# Requirement Analysis

## General Objective

The main objective of the work is to implement the ongoing project of a 3GPP complaint 5G Core network by OpenAirInterface Software Alliance (OSA) on a network emulation software called CORE emulator and test it with a simulated access network. The emulated 5G network shall provide the working 5G environment using multiple core networks and access networks. Simulated user equipment (UE) should be able to communicate with 5GC via access network and other UEs as well.

Below is the general structure of the system which needs to be achieved. As can be seen in Fig., the complete network shall be running in a Core emulator, which will be further running in an Ubuntu virtual machine. The UEs will be connected to the Access network, which will be further connected to the 5G core network and then the Data network or any other UE. The UE will communicate with each other or the data network leveraging the core network and the access network.

Chart, treemap chart

Description automatically generated

Figure 3.1General Structure of 5G envrionment needs to be achieved

## Clarifying the requirements

The main functionality required is to produce a network emulation of a standalone 5G environment in CORE. This 5G environment shall provide the service of 5G Connectivity within the UEs or the with the data network.

The Access network and core network of 5G environment shall be created in individual server hosts provided by CORE emulator. These server hosts nodes will have their own process environment and private network stack. The filesystem will be shared by both nodes. For a working standalone 5G network. These two hosts shall be connected with each other and able to communicate. Below is the architectural overview of the required system in CORE.

Diagram

Description automatically generated

Figure 3.2Architecture overview in Core Emulator

### OAI 5G Core network

The OAI 5G Core network requires its network functions to be deployed on Docker containers and integrated to provide the core network services. As these network functions are integrated using the docker network bridge of one host system, the CORE emulator provides this required server host in its environment. Using its docker test service, CORE lets the containerized network functions run in its isolated server host. These network functions shall communicate with each other using the docker network bridge to provide their dedicated services. Figure 4.3 describes where and in what order these network functions shall be started in the 5GC server host.

Diagram

Description automatically generated

Figure 3.3 5GC Network Functions running in a host on CORE Emulator

The deployment order of these network functions is crucial for a successful running 5G core network. Hence it should strictly follow with the help of docker tools and python scripting. The numbering depicts in the above figure that the Mysql container should be the first to start. After it, the NRF container shall start, followed by AMF, SMF, and UPF containers. In the last, the DN container should start.

### UERANSIM 5G Access Network

UERANSIM 5G Access network requires a host system with few Linux packages in order to run its simulation environment. A similar to 5GC server host system to 5GC shall be utilized to run a simulated 5G access network and UEs in the CORE Emulator. Leveraging the Linux tools, simulation shall run in this isolated server host. This host should be able to communicate with the 5GC host and thus the simulated UEs and gNodeBs with containerized network functions running in it. The below figure describes the simulation that shall be running inside the 5G AN host. There shall be several UEs and gNodeB to test the 5G services provided by the 5GC host.

A picture containing text, screenshot, jack

Description automatically generated

Figure 3.4Simulation of UEs and gNodeBs inside 5G AN host on Core Emulator

These UEs and gNodeBs should utilized the IP routing and forwarding in order to reach CNFs in 5GC host.

## Target State

For the implementation of the prototype, the final state shown in Figure 4.5 is expected. A 5G core with 5G enabled UEs, and RAN is provided in the developed network emulation. UE will establish a PDU session with the data network with the help of RAN and the core network. The simulated UE will first connect to the AMF container running in the 5GC host using the direct connectivity between 5G UERAN and 5GC. It’ll send a PDU session request to AMF. Now AMF will communicate with NRF to get the IP address of the SMF container. Using the IP address, it’ll speak with the SMF and will forward the PDU session request. SMF will need the subscription data for authentication/authorization of UE for an end-to-end PDU session between UE and Data network. It shall communicate with UDM(MySQL container) to get this data, where this data should be stored beforehand.

Now the control will turn to UPF to get the required information to create a GTP tunnel between UE and DN. SMF shall communicate with UPF to get this information. Now SMF shall create a tunnel endpoint and send the tunnel details and PDU session details to AMF. Now the AMF forwards these parameters to RAN running in the 5G UERAN host. Upon receiving the information, NG RAN(gNB) setups the GTP Tunnel based on the information received from AMF and configure the Tunnel Endpoint (or Bearer). And then, gNB forwards the message to UE for the setting of the PDU session. Now the UE communicates end-to-end with the DN using the established PDU session.

