教育部 5G 行動寬頻人才培育跨校教學聯盟計畫

5G 行動網路協定與核網技術聯盟中心示範課程

4G/5G 行動寬頻協同網路

實驗四

5G Emulator 仿真模擬實驗

副教授：吳俊興

助教：林原進、吳振宇

國立高雄大學 資訊工程學系

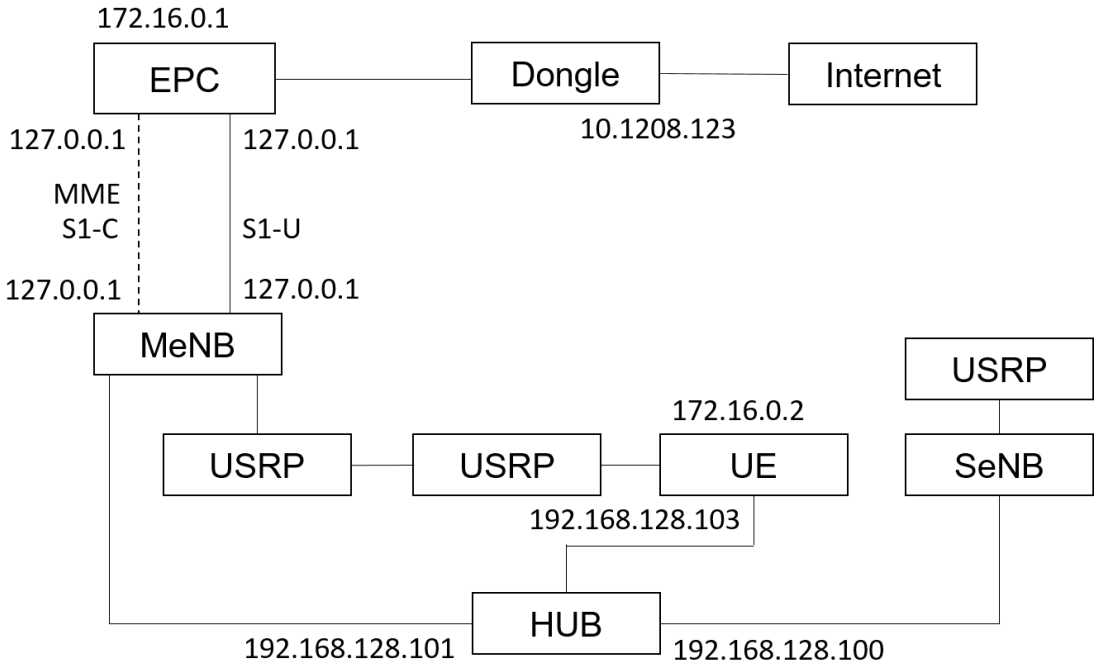
目錄

實驗四 1
一、 平台架構 4
1. 實驗架構 4
2. 實驗環境 4
二、 軟硬體需求 5
1. 硬體 5
2. 軟體 5
三、 環境安裝 7
1. Linux Kernel 安裝 7
1.1. 安裝 Kernel 7
1.2. 安裝過程 7
1.3. 修改開機選單和設定 7
1.4. 更新 grub 設定 8
1.5. 檢查 Kernel 版本 9
2. 安裝相關套件 9
2.1. 一般套件 9
2.2. RF Front-end Driver 9
2.3. mbed TLS 9
2.4. srsGUI 9
2.5. srsLTE 10
四、 設定 srsLTE 12
1. 設定 EPC 12
2. 設定 eNB 12
2.1. 修改 conf 檔 12
2.2. 修改 lwaap_entity.h 檔案 13
3. 設定 UE 13
4. 重新編譯 14
五、 執行 srsLTE 15
1. 執行 EPC 15
2. 執行 eNB 15
3. 執行 UE 16
六、 測試 18
1. 互通測試 18
2. 調配封包傳送比例 18
3. Wireshark 介面查看 19

一、 平台架構

1. 實驗架構

本實驗架構如圖所示，共使用三台電腦，分別開啟 EPC 和 MeNB、SeNB、UE，三台電腦皆連至 HUB 供內網傳輸，另外 EPC 額外增加外接網卡，供對外連線使用，MeNB 和 UE 則是透過 USRP 和 SMA 線連接。



