차세대셀룰러통신

보고서

실습1

주민철 교수님

서울시립대학교

전자전기컴퓨터공학부

학번: 2019440100

이름: 이준헌

목차

1. 개요

1.1 실습 목적

srsRAN 서버를 이용하여 5G 네트워크 시뮬레이션 환경을 구축해보고, 동작 원리에 대한 이해도를 높인다. O-RAN 기술과 Core Network, gNB, RIC, UE 등을 실습하여 네트워크 동작 방식을 실제 환경에서 적용시킨다.

2. 실습 환경 및 준비

2.1 실습 환경 및 설명

2.1.1 소프트웨어

srsRAN은 네트워크 시뮬레이션을 지원하는 오픈 소스 소프트웨어로 해당 실습에서는, srsRAN에서 제공하는 자원들을 통하여 각 네트워크 요소들을 실행해보고 상호작용할 수 있도록 실습하였다.

mobaXterm은 원격으로 서버에 접속하여 관리할 수 있도록 하는 터미널 소프트웨어이다. 이를 통하여, srsRAN 서버에 SSH를 통하여 연결하였다.

2.1.2 SSH

MobaXterm에서 SSH 프로토콜을 통해 실행하고, host ip 203.246.112.10~19 중 하나로 설정한 후, port는 10022번을 이용하였다.

2.1.3 구성 요소

- 5G Core

5G 네트워크의 핵심 제어 및 데이터 처리를 담당한다.

- RIC

RAN Intelligent Controller의 약자로서, RAN의 최적화 및 관리를 담당한다.

- gNB

5G 네트워크의 기지국 역할을 담당한다. UE와 네트워크를 연결한다.

- UE

사용자가 5G 네트워크에 접속하는 장치를 의미한다.

3. 실행 및 과정

해당 실습에 앞서서 모든 과정은

```
# sudo su - root
```

우선 위 명령어를 사용하여 관리자 모드에서 실행하였다.

3.1 5G Core 실행

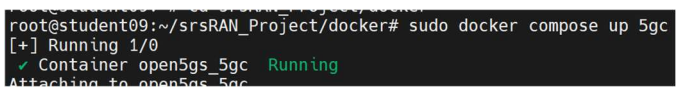
우선

```
# cd ~/srsRAN_Project/docker/
```

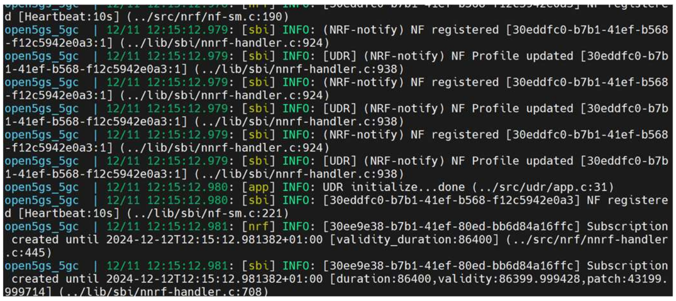
디렉터리로 이동하여,

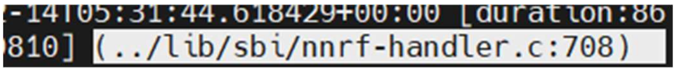
```
# sudo docker compose up 5gc
```

명령어를 실행하여, 5G Core을 실행시켰다.



(중략)





위와 같이 5G Core가 제대로 실행되는 것을 확인할 수 있다.

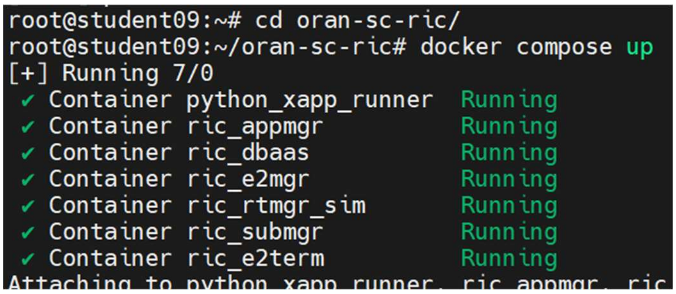
3.2 RIC 실행

```
# ~/oran-sc-ric
```

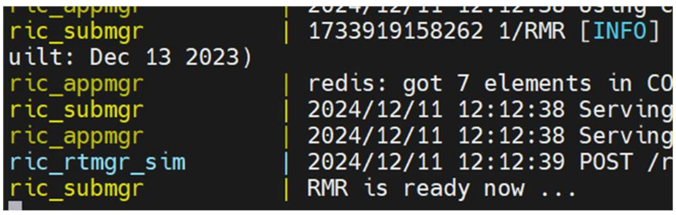
해당 RIC 디렉터리로 이동하여,

```
# sudo docker compose up
```

명령어를 실행하여 RIC를 실행하였다.



(중략)



위와 같이

```
RMR is ready now...
```

메시지가 출력되는 것을 보아, RIC가 제대로 실행되는 것을 확인할 수 있다.

