End-to-End Connectivity Test

Pierre Gavin Tan
Electrical Engineering Department
from
Institut Teknologi Bandung
Bandung, Indonesia

***Abstract*—This report presents an end-to-end connectivity test using OpenAirInterface (OAI) implemented in a virtualized environment set-up on VirtualBox. The primary objectives of this test include setting up a virtual machine, configuring core OAI components (CN5G, gNB, and nrUE), and evaluating network performance through round-trip time (RTT) measurements by ping test. The test was conducted to analyze the latency and transmission reliability. The results demonstrated 0% packet loss, an average RTT of 9.871 ms and 2.051 ms jitter indicating a stable and consistent 5G connection between the nrUE and CN. These results confirm that OAI can be deployed in a virtualized environment with low latency and reliable connectivity.**

Keywords—end-to-end connectivity, OAI, VirtualBox, configuration, RTT

I. Introduction

With the increasing adoption of 5G networks for wireless communication systems, researchers and engineers have been developing and experimenting with networks through simulated environments. OpenAirInterface (OAI) is an open-source project that provides the necessary tools for 4G/5G wireless communication system implementation that is often used for experimenting on 5G networks. With that in mind, this report presents an end-to-end connectivity test conducted using OAI to observe the functionality and stability of a simulated 5G network. The primary objectives of this test include :

- Setting up a virtualized testing environment to run OAI components
- Configuring the necessary parts of OAI such as CN5G, gNB, and nrUE
- Performing Round-Trip Time tests (RTT) to evaluate performance of connectivity

II. Theoretical Basis

A. Virtual Machines (VM)

[1] A Virtual Machine (VM) is a digitized version of a physical computer, or in other words, a computer system created using software to emulated functionality of another physical computer. Therefore, a VM can run isolated testing environments for different OAI components without interfering with the host system. For this test, Ubuntu Linux 24.04.2 LTS is used as the host operating system due to its compatibility with networking tools and OAI dependencies. To run a VM, Oracle VirtualBox is used for this test due to its general-purpose nature and its ease in configuration with simple interfaces and complete functionality while being free and open-source.

B. Overview of OpenAirInterface (OAI)

[2] OpenAirInterface (OAI) is an open-source project that implements 3GPP stack where the radio access network (RAN) that includes eNB, gNB and 4G, 5G UE as well as the core network EPC and 5G-CN are configured and set-up to satisfy the user's needs in a wireless communication system. The OAI source code is split into two projects. The first project being OAI-RAN that implements 4G LTE and 5G RAN with both NodeB and User Equipment. The second project being OAI-CN (Core Network) which implements 4G LTE Evolved Packet Core (EPC) and 5G Core Network.

C. Round-Trip Time (RTT)

[3] Round-trip time (RTT) is the duration in milliseconds it takes for a network request to go from a starting point to a destination and back again to the starting point. RTT is an important metric in determining the condition of a connection on a network and is commonly utilized by network administrators to diagnose the speed and reliability of network connections. Improvements in latency can be measured in the reduction of round-trip time. The ping utility, available on most computers, is a method of estimating RTT between two network addresses. The common factors that affect RTT are nature of the transmission medium, LAN traffic, server response time, node count and congestion, as well as the physical distance between the starting point and the destination of the trip.

III. Methodology

This section details the step-by-step process taken to conduct the end-to-end connectivity test as well as the pre-requisites needed to be fulfilled before conducting the test.