2. 實驗環境

二、 軟硬體需求

1. 硬體

名稱	規格	數量	目的
EPC+eNB1	電腦型號 : ASUS VivoMini UN65H	1	啟動 MME,S-GW,P-GW
	Ethernet Network Cards	2	一張連接內部網路 (PCI-E : Realtek RTL8111/8168) 一張連接外部網路 (USB : TP-LINK UE300)
	USRP B210	1	啟動 srsLTE eNB
eNB2	電腦規格 : CPU : i7-6700 , RAM : 32G	1	模擬第二個基地站
	USRP B210	1	啟動 srsLTE eNB
UE	電腦型號 : ASUS NB M580V	1	模擬 UE
	USRP B210	1	啟動 srsLTE UE
Hub	型號 : TP-LINK WR1043ND	1	分配內部網路

2. 軟體

名稱	軟體	版本
EPC	OS : Ubuntu	Ubuntu 16.04
		Kernel : 4.15.0-041500-lowlatatency
	srsLTE EPC	srsLTE 18.06.1 470953bf9c5875646e4d5049c8f213d202fa84fd
eNB	OS : Ubuntu	Ubuntu 16.04
		Kernel : 4.15.0-041500-lowlatatency
	srsLTE eNB	srsLTE 18.06.1 470953bf9c5875646e4d5049c8f213d202fa84fd

UE	OS : Ubuntu	Ubuntu 16.04
		Kernel : 4.15.0-041500-lowlatatency
	srsLTE UE	srsLTE 18.06.1 470953bf9c5875646e4d5049c8f213d202fa84fd

三、 環境安裝

1. Linux Kernel 安裝

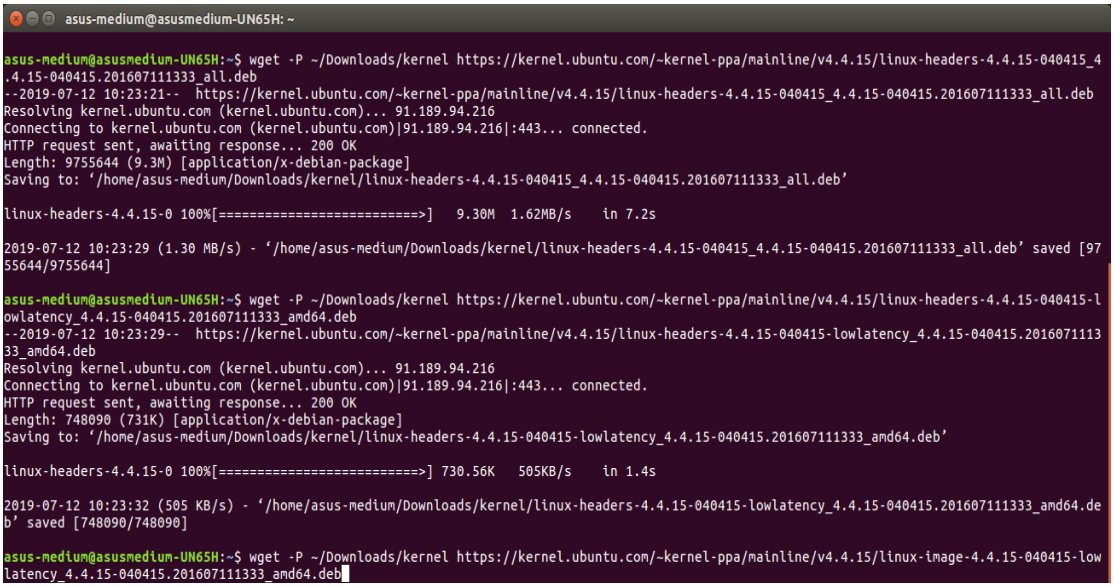
1.1. 安裝 Kernel

開啟一個終端機(Terminal)，並且依序輸入

wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-headers-4.4.15-040415_4.4.15-040415.201607111333_all.deb
wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-headers-4.4.15-040415-lowlatency_4.4.15-040415.201607111333_amd64.deb
wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.4.15/linux-image-4.4.15-040415-lowlatency_4.4.15-040415.201607111333_amd64.deb
sudo dpkg -i ~/Downloads/kernel/*.deb

