Interacting with Core Network in the Free5GC Simulator

Yurun Song

Abstract

Explores the inner workings of the 5G network by interacting with the control program of the Free5GC simulator is the best way we learn from the wireless network and implement the Software Defined Networking (SDN) concepts behind 5G technology. In the project, we simulates a local 5G network using virtual machines and establish the communication channel for the different User Equipments (UEs). We capture their routing packets using network tools such as Wireshark. We then analyze the data collected to evaluate the effectiveness of the network performance. We also configure different network elements and experiment with various network scenarios to observe the effects on the network's performance.

1 Introduction

The world of mobile technology has seen tremendous advancements in the past decade, and the emergence of 5G technology has revolutionized the way we communicate and use mobile devices. As the latest mobile network technology, it provides ultra-fast internet speeds, improved reliability, and low latency.

In this project, I will explore the inner workings of the 5G network by interacting with a control program of a 5G simulator called free5gc[1]. This open-source 5G core network implementation provides a complete solution for the various core network functions of the 5G network. By simulating a local 5G network using virtual machines, we will have the opportunity to delve into the different Network Functions (NFs) of the 5G core network and their functionality.

The primary goal of this project is to showcase the SDN concepts behind the 5G technology and explore the different NFs and their functionality in the 5G core network. Additionally, I will achieve data communication between simulated UEs , which will allow us to test the routing path of the network under different conditions. Moreover, this project aims to explore the potential use cases and benefits of 5G technology in different high level in the future work. Such as building the custom network function to provide real-time remote monitoring, improve efficiency, and enhance overall user experience when interacting with entire 5G network using the natural language.

2 Motivation

To begin with, 5G represents the newest advancement in wireless technology, boasting faster speeds, reduced latency, and enhanced reliability compared to previous iterations. As a result, it has become increasingly attractive for applications requiring high bandwidth and low latency, such as remote surgery, autonomous vehicles, and virtual reality.

Furthermore, simulating a 5G network with a core network enables us to assess and evaluate the performance of custom network functions in a controlled environment, which may be challenging to find a real deployed 5G core network. This also allows us to add extra application functions for future experimentation.

Free5GC is one of the most practical 5G simulators with a core network implementation available. Other simulators, like simu5G[2] and OSM[3], either lack a control plane or are too complicated for research-level implementation. Free5GC is cost-effective and highly flexible, allowing it to be integrated with physical hardware or infrastructure such as small 5G gNBs from ALPHA Network Inc. and 5G UEs from APAL MiFi devices.

Lastly, utilizing a 5G simulator with a core network can provide familiarity with SDN and other technologies used in the Advanced Network course. By understanding the inner workings of the 5G control plane, one can gain insights into how different network elements interact with each other, how signaling messages are exchanged, and how network resources are allocated and managed.

3 Related Work

The journey from 1G to 5G has been a long and exciting one, marked by continuous improvements and new features. The 1G mobile networks were introduced in the 1980s and were characterized by sound quality and limited coverage. With the development, 2G networks were introduced in the 1990s. It was digital and enabled the use of text messaging and basic internet connectivity. The 3G networks introduced in the early 2000s improved mobile internet speeds, enabled multimedia messaging, and faster internet browsing, while the 4G networks introduced around 2008 offered even faster internet speeds, lower latency, and supported high-quality video streaming, online gaming and other data-intensive applications. [3]

Now, we have the 5G mobile networks, which represent a significant leap forward in terms of speed, reliability, and functionality. 5G networks promise ultra-low latency, massive device connectivity, and faster data transfer speeds. They enable real-time applications, such as augmented reality and