A. Pre-Setup Requirements

Minimum hardware requirements to conduct this test include a laptop/desktop/server for OAI CN5G and gNB with the specifications listed as :

- Operating System : Ubuntu 24.04 LTS
- CPU : 8 cores x86_64 @ 3.5 GHz
- RAM : 32 GB

Along with a laptop for OAI nrUE with the specifications listed as :

- Operating System : Ubuntu 24.04 LTS
- CPU : 8 cores x86_64 @ 3.5 GHz
- RAM : 8 GB

For this test, a laptop with the specifications listed below is used :

- Operating System : Windows 11 Home 64-bit
- CPU : 20 cores x86_64 @ 2.7 GHz
- RAM : 16 GB + 8 GB shared from GPU

Using VirtualBox, a virtual machine is set up on the laptop with the following specifications :

- Operating System : Ubuntu 24.04.2 LTS
- CPU : 8 cores x86_64 @ 2.7 GHz
- RAM : 8 GB

B. *Configuration for VM using VirtualBox*

- Install Ubuntu 24.04.2 LTS ISO File from the official Ubuntu website
- Install Oracle VirtualBox on the host machine
- Create a new virtual machine and assign the operating system, memory, and other system specifications for the virtual machine accordingly
- Start up the virtual machine and complete the Ubuntu software startup configuration

C. *Configuration for OAI CN5G*

- Open terminal on Ubuntu and install the necessary dependencies for OAI CN5G
- Reboot the system
- Download OAI CN5G configuration files
- Pull OAI CN5G docker images
- Start OAI CN5G and stop if needed

D. *Configuration for OAI gNB and OAI nrUE*

- Setup pre-requisites by building UHD from source
- Build OAI gNB and OAI nrUE
- Make sure OAI CN5G is running
- Run OAI gNB with the network interface used, in this case RFSimulator for a fully virtual connection
- Run OAI nrUE in another terminal with the same network interface used by OAI gNB

E. *End-to-end Connectivity Test*

- Ensure all configurations are properly implemented
- Open another terminal for the test
- Ping from the UE host to the CN5G on the address 192.168.70.135 from oaitun_ue1
- Observe the packets being sent and received
- After a few pings, press Ctrl+C to stop the test and get the measured round-trip time (RTT)

IV. Analysis and Discussion

Since the advised specifications for OAI CN5G and OAI gNB were not used, where the RAM and the processor's clock speed were lower than the requirement. Some of the functionality during the end-to-end connectivity test might deviate from the expected outcome.

During the OAI configuration, there were several problems that were encountered. However, the problems were eventually solved after deeper observation and simple debugging. The first problem encountered was an undetected USRP device, or the network interface for OAI gNB shown in figure 1.

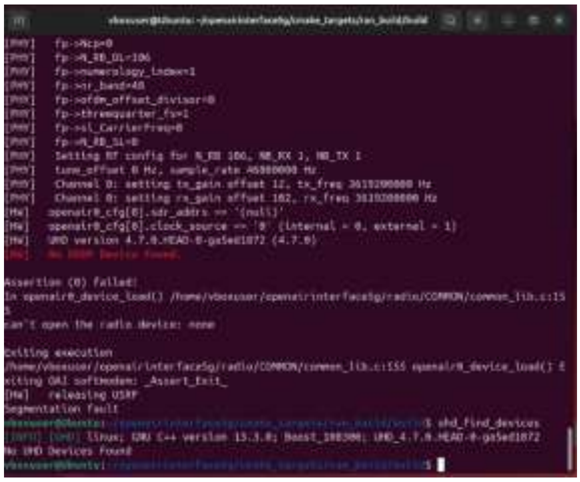

Fig. 1. Network Interface Error for OAI gNB

This common issue occurred because of the wrong network interface chosen during the configuration phase, hence choosing the RFSimulator interface solved this issue and OAI gNB was successfully started up as shown in figure 2.