Diagram

Description automatically generated

Figure 3.5Target System in an overview

## Software and Hardware Requirements

### Required Softwares

#### ****CORE Emulator****

To emulate the whole network of 5G environment, latest version of CORE emulator shall be used. Release 27, CORE 7.5.2 is available on github which requires to be cloned on the host VM and install. The GUI provided by the CORE is not supporting by default for the docker services in it. Hence to deploy the 5GC network functions, on docker containers, Docker test service shall be configured on the emulator. The GUI provides drag and drop function to put network functions like server hosts and their linking in the emulation which is required to deploy all of the 5G components.

#### ****Docker Containers****

The 5G core network is designed to use cloud-native architectures to boost scale and versatility, consisting of net-work functions operating on container-based platforms called containerized network functions (CNFs). This allows the deployment of core network functions in any private or public cloud location. CNFs are decomposed into microservices as much as possible, which are the services working together as a distributed system (Intel, 2020). Docker Engine and Docker-compose host will be required to run network functions 5G Core. Due to the compatibility with CORE Emulator, a specific version of Docker engine must be used. Docker version 17.12.1-ce, build 7390fc6 is supported by CORE emulator GUI and hence the same shall be used. To run the CNFs in specific order, docker-compose will utilized and it will be run from a python script.

#### ****Ubuntu****

Having various features such as scalability, fast, simple, secure and minimal operating system, Ubuntu is default choice for many containers related operations. Ubuntu 18.04 (bionic server) distribution is required for building the containers in 5GC. Also, to host these containers on CORE emulated network nodes along, Ubuntu 18.04.4 LTS OS is required. This virtual machine is to be created on a Virtual Box which is explained in the next section.

#### ****Virtual Box****

Oracle VM VirtualBox is a cross-platform virtualization application that enables users to expand their existing machines to run several operating systems at once. It is intended for IT professionals and developers that operate on Microsoft Windows, Mac OS X, Linux, and Oracle Solaris systems. Oracle VM VirtualBox is ideal for testing, designing, demonstrating, and implementing solutions across various platforms on a single computer (Oracle, 2019). A virtual machine with enough computing resources as mentioned below shall be created.

* 2GB RAM
* 2 CPU
* 20 GB VDI
* Vboxnet Host-Only network adapter

#### ****Wireshark & tshark****

To analyse the traffic and 5G process, latest version of Wireshark and tshark must be installed on the host system.

Wireshark is the world's most popular network traffic analyser, and it's a must-have for any security or systems administrator. This free software allows one to analyse network traffic in real-time and is often the most effective tool for troubleshooting network issues. tshark is the terminal-based version of Wireshark which is often in container environments (Porup, 2018). It’ll be used the analyse the traffic inside 5G CNFs.

#### ****Linux Packages****

Some linux packages shall be installed on the VM to achieve the objective.

* Git
* Docker Engine 19.03.6,
* Docker-compose 1.27.4
* Python 3.6.9
* CMake 3.17 or later
* gcc 9.0.0 or later

### Required Hardware

#### ****Host PC****

As the OAI 5G core is designed as open source it can be deployed normal computing hardware with enough computing capacity. Intel architecture-based x86 PC is required to host virtual machines for the CORE emulation to work. The PC should have at least 4 cores, >3GHz for smooth and seamless deployment.

## Use Cases for the Prototype

To demonstrate the functionality, an emulation of the standalone 5G network has to be implemented.

The network emulation provides the implementation of the 5G Core network by OAI, which renders 5G services.

* Using the cli of 5GC host in CORE emulator, a python script starts 5G Core.
* The OAI 5GC has containerized network functions deployed on docker containers which starts in a specified order. These CNFs are integrated and provide their dedicated services.
* CNFs are connected using a docker network bridge to communicate with each other, hence getting the required services.
* The Data network to test the high bitrate for the 5G Core service resides inside the 5GC host and running a dedicated docker container.
* The python script deploys the CNFs and make them up and running.
* A heartbeat functionality of docker services will continuously check health of each CNF.
* Two types of deployment can be performed using the provided python scripts.