[1] https://www.free5gc.org/

[2] http://simu5g.org/

[3] https://osm.etsi.org/

virtual reality, and support new use cases, such as smart cities, autonomous vehicles, and the Internet of Things (IoT). The 5G network is generally can be visualized as two components from the way of SDN, control plane and data plane. The control plane is called Core Network, which is usually located on the cloud servers. The data plane consists of Radio Access Network (RAN) and plenty of User Plane Functions (UPFs), which are physically deployed across different areas. In the core network, there are several Network Functions (NFs) responsible for managing and controlling the traffic and services offered by the network. These NFs include the AMF (Access and Mobility Management Function), which manages user authentication and mobility between different network areas, and the SMF (Session Management Function), which is responsible for setting up and managing user sessions. Other functions such as PCF (Policy Control Function), UDM (Unified Data Management), UDR (Unified Data Repository), and NRF (Network Repository Function) are responsible for maintaining the core network functionality in the control plane.
Based on the 3GPP [1] requirements, the carriers are responsible to create these fundamental NFs in the core network. However, the combination of SDN with Network Functions Virtualization (NFV) also allows for the creation of auxiliary virtualized network functions, enabling more efficient deployment and management of 5G network services.
The control plane and data plane work together to provide seamless connectivity and communication between the UE connected to the 5G network. By utilizing the principles of SDN, the 5G network can adapt to varying traffic and service demands, providing a more flexible, dynamic, and programmable approach to network management.
Simulators such as Simu5G, OMNeT++ [4], and 5G Matlab & Simulink have gained popularity for 5G simulations. However, Free5GC stands out as a free-to-use implementation specifically designed for simulating the entire 5G network. Unlike other simulators that can only simulate either the control panel or the data panel, Free5GC offers the capability to simulate both aspects. Additionally, some industry-level simulators are overly complex for implementation and research purposes. As a result, we have chosen Free5GC as our primary research tool.
There are many other similar work explore and extends the 5G core network with free5gc, such as [8] develops a Hybrid SDN Mobile Core Network for 4G and Non-Standalone 5G, [6] evaluates NF visualization-enabled Network Slicing for 5G Core, [7] implements a P4 based user plane function for free5gc, etc.

4 Architecture

In the figure 1, we can see the architecture of the 5G network can be split into data and control. The control panel is responsible for managing the network resources and making decisions on how to route data through the network, while the data plane is responsible for forwarding the actual data packets. This separation of control and data plane is a key feature of SDN, which allows for more efficient and flexible management of network resources. In the 5G architecture, this separation enables network operators to dynamically allocate resources to meet the needs of different applications and services, and to optimize network performance and efficiency.

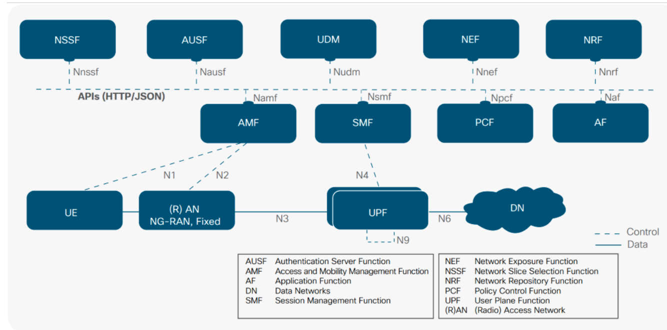

Figure 1: This is the 5G Network architectures

4.1 5G Core Network Functions

The 5G Core comprises many different Network Functions and other customised application functions, and in this project, we primarily interact with four key network functions: AMF, SMF, UDM, and UPF. AMF is in charge of user access, mobility, and connection management within the network. It is vital for device registration, authentication, and authorization processes. Additionally, AMF manages mobility aspects, such as tracking area updates and handovers, ensuring uninterrupted connectivity for users as they traverse different coverage zones. SMF handles the management and maintenance of data sessions between users and the network. It allocates IP addresses, establishes and modifies data paths, and enforces Quality of Service (QoS) policies to optimize network resource usage. Furthermore, SMF assists in managing charging and billing data, enabling operators to bill users based on their consumption. UDM manages and stores user-related data, including subscription information, authentication credentials, and access policies. Acting as a central repository for user data, it supplies other network functions with the necessary information to perform their duties. UDM also supports network slicing, which allows operators to tailor networks for specific use cases or customer segments. UPF is responsible for managing user data traffic within the network. It directs data packets between users and the internet while applying QoS policies and charging rules. UPF also facilitates traffic offloading, enabling operators to direct specific traffic types straight to the internet without passing through the core network, reducing latency and enhancing overall network performance.
These network functions communicate through various net-

work interfaces. For instance, the N11 interface connects AMF and SMF and is responsible for conveying NAS messages and notifying the protocol data unit (PDU) session and UE connection status. Each interface serves a unique purpose, facilitating the exchange of information and data between different network functions. The APIs employed to create these interfaces guarantee efficient and secure communication between network functions, providing users with fast and dependable 5G connectivity.