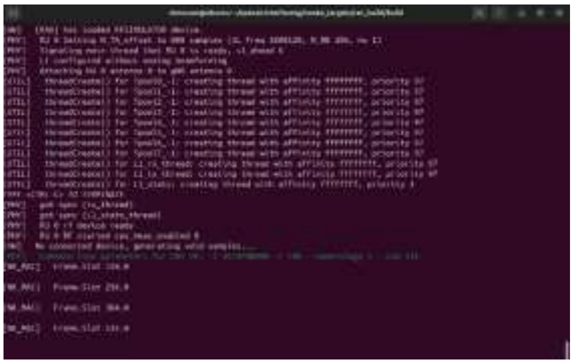

Fig. 2. OAI gNB terminal running successfully

From the terminal, it is observed that there are frame slots that update 128 incrementally and eventually resets after reaching 896. These frame structure is a common slot timing method applied to 5G NR that divides time into frames, subframes, and slots. There are key parameters in the 5G network's physical layer frame structure for different numerologies, which indicates different characteristics in subcarrier spacing, OFDM symbol duration, cyclic prefix duration, and total duration of the OFDM symbol including the cyclic prefix. These numerology values range from 0 to 4 and from the green line in the terminal shown in figure 2, it is stated that the environment is set-up with the numerology value $\mu = 1$ with the subcarrier spacing 30 kHz and a subframe with 2 slots in it, which means the radio frame contains 20 slots and a number of OFDM symbols within a slot is 14. The fact that the frame slots are resetting every time it reaches 896 shows that the system specification concerns we had earlier might be causing inconsistencies coming from the CPU processing limitations and jitter from the system itself. Since the OAI environment is ran in a virtualized machine, frame desynchronizations or unexpected counts might appear in the terminal. However, no error logs were observed after running the OAI gNB, hinting that the system might be using a continuous slot counter before rolling over by tracking the slots globally instead of resetting every 200 slots (according to the expected slot).

The second error was found during the OAI UE startup process. There was a missing configuration file in an unspecified directory "./nr-uesoftmodem" which assumably was not included directly in the initial build. This problem was solved by creating the config file directly with the nano command and adding the file shown in figure 3.

```
uicc0 = {
  imsi = "001010000000001";
  key = "fec86ba6eb707ed08905757b1bb44b8f";
  opc = "C42449363BBAD02B66D16BC975D77CC1";
  dnn = "oai";
  nssai_sst = 1;
}
```

Fig. 3. nrUE missing configuration file

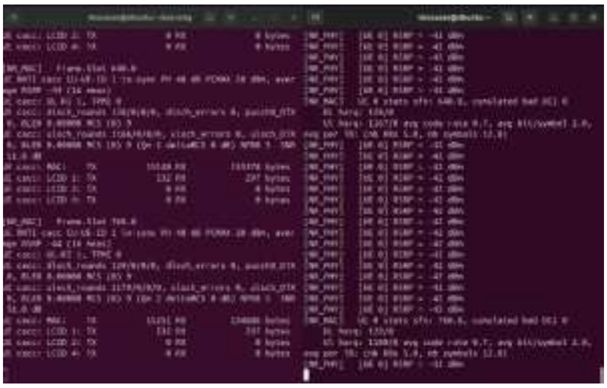

Fig. 4. OAI environment initial startup (left side gNB, right side UE)

Next, the OAI CN5G, gNB, and nrUE are started up simultaneously as shown in figure 4. It is observed that on the frame index 640 with slot number 0 marks that 32 full radio frames have passed. Within this frame, the "in-sync" status indicates the successfully synchronization between OAI UE and gNB. Next are the logs observed from the OAI gNB which explains the characteristics of the signals being transmitted and received within the network. PH 48 dB marks the pathloss, PCMAX 20 dBm marks the maximum transmitted power from UE, and the RSRP -44 dBm marks the measurements taken where ideally the value is closer to zero. These logs indicate that the signal have a small signal attenuation (from pathloss), a good PCMAX value, and a strong signal measured by the RSRP value. The downlink (tranmission from gNB to UE) performance indicate that there are 128 DL transmissions attempted with 0 errors, no discontinuous transmission issues, no packet loss, and 9 data packed in each transmission. The uplink (transmission from UE to gNB) performance indicates that there are 1166 uplink transmissions attempted with 0 errors, no failed UL transmissions, no packet loss, very strong signal-to-noise ratio of 51 dB (indicating signal unlikely to be interrupted by noise), 5 physical resource blocks used, quadrature phase shift keying (QPSK) used meaning each symbol carries 2 bits. These logs indicate that there are no errors between transmitting data and receiving data within the network of UE and gNB. Below those logs, there are UE cacc logs that indicate the bytes being transferred between the network. The MAC layer shows that there are 15140 bytes sent by UE through the TX channel and 1333378 bytes received by UE through the RX channel. Meanwhile the LCID (logical channel) shows that for LCID 1 which corresponds to the main data channel there were 532 bytes in the TX channel and 297 bytes in the RX channel, with no traffic detected in LCID 2 and 4.