3.3 gNodeB 실행

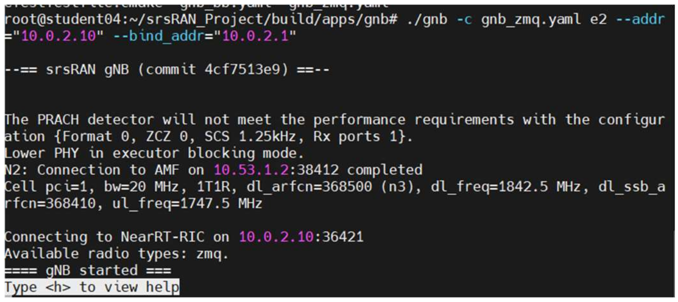
```
# ~/srsRAN_Project/build/apps/gnb
```

해당 디렉터리로 이동한다.

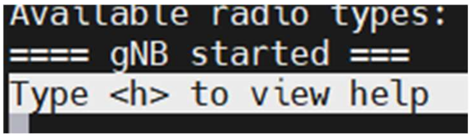
```
# wget -O gnb_config.yaml
https://docs.srsran.com/projects/project/en/latest/_downloads/cd7bc1de5ec01c261b2b112e21700e77/gnb_zmq.yaml
```

위 명령어를 실행하여, 설정 파일을 다운로드한다.

```
# ./gnb -c gnb_zmq.yaml e2 --addr="10.0.2.10" --bind_addr="10.0.2.1"
```



위와 같이 gnb를 실행하였다.



gNode가 제대로 실행되는 것을 확인할 수 있다.

3.4 UE 실행

```
# ip netns add ue1
```

위 명령어를 통하여, 네트워크 Name space를 생성한다.

```
# ~/srsRAN_4G/build/srsue/src
```

해당 디렉터리로 이동한다.

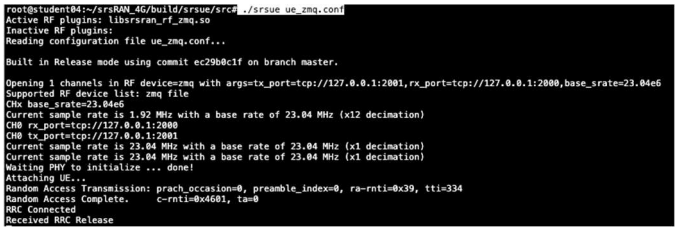
```
#wget -O srsue_config.yaml
https://docs.srsran.com/projects/project/en/latest/_downloads/fada80ca15bd69b34a0ad8425bbc6560/ue_zmq.conf
```

UE 실행에 필요한 설정 파일을 다운로드한

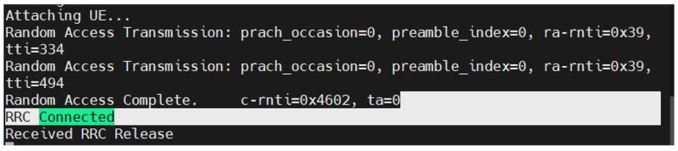
다.

```
# ./srsue ue_zmq.conf
```

그리고 위와 같이 실행하였다.



위와 같이



메시지가 출력되는 것으로 보아, UE 또한 제대로 실행되는 것을 확인할 수 있다.

이를 통해 성공적으로 5GC부터 UE까지 연결된 것을 확인할 수 있었다.

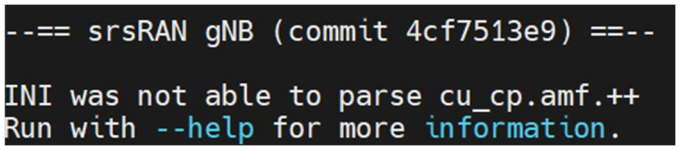
4. (교안 수정 전) 문제 상황 및 고찰

해당 부분은 교안 수정 전 오류 확인 및 탐구 과정을 다룬다.

4.1 gNodeB 실행 중 오류


```
# ./gnb -c gnb_config.yaml e2 --
addr="10.0.2.10" --bind_addr="10.0.2.1"
```

위와 같이 gNodeB를 실행하는 경우,



위와 같이 실행되지 않은 것을 확인할 수 있었다.

4.2 문제 원인 분석

해당 오류 문구 INI는 cu_cp.amf의 특정 세션을 읽지 못했다는 것을 의미한다. cu_cp.amf 부분에 문제가 없다면, e2ap_enable이 true로 설정되어 있을 때, e2ap 데이터를 기록한다. 이때, gnb_e2ap.pcap에 문제가 생기면 설정 파일을 올바르게 읽더라도 실행이 중단되게 된다.

우선, cu_cp.amf 부분의 주소 지정에 문제가 없는지 확인하고, gnb_e2ap.pcap 부분에서 문제 원인을 찾아보았다. gnb_config.yaml의 다른 부분은 상대적으로 단순하여 오류를 유발할 부분이 적어서, 해당 부분을 중점으로 분석하였다.