4.2 Packet Routing

First, the UE establishes a connection with the nearest base station (BS) or access point (AP) using radio waves. This connection is known as the air interface, and it operates on a specific frequency band assigned to the operator. When a UE sends a packet to a 5G AN (gNB), it initiates a PDU session by sending a NAS (Non-Access Stratum) PDU session establishment request to the AN and AMF . The SMF creates a PDU session for the AMF and responds with an accept message. The AMF then signals the base station by transferring this message using the Next Generation Application Protocol (NGAP).
IP address allocation is dependent upon the type of PDU session, and the SMF selects an appropriate UPF during the setup of the PDU session. The UPF establishes a GTP-u (GPRS Tunnelling Protocol for user plane) tunnel with the base station, and the IP header specifies the routing packet between the base station and UPF. The UDP (User Datagram Protocol) header is used for the port number for GTP-u, and the GTP-u header specifies a Tunnel Endpoint identifier (TEID) that links the user plane packet to a specific PDU session. UPF allocates TEID for uplink data transfer, and for the downlink data, the TEID is signaled from the UPF to the base station via the SMF and AMF.
The SMF selects an appropriate Policy Control Function (PCF) during the setup of a PDU session and is responsible for generating Service Data Flow (SDF) templates that map each downlink packet onto specific QoS (Quality of Service) flows, such as source/dest IP & port. The UPF removes the GTP-u header, forwards the packet to the corresponding Data Network, and keeps a recording of a mapping from GTP-u, PDU session, and Data Network. The UPF also tracks the volume of data transferred by each PDU session and reports it to the SMF for billing purposes. The data packet is then encapsulated in a PDU that contains the necessary header and trailer information, such as the source and destination addresses, the packet type, and the sequence number. The PDU is then sent to the RAN, which converts it into a series of radio waves that are transmitted over the air interface.
At the receiving end, the destination UE receives the radio waves and converts them back into a digital data packet. The UE then checks the packet header and verifies that the packet is intended for it. If the packet is valid, the UE sends an acknowledgement back to the source UE, indicating that the packet has been received successfully.