For the UE log, some other characteristics such as RSRP, HARQ performance, and the physical layer statistics. The Reference Signal Received Power (RSRP) shows -42 dBm which deviated slightly from the OAI gNB value, but is still consistent within the acceptable range of tolerance. The next log shows the frame and slot countt "640.8" marking the system frame number (sfn) in slot 8 indicating the environment updating the slot timings properly and synchronizes the OAI UE and gNB with no errors. The Hybrid ARQ Retransmissions (HARQ) performance was also measured with 0 retransmissions for both downlink and uplink marking a stable connection with perfect linking. The next line explains the code rate and modulation efficiency with 70% of transmitted bits carrying useful data while 30% of them are error correction bits as well as the average bits per symbol being 2.0 marking the QPSK modulation from the gNB logs working properly. The transport block statistics line marks that the UE is allocated 5 PRBs for data transmission and around 13 OFDM symbols per slot which indicates a slight difference from the value of 14 OFDM symbols per slot. This means that there are some symbols reserved for control data or scheduling.

After confirming the connection between the configured OAI gNB and UE, the end-to-end connectivity test is going to be conducted through a ping test. A slight problem was also found here where the OAI UE was not connected properly because of unintended disabling of the UE when running the ping test as shown in figure 5. This log shows that the ping test can only be ran at the destination of 10.0.0.2 shown on figure 6.

```
vboxuser@Ubuntu:~/openairinterface5g/cmake_targets/ran_build/build$ ip route show
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.0.0/24 dev oaitun_ue1 proto kernel scope link src 10.0.0.2
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
192.168.70.128/26 dev oai-cn5g proto kernel scope link src 192.168.70.129
vboxuser@Ubuntu:~/openairinterface5g/cmake_targets/ran_build/build$
```

Fig. 5. Routing check between OAI UE and OAI gNB

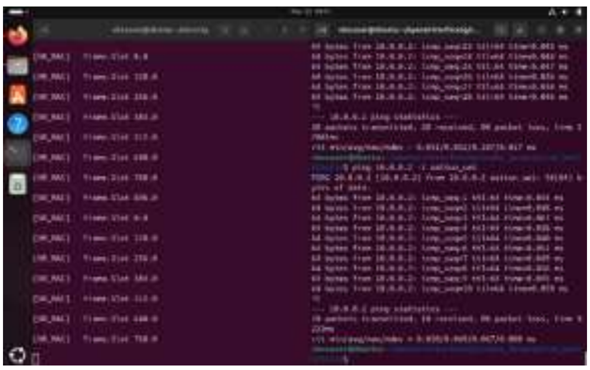

Fig. 6. Disconnected problem during ping test

These logs between the terminals shows that the OAI gNB and UE is not connected properly. The previous packets between the UE and gNB shown on figure 4 is not present in this test. The timing values shown on the right-side of the terminal can not be concluded as a successfully round-trip time test since it's most likely a signal sent locally and returned without passing through the OAI gNB and UE connection that was expected. Hence, the connection between the OAI gNB and UE neeeded to be restored. This issue is resolved by restating both the OAI gNB and the OAI nrUE in different terminals from the testing environment. This means that there needed to be three different terminals, each serving their own purpose for OAI gNB, OAI nrUE, and for the connectivity test itself.

