEC applications to manage the mobility of UEs across different network nodes, ensuring service continuity during handovers between cells or access points. By leveraging the AMS API, applications were configured to minimize latency and prevent service interruptions during transitions. This ensured seamless connectivity, particularly in scenarios involving high mobility, such as vehicular networks.
- WLAN Access Information Service (WAIS): The WAIS API provides details about available Wi-Fi networks, including signal strength, access point characteristics, and network load. This API was integrated into MEC applications to optimize performance in environments where both 4G/5G and Wi-Fi connectivity were available. Applications utilized the WAIS API to dynamically switch between networks, ensuring optimal connectivity and resource utilization.
- V2X Information Service API: The V2X Information Service API facilitates interaction with vehicles in Vehicle-to-Everything (V2X) environments by providing data such as vehicle status, position, and velocity. This API was employed to enable real-time applications for traffic management and autonomous driving. The integration of the V2X API ensured high reliability and low latency, critical for safety-critical applications in connected and autonomous vehicle systems.

Finally, experiment of multiple APIs by ETSI MEC sandbox has been successfully conducted.

III. Results and Analysis

A. Network Simulation:

The successful connection between the simulated UE and gNodeB in UERANSIM was validated using packet capture tools like Wireshark. Protocol exchanges (e.g., NGAP and SCTP) were analyzed, demonstrating efficient communication and error-free registration when executed with proper configurations.

B. MEC API Performance:

MEC011 API enabled effective resource allocation, achieving low latency in real-time data processing applications. **RNIS API** optimized application performance by dynamically adjusting transmission parameters based on radio conditions. **AMS API** ensured uninterrupted service continuity during UE handovers, even in heterogeneous network environments.

IV. Conclusion

This project demonstrated the feasibility and effectiveness of using open-source tools like Open5GS and UERANSIM for simulating 5G networks. The integration of MEC capabilities, tested in the ETSI MEC Sandbox, highlights the potential of edge computing in enhancing 5G applications' performance. Future work may involve scaling the network simulations to include more complex scenarios and integrating AI-based optimization techniques for dynamic resource management.

V. Project Contribution

In section of Open5Gs and UERANSIM, setup of Open5Gs was carried out by Dipean Dasgupta and also had a major contribution in final report preparation. Standalone and non standalone architecture was also setup. To create subscribers and get their tracking user interface was made open. UERANSIM setup gnodeB connection was performed by Archit Verma and he also had important contribution to the report through content generation.
In section of ETSI MEC API testing, Shobhit Gupta and Rahul Rathore had significantly contributed in completing the experiment. All the APIS were tested by them in ETSi MEC sandbox. APIs were bandwidth API, Location API, RNI. Subscriber interface for interaction with these APIS were also formed. On the application Side from swagger, both the server and client application was taken and configured to interact between them. Apart from this, Shobhit Gupta also setup Free5Gc with golang packages to try the simulation of UERANSIM and used the NRF and SMF functions. Both also had contributed to report content with their experiment results.

VI. Acknowledgment

We are extremely grateful to Dr. Bhupendra Kumar, course instructor of 5G Communication and Network for guiding us throughout the project.

VII. Project Links

The Project report is originally saved in: Report Latex
A Short video providing overview of the project: Project Video
Project Gihub repository: Project Github

References

[1] ETSI, *MEC in 5G networks*, ETSI White Paper 28, Sophia Antipolis CEDEX, France. [Online]. Available: https://www.etsi.org.
[2] ETSI, *Developing Software for Multi-Access Edge Computing*, ETSI White Paper 20, Sophia Antipolis CEDEX, France.
[3] ETSI, *Mobile Edge Computing (MEC); Mobile Edge Platform Application Enablement*, ETSI GS MEC 011, V1.1.1, 2017.
[4] ETSI, *Multi-access Edge Computing; Framework and Reference Architecture*, ETSI GS MEC 003, V2.1.1, 2019.
[5] ETSI, *Multi-access Edge Computing; UE Application Interface*, ETSI GS MEC 016, V2.1.1, 2019.
[6] ETSI, *Swagger Editor for MEC App Support API*. [Online]. Available: https://forge.etsi.org/gitlab/mec/gs011-app-enablement-api/raw/master/MecAppSupportApi.yaml.