4.3 해결 시도

4.3.1 cu_amf 주소 오류 확인

cu_amf 부분을 파싱하지 못하여 생긴 원인이므로, yaml 파일의 해당 부분을 확인하였다.

```
cu_cp:
  amf:
    addr: 10.53.1.2
    bind_addr: 10.53.1.1
```

Addr과 bind_addr은 각각 10.53.1.2,

10.53.1.1이었다.

Addr은 gNodeB가 연결하려는 AMF의 주소이이고, bind_addr은 gNodeBd의 로컬 주소이다. 먼저 5gc에서 amf의 실행 여부를 확인하고 해당 주소를 확인하기 위해, 컨테이너 내부로 들어가 해당 프로세스를 확인한다. 이때 docker ps 명령어로 확인한 컨테이너의 이름은 open5gs_5gc이었다.

```
docker exec -it open5gs_5gc bash
```

위 명령어로 컨테이너 내부에 들어가여, amf.yaml을 확인한 결과는 아래와 같다.

```
amf:
  sbi:
    server:
      - address: 127.0.0.5
        port: 7777
    client:
      scp:
        - uri: http://127.0.0.22:7777
  ngap:
    server:
      - address: 10.53.1.2
```

ngap의 address가 10.53.1.2으로 제대로 설정되어 있는 것을 확인할 수 있었다.

4.3.2 pcap 파일 여부 확인

```
# ls -l /tmp/gnb_e2ap.pcap
```

위와 같이 확인해본 결과,

```
-rw-r--r-- 1 root root 0 Dec 11 12:51 /tmp/gnb_e2ap.pcap
```

해당 파일이 존재하는 것을 확인할 수 있었다. 해당 파일이 존재하지 않아 생긴 오류가 아님을 확인할 수 있다.

4.3.3 YAML 설정 파일 내 pcap.e2ap_filename의 파일명과 일치 여부 점검

```
pcap:
  mac_enable: false
  mac_filename: /tmp/gnb_mac.pcap
  ngap_enable: false
  ngap_filename: /tmp/gnb_ngap.pcap
  e2ap_enable: true
  e2ap_filename: /tmp/gnb_e2ap.pcap
```

Yaml 파일에서 해당 부분의 내용을 확인하여, 경로와 파일명이 일치하는 지 확인하였으나 문제가 없었다.

4.3.4 파일 권한 수정

```
# chmod 755 /tmp/gnb_e2ap.pcap
```

위와 같이 해당 파일의 접근 권한을 수정하고 다시 실행시켜 보았으나, 같은 오류로 실행되지 않았다.

4.3.5 파일 경로를 변경하고 YAML 파일을 수정하여 경로 재설정

```
pcap:
  mac_enable: false
  mac_filename: /tmp/gnb_mac.pcap
  ngap_enable: false
  ngap_filename: /tmp/gnb_ngap.pcap
  e2ap_enable: true
  e2ap_filename: /test/gnb_e2ap.pcap
```

위와 같이 /test 디렉터리로 수정하고

```
# mkdir /test
# cp /tmp/gnb_e2ap.pcap /test/
```

해당 파일 또한 이동하였으나, 같은 이유로 실행되지 않았다.

5. 결과 분석 및 결론

해당 실습에서는 5GC와 UE 간의 네트워크

연결을 Core Network, gNB, RIC, UE 등의 단계로 구분하여, O-RAN 기술과 네트워크 동작 방식을 실제 환경에서 구현하여 보았다.

gNodeB 실행 중 발생한 오류는 설정 파일과 환경이 제대로 맞물지 않아서 생긴 문제라고 예상하였으나, 해당 오픈 소스를 최신화 하지 못하여 생긴 문제였다.

특히나 해당 INI 오류 메시지는 구체적인 문제 원인을 명시해주지 않아 더더욱 어려움이 있었다. Amf 주소, YAML 파일과 gnb_e2ap.pcap 생성 여부부터 파일의 접근 권한과 경로 등 많은 시간을 할애해서 확인하였으나, 동일한 오류로부터 벗어나지 못하였다. 이는 설정 파일이 올바르게 구성되더라도, 환경이나 다른 요소들이 문제의 원인이 될 수 있음을 알 수 있었다.

결국 최신화 된 방법을 찾아 문제를 해결하였으며, 최종적으로 Core Network, gNB, RIC, UE 모두 정상적으로 작동시키고, 테스트 또한 성공적으로 작동하는 것을 확인할 수 있었다.

이번 실습을 통해 O-RAN의 기초적인 구조를 피부로 받아들일 수 있게 되었을 뿐만 아니라, 오픈 소스를 활용하는 실험 환경에서 갖춰야 할 새로운 식견도 배울 수 있었다. 특히 오류를 맞이하였을 때 환경에 대한 부족한 이해로 인해 제대로 원인을 규명하지 못했다. 이를 통해 디버깅 과정에서의 부족함과 실습 환경에 대한 부족함을 실감할 수 있었다. 차후에는 다양한 사례들을 참조해가며 원초적인 능력 자체를 배양하여 유사한 문제를 예방할 수 있는 해결 능력을 키워야 한다고 생각한다.