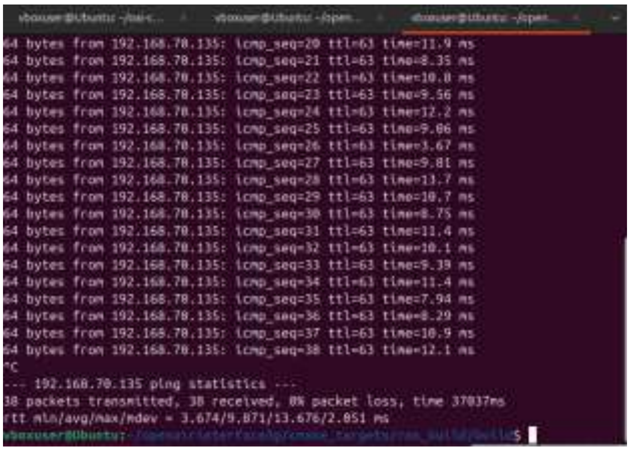

Fig. 7. End-to-end connectivity test result

After resolving the connectivity issue, the end-to-end connectivity test through pinging the address 192.168.70.135 through oaitun_ue1 provides a more logical result with observably longer time values than the previous result. From the ping test, there were 38 packets transmitted with 38 received marking that there is no packet loss. This indicates a stable, reliable, and functional end-to-end link with no disruption between the UE and the CN. From the RTT statistics, the minimum latency is 3.674ms, average at 9.871 ms, maximum latency at 13.676ms, and the maximum jitter at 2.051 ms. These time values indicate that the average RTT for the environment is good, especially for a virtualized test setup. Along with that, the jitter of around 2 ms marks a consistent network performance. There is one more thing to note from the logs which are the ttl values that stays consistently 63 marking a one hop network which is from the UE to the gNB and CN directly marking a direct and minimal network traversal, which is to be expected from the procedures taken.

V. Conclusion

Through the process of conducting this end-to-end connectivity test, there are several insights that can be concluded to complete the objectives that were initially set which are listed as :

- From the virtual environment setup for OAI, there are physical limitations that could affect performance during network testing.
- The OAI gNB and nrUE connection must be synchronized properly and confirmed through its stability and reliability.
- RTT measurements indicate a low latency and consistent connectivity within the environment.

References

[1] "What is a virtual machine? VM uses and benefits | google cloud," Google, https://cloud.google.com/learn/what-is-a-virtual-machine?utm_source=google&utm_medium=cpc&utm_campaign=japac-SG-all-en-dr-SKWS-all-all-trial-DSA-dr-1710102&utm_content=text-ad-none-none-DEV_c-CRE_655856181068-ADGP_Hybrid%2B%7C%2BSKWS%2B-%2BBRO%2B%7C%2BDSA%2B-All%2BWebpages-KWID_39700076131769766-dsa-1456167871416&userloc_9120583-network_g&utm_term=KW_&gad_source=1&gclid=Cj0KCQjw16O_BhDNARIsAC3i2GCHhHpz2B2EENvlKHdVxLtuE3YwjvTB3shQi6bojn0ftAXkyQy4GqgaAjQmEALw_wcB&gclsrc=aw.ds (accessed Mar. 31, 2025).

[2] "Overview," OpenAirInterface, https://openairinterface.org/overview/ (accessed Mar. 31, 2025).

[3] What is round-Trip time (RTT)? | RTT meaning | cloudflare, https://www.cloudflare.com/learning/cdn/glossary/round-trip-time-rtt/ (accessed Mar. 30, 2025).

[4] "OAI / openairinterface5G · GITLAB," GitLab, https://gitlab.eurecom.fr/oai/openairinterface5g (accessed Mar. 31, 2025).

[5] "What is Round trip time (RTT): Behind the ping: Cdn guide: Imperva," Learning Center, https://www.imperva.com/learn/performance/round-trip-time-rtt/#:~:text=Round%2Dtrip%20time%20(RTT),load%20time%20and%20network%20latency. (accessed Mar. 31, 2025)