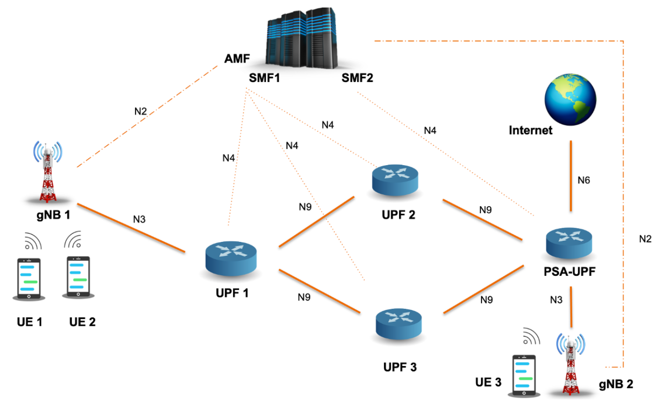

Figure 2: 5G Network Simulation Design

5 Design

In this section, we will present the design of our simulator, which is built to model and simulate the deployment and behavior of the UEs, RANs, and core network components in our system. The design steps are shown in the Figure 2. We are planning to deploy three UE devices and two RAN in our system. Specifically, UE1 and UE2 will be connected to RAN1, while UE3 will be linked to RAN2. As a result, UE1 and UE2 will have overlapping paths, and UE3 will able to communicate with both UE1 and UE2.
Regarding PDU sessions, we have prepared three sessions, all of which are linked to the internet. However, they will pass through different UPF entities. Session 1 and session 2 will travel from gNB1 to PSA-UPF (PDU Session Anchor UPF), each using UPF2 and UPF3, respectively. UE1 and UE2 will utilize these two sessions to communicate with Internet individually. Consequently, during the evaluation when we examine the captured packets by UPF2 and UPF3, only the packets from UE1 can pass through UPF2 and UE2 can pass through UPF2. The presence of the PSA-UPF, which serves as a PDU Session Anchor UPF, is significant. It acts as a central point for managing the PDU sessions and enables direct connectivity between UE3 and the internet without the involvement of additional UPFs. UE3 directly communicates with PSA-UPF using RAN2 and session3, demonstrating that PSA-UPF can perform the same role as other UPFs, albeit with a direct Data Network connection.
Additionally, In our network design, it is important to clearly define the network interfaces between different components to ensure proper communication and data transfer. Here are the specific network interfaces that is outside of the core network:
RAN and Core Network (N2): Connects RANs to the Core Network, enabling control and signaling communication. UPF

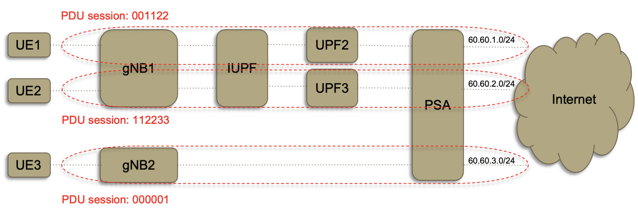

Figure 3: The connectivity of PDU Session in the experiments

with Core Network (N4): Establishes connection between UPFs and Core Network functions for session management and policy enforcement. RAN and UPF (N3): Links RANs with UPFs for user data packet forwarding and routing. UPFs and UPFs (N9): Interconnects UPFs for user data packet exchange and control information sharing. UPFs and Data Networks (N6): Connects UPFs with external Data Networks for data transfer.

This simple setup effectively illustrates successful data transmission between UEs and the internet. Furthermore, it is important to note that in the initial state, before the deployment and configuration of the UEs and RANs, none of the devices are capable of communicating with the data network. They are essentially disconnected and unable to transmit or receive any data. This setup represents a starting point where the devices are isolated from the network and need to be properly configured to establish connections and enable data transmission. Upon deployment, each UE is assigned to their RAN based on the network architecture and starts the connection. In conclusion, this deployment scenario and configuration showcase the successful data transmission between UEs and the internet.

6 Experiments

In order to validate and demonstrate the design described in the previous section, we conducted a series of experiments using our simulator. The experiments were designed to assess the performance and functionality of the deployed UEs, RANs, and network components in various scenarios.

Step 1: Initial Network Setup

In this experiment, we utilized seven virtual machines (VMs) to deploy the entire system. The VMs were allocated specific roles to represent different network components. Here is an overview of the VM allocation:

Core Network VM: This VM served as the core network, where all the NFs were deployed. It was directly under the host-only network, ensuring a secure and isolated environment for network operations.

UPF VMs: Four VMs were dedicated to the UPF entities. Each UPF was deployed on a separate VM, providing isolation and scalability. These four VMs played a crucial role in handling the packet forwarding and routing functions of the system.

RAN and UE Group VMs: Two VMs were allocated for the RAN and UE groups. In these VMs, it simulated the behavior and interaction of the RANs and their corresponding groups of UE. The RANs were responsible for connecting the UEs to the core network and managing their communication.

we examined the network's initial state where all devices were unable to communicate with the data network. We verified that the UEs and RANs were properly configured and initialized, ensuring they were ready for data transmission.

Step 2: RAN and UPF Connection

To establish the connection between the RAN and UPF components, it is necessary to configure the network adapters on each VM beforehand. Setting up the network adapters involves specifying the network settings and configurations required for communication to be established. In this process, each VM representing a RAN or UPF component needs to configure its network adapter by providing details such as IP addresses, subnet masks, gateway addresses, and other settings. These settings ensure that the VMs can communicate with each other and establish the necessary network connections.