1. Minimalist Deployment (This deploys a functional 5G core network with minimal required network functions). A below strict order of deployment must be followed

mysql --> oai-nrf --> oai-amf --> oai-smf --> oai-upf

1. Basic Deployement (This deploys a a funcational 5G core network with all the network functions). A below strict order of deployment must be followed

mysql --> oai-nrf --> oai-udr --> oai-udm --> oai-ausf --> oai-amf --> oai-smf --> oai-upf

The network emulation also provides the implementation of simulated Access Network and UEs using the UERANSIM simulator running inside the 5G UERAN host server. The 5GC is tested using this simulation of UEs and gNodeBs.

* Using the cli of 5G UERAN host, a user starts the gNB first with a configured .yaml file
* The gNB will communicate with AMF container in 5GC host and establish an SCTP connection between them.
* After it, the gNB will setup an NG interface between gNB and AMF.
* Now, a user can start the UE using another configured yaml file.
* UE will connect to the gNB and AMF and will send initial registration request to AMF.
* The authentication will happen for the UE using the subscriber stored data in UDM inside 5GC.
* Successful authentication will proceed to the PDU session establishment process with Data network.
* After an established PDU session between a UE and Data network, a high bitrate transfer can be analysed using iperf tool.

The live running emulation of alone OAI 5GC can also connected to a real physical gNB and COTS UE. It provides easy testing of 5GC by taking advantages of tools provided by the Linux OS.

# Implementation

# Summary and Perspectives

# Abbreviations

|  |  |
| --- | --- |
| **3** |  |
| 3GPP | Third Genration Partner Project |
| **5** |  |
| 5GS  5G CN  5G AN  **A**  AMF  AUSF | 5G System  5G Core Network  5G Access Network  Access and Mobility Management Function  Authentication Server Function |
| **C**  COTS  CUPS  CI/CD  CNF  **D**  DN | Commerical Off-The-Shelf  Control and User Plane Function  Continous Intergration/ Continous Development  Containerzied Network Function  Data Network |
| **E**  EPC | Evolved Packet Core |
|  |  |
| **G** |  |
| GSM | Global System for Mobile communications |
| GPRS | General Packet Radio Service |
| gNB | G Node B |
|  |  |
| **I**  ISDN  IMS  **L**  LTE  **N**  NF  NFV  NEF  NRF  NSSF  **O**  OSA  OAI  **P**  PDU  PFCP  PCF  PDN  PGW-C | Integrated Services Digital Network  IP Multimedai Subsystem  Long Term Evolution  Network Function  Network Function Virtualization  Network Exposure Function  Network Repository Function  Network Slice Selection Function  OpenAirInterface Software Alliance  Open Air Interface  Protocol Data Unit  Packet Forwardubg Control Protocol  Policy Control Function  Packet Data Network  PDN Gateway Control |
| **S** |  |
| SDN  SBA  SMF  SGW-C  SPGW-U  **U**  UE  UPF  UDM  **V**  VM  VDI | Software-Defined Networking  Service Based Architecture  Session Management Function  Service Gateway Control  User Plane of Packet Data Network Gateway  User Equipment  User Plane Function  Unified Data Management  Virtual Machine  Virtual Desktop Infrastructure |
|  |  |

# References

# Irving, P. J. & Ochang, P. A., 2016. Evolutionary Analysis of GSM, UMTS and LTE Mobile Network Architectures. World Scientific News, Issue 54, pp. 27-39.

# 3GPP TS 23.50, 2017-12. V15.0.0. System Architecture for the 5G System (Stage 2). s.l.:s.n.

# 3GPP TS 23.002, 2014. V12.5.0 Digital cellular telecommunications system (Phase 2+). s.l.:s.n.

# 3GPP TS 23.228, 2013. V11.10.0 IP Multimedia Subsystem (IMS) Stage 2. s.l.:s.n.

# Trick, P. U., 2020. Mobile Computing, s.l.: s.n.

# Accenture, 2007. IMS Architecture Overview, Napoli: s.n.

# Salchow, K. (., 2007. Introduction to the IP Multimedia Subsystem (IMS): IMS Basic Concepts and Terminology. F5 Networks Inc, August.

