Automated Generation of RAN Scenarios for Experimentation

Heitor Anglada, Cleverson Nahum, Glauco Gonçalves, Ilan Correa, Silvia Lins and Aldebaro Klautau

***Abstract*—This paper presents RANGEN, a tool that generates diverse RAN scenarios with the UE (User Equipments) and RAN simulator UERANSIM containers given a selected number of UEs and a selected number of gNBs (gNodeB), where certain containers are responsible for a pack of UEs and others for the gNB connected to these UEs. This tool enables the quick and scalable creation of test environments to develop and experiment with variable-size 5G RAN connected to a 5G core, using Free5GC, for example. This paper also analyzes the computational cost due to different configurations of UEs and gNBs on the 5G Core Virtual Network Functions. Results indicate stable CPU usage for core network functions (AMF, UPF) irrespective of gNBs and UEs variation, whereas CPU utilization of UE and gNB containers decreases with increasing gNBs. This behavior underscores RANGEN's efficiency and scalability in 5G network testing and experimentation.**

***Index Terms*—Network Function Virtualization, Virtual Network Functions, Docker.**

I. Introduction

The 5th generation core network (5GC) is an essential part of 5G networks designed to work with 5G New Radio and other wireless standards [1]. 5GC is intended to be highly flexible and modular, enabling different network functions to be implemented and scaled according to the specific needs of a use case. To conduct cost-effective and realistic end-to-end testing and experimentation of 5G networks, telecom operators and researchers may employ software models and software packages to simulate essential 5G components [2]. Currently, there is a range of open source 5GC implementations [3], and for simulating the Radio Access Network (RAN) components User Equipment (UE) and gNBs, there is UERANSIM 3.2.6, which allows building customized RAN setups.

However, for extensive 5G network simulations, manual configuration of both UEs and gNBs in UERANSIM can be time-consuming and error-prone as managing and configuring individual files becomes difficult with an increasing number of UEs and gNBs. Each container requires parameter settings, such as network addresses and the number of UEs, and International Mobile Subscriber Identity (IMSI), which must be manually replaced in various configuration files with sequential values based on the total number of UEs to be deployed and the network addresses of the gNBs, which are now embedded in UEs.

Heitor Anglada, Cleverson Nahum, Glauco Gonçalves, Ilan Correa and Aldebaro Klautau are with LASSE - Telecommunications, Automation and Electronics Research and Development Center, Belém-PA, Brazil . Silvia Lins is with the Innovation Center, Ericsson Telecomunicações S.A, Brazil. E-mails: heitor.anglada@itec.ufpa.br, cleversonahum, glaucogoncalves, ilan, aldebaro@ufpa.br) silvia.lins@ericsson.com). This work was supported by the Innovation Center, Ericsson Telecomunicações S.A., OpenRAN Brazil - Phase 2 project (MCTI grant Nº A01245.014203/2021-14), SAMURAI project (FAPESP grant #20/05127-2), and CNPq-Brasil (grant 405111/2021-5).

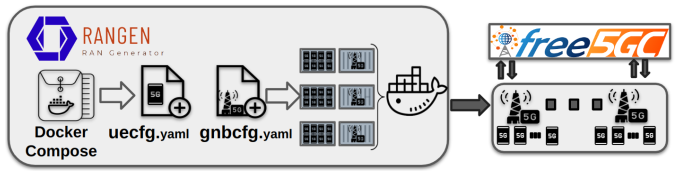

Fig. 1. RANGEN configuration files and overview of the scenario.

In the context of the [4], a tool capable of deploying gNB and UE containers could be a valuable asset for deploying the RAN components in a cloud-native environment. This approach emphasizes using microservices, containers, and dynamic orchestration to create flexible and scalable systems. In such an environment, the ability to automatically generate many UE and gNodeB containers would be highly beneficial.

This paper proposes RANGEN, a tool that automates the process of creating RAN scenarios for testing and experimentation. Using this tool, one can effortlessly generate multiple emulated UE and gNodeB components, which can be used to connect and generate traffic to the core network, allowing the testing of several different RAN setups. The paper describes this tool and shows some experiments and results.

II. RANGEN

RANGEN[1] is a shell script-based tool that automates the creation of UE and gNodeB containers in UERANSIM. By specifying the desired number of UEs connected to each gNodeB container and the total number of gNodeB containers, the script automates the generation of these containers. This tool eliminates manual creation and configuration, making deploying and managing many UEs and gNBs easier without writing additional code. Fig. 1 shows how RANGEN works. The first step is creating a Docker compose file, organized to build an equal number of UEs and gNBs containers. Each UE container can emulate a number of UEs, which will access the network through its respective gNB container. This entire process is based on predefined configuration templates.

After container creation, the next step involves substituting values. RANGEN replaces key values in the configuration files of UEs and gNBs. These parameters ensure the network operation and correct communication between the RAN components and the 5GC in the context of 5G. These values include the IMSI, UE number, and interface network addresses.