To enable data communication with the corresponding destinations, gateways were set up in the PSA-UPF. Each PDU session was associated with a specific DNN list, specifying the gateway for communication with its intended destination. This ensured that data traffic from the UEs was directed through the appropriate gateway towards the desired destination, enhancing the efficiency and effectiveness of data transmission.

To ensure clear and effective connectivity, it is important to specify the configuration of RANs and UPFs before connection, including their virtual connections to neighboring components through different network interfaces.

Step 3: PDU Session Setup

In this experiment, our focus was on configuring the PDU sessions for data transmission within the network. The are seveal steps were followed to set up these sessions:

Configuring PDU Sessions: We configured three PDU sessions in the AMF and SMF configuration files. These sessions were defined with specific parameters and settings, along with policy that UPF allows.

Session Registration in Database: The configured PDU sessions were registered in the mongoDB, allowing for efficient management and retrieval of session information. This ensured that the sessions were properly stored and associated with their corresponding UPFs.

PDU Session Association: When the UPFs established a connection with the Core Network, the Core Network examined their supported Data Network list and verified the compatibility with the configured PDU sessions. The Core

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	60.60.2.1	8.8.8.8	GTP <ICMP>	144	Echo (ping) request id=0x0003, seq=470/54785, ttl=64 (reply in 2)
2	0.002976	8.8.8.8	60.60.2.1	GTP <ICMP>	136	Echo (ping) reply id=0x0003, seq=470/54785, ttl=61 (request in 1)
3	0.206989	60.60.1.1	8.8.8.8	GTP <ICMP>	144	Echo (ping) request id=0x0004, seq=448/49153, ttl=64 (reply in 4)
4	0.209817	8.8.8.8	60.60.1.1	GTP <ICMP>	144	Echo (ping) reply id=0x0004, seq=448/49153, ttl=61 (request in 3)
5	0.313820	60.60.3.1	8.8.8.8	GTP <ICMP>	144	Echo (ping) request id=0x0006, seq=701/48386, ttl=64 (reply in 6)
6	0.316732	8.8.8.8	60.60.3.1	GTP <ICMP>	144	Echo (ping) reply id=0x0006, seq=701/48386, ttl=61 (request in 5)
7	1.001205	60.60.2.1	8.8.8.8	GTP <ICMP>	144	Echo (ping) request id=0x0003, seq=471/55041, ttl=64 (reply in 8)
8	1.004132	8.8.8.8	60.60.2.1	GTP <ICMP>	136	Echo (ping) reply id=0x0003, seq=471/55041, ttl=61 (request in 7)
9	1.208058	60.60.1.1	8.8.8.8	GTP <ICMP>	144	Echo (ping) request id=0x0004, seq=449/49409, ttl=64 (reply in 10)
10	1.210818	8.8.8.8	60.60.1.1	GTP <ICMP>	144	Echo (ping) reply id=0x0004, seq=449/49409, ttl=61 (request in 9)
11	1.314843	60.60.3.1	8.8.8.8	GTP <ICMP>	144	Echo (ping) request id=0x0006, seq=702/48642, ttl=64 (reply in 12)
12	1.317713	8.8.8.8	60.60.3.1	GTP <ICMP>	144	Echo (ping) reply id=0x0006, seq=702/48642, ttl=61 (request in 11)
13	1.539901	PcsCompu_2d:c7:65		ARP	44	Who has 172.16.3.2? Tell 172.16.3.103
14	1.540145	PcsCompu_8c:4f:d8		ARP	62	172.16.3.2 is at 08:00:27:8c:4f:d8
15	2.003165	60.60.2.1	8.8.8.8	GTP <ICMP>	144	Echo (ping) request id=0x0003, seq=472/55297, ttl=64 (reply in 16)
16	2.006070	8.8.8.8	60.60.2.1	GTP <ICMP>	136	Echo (ping) reply id=0x0003, seq=472/55297, ttl=61 (request in 15)
17	2.211263	60.60.1.1	8.8.8.8	GTP <ICMP>	144	Echo (ping) request id=0x0004, seq=450/49665, ttl=64 (reply in 18)
18	2.214028	8.8.8.8	60.60.1.1	GTP <ICMP>	144	Echo (ping) reply id=0x0004, seq=450/49665, ttl=61 (request in 17)
19	2.315859	60.60.3.1	8.8.8.8	GTP <ICMP>	144	Echo (ping) request id=0x0006, seq=703/48898, ttl=64 (reply in 20)
20	2.318738	8.8.8.8	60.60.3.1	GTP <ICMP>	144	Echo (ping) reply id=0x0006, seq=703/48898, ttl=61 (request in 19)
21	3.004577	60.60.2.1	8.8.8.8	GTP <ICMP>	144	Echo (ping) request id=0x0003, seq=473/55553, ttl=64 (reply in 22)
22	3.007479	8.8.8.8	60.60.2.1	GTP <ICMP>	136	Echo (ping) reply id=0x0003, seq=473/55553, ttl=61 (request in 21)