# Firmin, F., 2020. The Evolved Packet Core, s.l.: 3GPP MCC.

# Chourasia, S. & Sivalingam, K. M., 2015. SDN Based Evolved Packet Core Architecture For Efficient User Mobility Support. London, UK, IEEE Conference on Network Softwarization (NetSoft).

# Rizk, C. N., M., T. M. R. & Mokhtar, B. M., 2020. Software Defined Network-Based Management for Enhanced 5G Network Services. IEEE Access, Band 8, pp. 53997-54008.

# Singh, S. & Jha, R. K., 2016. A Survey on Software Defined Networking: Architecture for Next Generation Network. Journal of Network and Systems Management, Band 25, p. 321–374.

# Tipantuña, C. & Yanchapaxi, P., 2017. Network functions virtualization: An overview and open-source projects. Salinas, Ecuador, IEEE Second Ecuador Technical Chapters Meeting (ETCM).

# Brown, G., 2017. Service-Based Architecture for 5G Core Networks. White Paper.

# 3GPP TR 23.501, 2017. V17.0.0 Study on User Plane Function (UPF) enhancement for control and 5G Service Based Architecture (SBA). s.l.:s.n.

# Mayer, G., 2017. 3GPP 5G CoreNetwork. s.l.:Webinar.

# 3GPP TS 23.501, 2021. V15.12.0 System architecture for the 5G System (5GS). s.l.:s.n.

# 3GPP TS 29.509, 2018. V15.1.0 5G System Authentication Server Services Stage 3. s.l.:s.n.

# Docker, 2021. What is a container. [Online] Available at: https://www.docker.com/resources/what-container [Zugriff am 2021].

# Intel, 2020. Why Use Containers and Cloud-Native Functions Anyway. [Online] Available at: <https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/containers-and-cloud-native-functions-white-paper.pdf> [Zugriff am 25 03 2021].

# RedHat, 2021. OpenShift Container Platform 4.4 Architecture. [Online] Available at: <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.4/pdf/architecture/OpenShift_Container_Platform-4.4-Architecture-en-US.pdf> [Zugriff am 26 03 2021].

# IONOS, 2020. GitLab Tutorial. [Online] Available at: https://www.ionos.com/digitalguide/websites/web-development/gitlab-tutorial/ [Zugriff am 26 03 2021].

# Selhorn, S., 2021. GitLab Docs. [Online] Available at: https://docs.gitlab.com/ee/README.html [Zugriff am 26 03 2021].

# 5G Americas, 2017. 5G Services and Use Cases. s.l.:Whitepaper.

# Digi International, 2019. 5G Applications and Use Cases. [Online] Available at: <https://www.digi.com/blog/post/5g-applications-and-use-cases> [Zugriff am 29 March 2021].

# 3GPP TS 29.244, 2017. Interface between the Control plane Plane and the User Plane of EPC Nodes. V14.0.0 Release 14: s.n.

# Stanley, S., 2016. COTS Platforms for the New IP Era: A Competitive Analysis, s.l.: Heavy Reading.

# Balazs Bertenyi1, R. B., Masini, G., Sirotkin, S. & Gao, Y., 2018. (NG-RAN), NG Radio Access Network, s.l.: River Publishers.

# Speicher, S., Sirotkin, S., Palat, S. & Davydov, A., 2020. 5G System Overview. In: S. Sirotkin, Hrsg. 5G Radio Access Network Architecture: The Dark Side of 5G . s.l.:Wiley-IEEE Press, pp. 37-122.

# 3GPP TS 29.561, 2018. Interworking between 5G Network and external Data Networks. v15.0.0 : Release 15.

# Tabassum, M. & Mathew, K., 2014. Software Evolution Analysis of Linux (Ubuntu) OS. International Conference on Computational Science and Technology (ICCST), pp. 1-7.

# Oracle, 2019. Oracle VM VirtualBox Overview. An Oracle White Paper. [Online] Available at: <https://www.oracle.com/us/technologies/virtualization/oracle-vm-virtualbox-overview-2981353.pdf> [Zugriff am 27 04 2021].

# Porup, J., 2018. What is Wireshark? What this essential troubleshooting tool does and how to use it. [Online] Available at: <https://www.csoonline.com/article/3305805/what-is-wireshark-what-this-essential-troubleshooting-tool-does-and-how-to-use-it.html> [Zugriff am 27 04 2021].