[1]https://github.com/lasseufpa/RANGEN

TABLE I
AVERAGE CPU USAGE IN DIFFERENT SCENARIOS

Scenario	UPF	AMF	UEs	gNB
1 gNB + 100 UEs	2.36%	1.12%	102.35%	12.96%
2 gNB + 50 UEs	2.39%	0.82%	15.38%	8.25%
4 gNB + 25 UEs	2.66%	0.95%	8.89%	5.11%
5 gNB + 20 UEs	2.41%	0.95%	6.78%	3.98%
LTS				

The third step is container initialization. The tool runs all the containers, in a safe order, through the Docker Compose file. It then executes the commands specified in this file on the gNB containers. The command set on the “start” file is also executed on the UE containers. This file is responsible for adequately booting the UEs and making them execute a command. By default, the UE runs the ping command to a destination server to analyze network connectivity.

III. EXPERIMENT

In this experiment, we deploy 100 UEs and analyze the average CPU usage of the 5GC network function, specifically the Access and Mobility Management Function (AMF) and the User Plane Function (UPF). Each UE establishes Internet Control Message Protocol (ICMP) traffic (ping) to the Internet.

We use Free5GC 3.3.0 to deploy the core network functions due to its open-source nature, community support, and compliance with 3GPP standards. The reason for selecting AMF and UPF for analysis is their critical roles in 5GC. AMF is responsible for session management, authentication, authorization, and mobility management. UPF handles forwarding user data packets, serving as the gateway for all UE traffic before the Internet. By examining the CPU usage of these components, we can gain insights into their performance and impact on the overall system.

We test four scenarios: 1 gNB and 100 UEs; 2 gNBs and 50 UEs per gNB; 4 gNBs and 25 UEs per gNB; and 5 gNBs and 20 UEs per gNB. Please note that a different number of containers is used in each case, respectively, 2, 4, 8, and 10 containers, since each gNB runs in an isolated Docker container and its associated UEs run in another container. We also measure the CPU usage of these containers, which are averaged according to this type (UE or gNB).

Moreover, we evaluate the time required for RANGEN to prepare each scenario. This aspect is particularly relevant when we scale the network infrastructure by setting up more gNBs.

The experiments were performed on a host running the long-term support Ubuntu 22.04.2. The machine was equipped with 8 GB of RAM (Random-access memory) and an Intel® Core™ i5-7600 CPU running at a clock speed of 3.50GHz, featuring four cores. To measure the CPU usage of the containers involved in the simulations during the experiments, we transformed and processed the output of the ”docker stats” command into a CSV file with data collected for 5 minutes of simulation, with data recorded every second.

IV. RESULTS

Table I provides the CPU utilization of the UPF and the AMF, as well as the containers of the UE and gNodeB. A

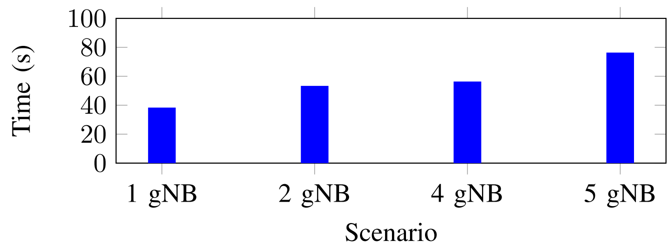

Fig. 2. Time taken by RANGEN to deploy each scenario.

careful analysis of the data in the table reveals that the UPF function, much like the AMF, does not follow a specific pattern but maintains a relatively constant CPU usage. This is expected since the number of gNBs is not expressive, and, in all cases, the number of UEs remains the same.

Regarding the other container types (UE and gNB), we observe a consistent pattern of decreasing average CPU usage across different scenarios. This occurs because as the number of gNBs grows, the load (from both types) is divided into multiple Docker containers, allowing better exploration of the number of processing cores in the host.

As illustrated in Fig. 2, an increase in the number of gNBs invariably increases the time necessary to complete the command execution. This observation aligns with our expectations, as a more extensive network setup naturally demands more resources and time for deployment.

V. CONCLUSION

UE and gNB container generation automation in UERANSIM offers significant benefits for 5G network simulations. However, currently, its configuration involves much manual effort for setup, which is laborious and error-prone. This way, RANGEN focuses on eliminating manual configuration, saving time, and ensuring a standardized approach for consistent RAN simulations.

While this work uses the Free5GC core for automation, the principles can be applied to other 5GC implementations, expanding the range of experiment options. Thus the tool allows researchers and developers to focus more on practical experimentation and analysis, thereby driving advancements in 5G technology. Further versions of RANGEN will include more diverse configurations provided by UERANSIM, allowing evaluation of use cases such as fault tolerance.

REFERENCES

[1] M. Wazid, A. K. Das *et al.*, “Security in 5G-Enabled Internet of Things Communication: Issues, Challenges, and Future Research Roadmap,” *IEEE Access*, vol. 9, pp. 4466–4489, 2021.

[2] R. Dayana, S. Krishnamurthy, A. Sarma *et al.*, “5G UE Simulation,” in *2023 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI)*. IEEE, 2023, pp. 1–5.

[3] F. J. D. S. Neto, E. Amatucci, N. A. Nassif, and P. A. M. Farias, “Analysis for Comparison of Framework for 5G Core Implementation,” in *2021 International Conference on Information Science and Communications Technologies (ICISCT)*. IEEE, 2021, pp. 1–5.

[4] O.-M. Ungureanu and C. Vlădeanu, “Leveraging the Cloud-Native Approach for the Design of 5G NextGen Core Functions,” in *2022 14th International Conference on Communications (COMM)*, 2022, pp. 1–7.