Figure 4: The ICMP packets sent from UE1, UE2 and UE3 to 8.8.8.8 are captured in the PSA-UPF.

Network then initiated the PDU session association among the UPFs based on the matching criteria. This association allowed the UPFs to establish connections with each other to build the real PDU sessions.

UE Connection and Session Request: When UEs connected to the Core Network via the RANs, they provided destination information and other Quality of Service details. The AMF registered this information and initiated a session request for the UE to connect to the desired destination.

Through these steps, we ensured that the PDU sessions were correctly established and bound to the appropriate UPF entities. Figure 3 shows the detail of the connectivity for each session.

Step 4: Data Transmission Evaluation

To assess the success of data transmission, we monitored the packet flow captured by the UPFs via wireshark. We specifically analyzed the packets passing through UPF2 and UPF3 to ensure that only UE1 and UE2 packets were transmitted through these UPFs, while UE3 utilized the direct communication path with PSA-UPF. This confirmed that the network was correctly routing and handling data according to the design specifications.

7 Evalution

Follows the experiment, we build the simulated 5G core network successfully, and conducted a series of experiments to validate its design and functionality. The experiments focused on assessing the performance and behavior of the deployed UEs, RANs, and network components in different scenarios.

To evaluate the success of data transmission, we monitored the packet flow captured by the UPFs using Wireshark. Specifically, we analyzed the packets passing through UPF2 and UPF3, confirming that only UE1 and UE2 packets were transmitted through these UPFs, while UE3 utilized the direct communication path with PSA-UPF. This validation demonstrated that the network was routing and handling data according to the design specifications.

Data transmission to Internet In our evaluation, we conducted packet capture and analysis for UE1, UE2, and UE3 to examine their routing paths from RAN1 and RAN2 to the gateways. Based on Figure 4, it is evident that the ICMP packet captured in the PSA-UPF employs the GPRS tunneling protocol, utilizing the gateway IP addresses 60.60.1.0, 60.60.2.0, and 60.60.3.0 for transmission and reception in both the forward and backward directions.

It indicates that data flows from UE1, UE2, and UE3 through RAN1 and RAN2, and is then routed to the gateways. The successful capture of this packet indicates that each path towards the outside data network has been successfully established.

PDU session on the correct UPF When comparing Figure 5 and Figure 6, we can observe that PDU Session 1 is connected to the PSA-UPF exclusively through UPF2. This indicates that UE1 is only allowed to utilize Session 0, and consequently, only packets from UE1 are captured at the location of UPF2. Similarly, in the case of UPF3, packets from UE2 through PDU Session 2 are exclusively captured.

This observation suggests that the packets are directed only along the designated paths assigned to them. Each UE's packets are routed through the specific UPF associated with their respective PDU sessions, ensuring that the packets are carried only through the intended paths.

No.	Time	Source	Destination	Protocol
1	0.000000	60.60.1.1	8.8.8.8	GTP <ICMP>
2	0.000020	60.60.1.1	8.8.8.8	GTP <ICMP>
3	0.003109	8.8.8.8	60.60.1.1	GTP <ICMP>
4	0.003115	8.8.8.8	60.60.1.1	GTP <ICMP>
5	1.001021	60.60.1.1	8.8.8.8	GTP <ICMP>
6	1.001045	60.60.1.1	8.8.8.8	GTP <ICMP>
7	1.004204	8.8.8.8	60.60.1.1	GTP <ICMP>
8	1.004210	8.8.8.8	60.60.1.1	GTP <ICMP>
9	2.002943	60.60.1.1	8.8.8.8	GTP <ICMP>
10	2.002966	60.60.1.1	8.8.8.8	GTP <ICMP>
11	2.006173	8.8.8.8	60.60.1.1	GTP <ICMP>
12	2.006180	8.8.8.8	60.60.1.1	GTP <ICMP>
13	3.004018	60.60.1.1	8.8.8.8	GTP <ICMP>
14	3.004037	60.60.1.1	8.8.8.8	GTP <ICMP>
15	3.007201	8.8.8.8	60.60.1.1	GTP <ICMP>
16	3.007211	8.8.8.8	60.60.1.1	GTP <ICMP>
17	4.004668	60.60.1.1	8.8.8.8	GTP <ICMP>
18	4.004687	60.60.1.1	8.8.8.8	GTP <ICMP>
19	4.007785	8.8.8.8	60.60.1.1	GTP <ICMP>
20	4.007793	8.8.8.8	60.60.1.1	GTP <ICMP>
21	5.006618	60.60.1.1	8.8.8.8	GTP <ICMP>
22	5.006640	60.60.1.1	8.8.8.8	GTP <ICMP>

Figure 5: Only ICMP packets sent from UE1 to 8.8.8.8 is captured by UPF2.

No.	Time	Source	Destination	Protocol
1	0.000000	60.60.2.1	8.8.8.8	GTP <ICMP>
2	0.000024	60.60.2.1	8.8.8.8	GTP <ICMP>
3	0.003208	8.8.8.8	60.60.2.1	GTP <ICMP>
4	0.003214	8.8.8.8	60.60.2.1	GTP <ICMP>
5	1.001168	60.60.2.1	8.8.8.8	GTP <ICMP>
6	1.001191	60.60.2.1	8.8.8.8	GTP <ICMP>
7	1.004259	8.8.8.8	60.60.2.1	GTP <ICMP>
8	1.004265	8.8.8.8	60.60.2.1	GTP <ICMP>
9	2.002179	60.60.2.1	8.8.8.8	GTP <ICMP>
10	2.002197	60.60.2.1	8.8.8.8	GTP <ICMP>
11	2.005338	8.8.8.8	60.60.2.1	GTP <ICMP>
12	2.005344	8.8.8.8	60.60.2.1	GTP <ICMP>
13	3.003219	60.60.2.1	8.8.8.8	GTP <ICMP>
14	3.003239	60.60.2.1	8.8.8.8	GTP <ICMP>
15	3.006365	8.8.8.8	60.60.2.1	GTP <ICMP>
16	3.006371	8.8.8.8	60.60.2.1	GTP <ICMP>
17	4.005375	60.60.2.1	8.8.8.8	GTP <ICMP>
18	4.005400	60.60.2.1	8.8.8.8	GTP <ICMP>
19	4.008663	8.8.8.8	60.60.2.1	GTP <ICMP>
20	4.008669	8.8.8.8	60.60.2.1	GTP <ICMP>
21	4.077185	PcsCompu_88:09:7f		ARP
22	4.077210	PcsCompu_88:09:7f		ARP

Figure 6: Only ICMP packets sent from UE2 to 8.8.8.8 is captured by UPF3.

8 Discussion

There are several points worth of discussing: In the initial network setup, we deployed seven virtual machines (VMs) to represent the various network components. These VMs were allocated specific roles, such as the Core Network VM, UPF VMs, and RAN and UE Group VMs. This allocation ensured that each component had its dedicated environment, promoting isolation and scalability. We verified that the UEs and RANs were properly configured and initialized, preparing them for data transmission.

To establish the connection between the RAN and UPF components, we configured the network adapters on each VM, including IP addresses, subnet masks, gateway addresses, and other settings. This configuration enabled communication between the VMs, establishing the necessary network connections. Additionally, gateways were set up in the PSA-UPF to direct data traffic from UEs through the appropriate gateway towards the intended destination. This ensured efficient and effective data transmission within the network.

We also focused on setting up PDU sessions for data transmission. This involved configuring the sessions in the AMF and SMF configuration files, registering them in the database, and associating the PDU sessions among the UPFs based on compatibility and matching criteria. When UEs connected to the Core Network via the RANs, the AMF registered their information and initiated session requests for the UEs to connect to their desired destinations. These steps ensured the correct establishment of PDU sessions and their association with the corresponding UPF entities.

Overall, these experiments showcased the successful deployment and configuration of the simulated 5G core network. The design effectively facilitated data transmission between UEs and the internet, demonstrating the functionality and performance of the deployed components. By simulating different scenarios and evaluating the results, we gained insights into the network's behavior and performance, ensuring its readiness for real-world implementations.

9 Conclusion

In conclusion, we presented the design and implementation of a simulator for modeling and simulating the behavior of UE, RANs, UPFs and core network components in a 5G network. The simulator showcased the successful data transmission between UEs and the internet, demonstrating the effectiveness of the designed network architecture and configurations.

In the future, it is important to consider the deployment of UE and RAN on real devices to better simulate the radio signal transmission between the UE and RAN. Currently, due to the lack of equipment for simulating radio signals, both UE and RAN are deployed on the same VM. However, with the availability of real-world equipment such as 5G UEs like Samsung S21 5G, APAL 5G MiFi, and gNodeB devices like Nokia gNodeB and Alpha gNodeB, the work can be extended to include real-world implementation and testing of entire network performance.

Additionally, there is potential to develop custom NFs within the 5G Core Network. Many works [5], [2] build their own NFs on top of the free5gc for their purposes. These custom NFs can serve various purposes, such as observing the global network abstract view and achieving network verification through direct human interaction. By designing an NF that provides a interface accepting natural language, operators can easily control the network path and make necessary configurations without requiring specific knowledge of network hardware configuration. This would enhance the flexibility and usability of the 5G Core Network, enabling operators to efficiently manage and control network operations.

Overall, the presented simulator and experimental results contribute to the understanding and development of 5G network deployments and configurations. The work serves as a foundation for further research and improvements in the field of 5G network simulation, leading to more efficient and reliable networks in the future.

References

[1] 3GPP. 5g; system architecture for the 5g system. *3GPP TS 23.501 version 15.2.0*, 2018.

[2] Ayoub Bergaoui, Youssouf Drif, Tanissia Djemai, Olivier Gremillet, Ahmad Al Sheikh, and Pascal Berthou. Demonstration of orchestration of 5g core network functions with a satellite emulator.

[3] José A del Peral-Rosado, Ronald Raulefs, José A López-Salcedo, and Gonzalo Seco-Granados. Survey of cellular mobile radio localization methods: From 1g to 5g. *IEEE Communications Surveys & Tutorials*, 20(2):1124–1148, 2017.

[4] Panagiotis K Gkonis, Panagiotis T Trakadas, and Dimitra I Kaklamani. A comprehensive study on simulation techniques for 5g networks: State of the art results, analysis, and future challenges. *Electronics*, 9(3):468, 2020.

[5] Cheng-Ying Hsieh, Yao-Wen Chang, Chien Chen, and Jyh-Cheng Chen. Design and implementation of a generic 5g user plane function development framework. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 846–848, 2021.

[6] Chia-Wei Liao, Fuchun Joseph Lin, and Yoichi Sato. Evaluating nfv-enabled network slicing for 5g core. In *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 401–404. IEEE, 2020.

[7] Robert MacDavid, Carmelo Cascone, Pingping Lin, Badhrinath Padmanabhan, Ajay Thakur, Larry Peterson, Jennifer Rexford, and Oguz Sunay. A p4-based 5g user plane function. In *Proceedings of the ACM SIGCOMM Symposium on SDN Research (SOSR)*, pages 162–168, 2021.

[8] Rui Silva, David Santos, Flávio Meneses, Daniel Corujo, and Rui L Aguiar. A hybrid sdn solution for mobile networks. *Computer Networks*, 190:107958, 2021.