1.2. 安裝過程

※請先確認是否連接網路正常



1.3. 修改開機選單和設定

開啟終端機(Terminal)輸入以下指令

```
sudo gedit /etc/default/grub
```

找到下列文字

GRUB_HIDDEN_TIMEOUT=0

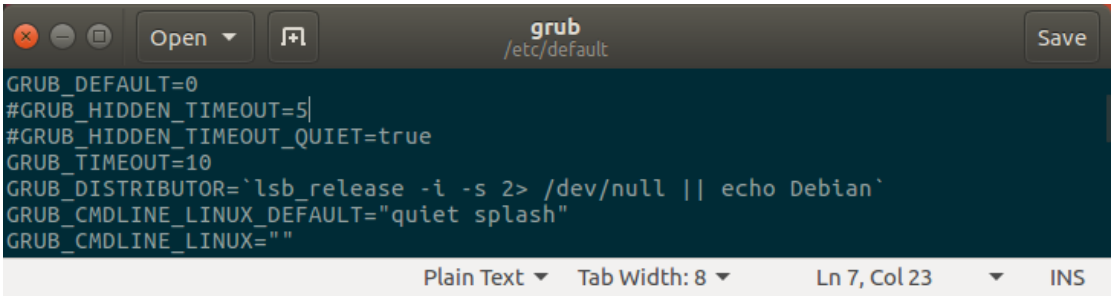
GRUB_HIDDEN_TIMEOUT_QUIET=true

改成

#GRUB_HIDDEN_TIMEOUT=0

#GRUB_HIDDEN_TIMEOUT_QUIET=true

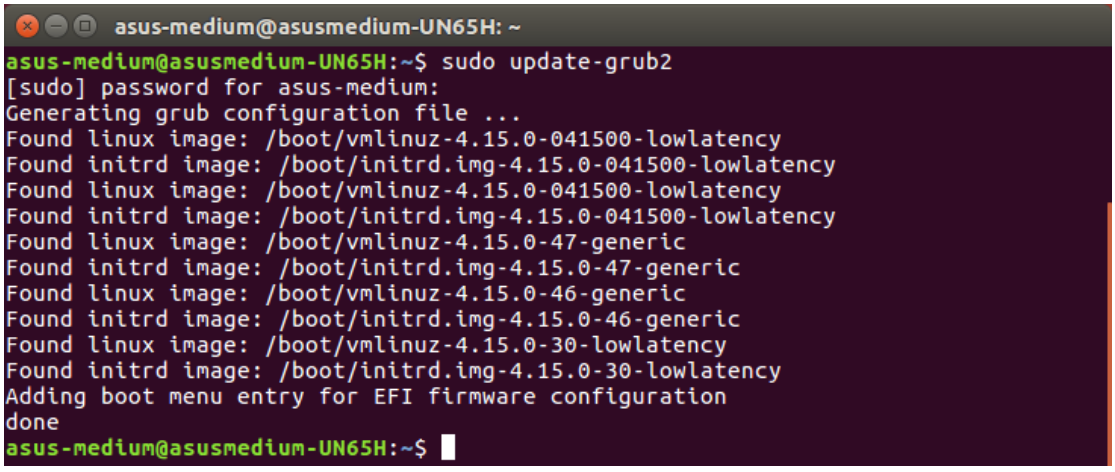
儲存後離開



1.4. 更新 grub 設定

開啟終端機並輸入以下指令

```
sudo update-grub2
```



接著輸入以下指令，重新啟動電腦

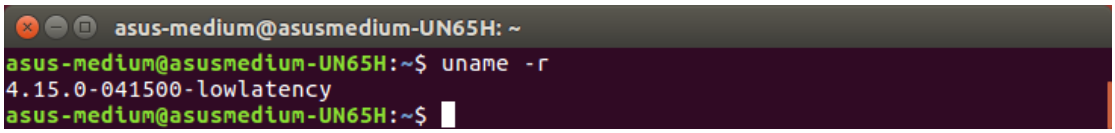
```
sudo reboot
```

然後在開機選單選擇剛才安裝的 lowlatency

1.5. 檢查 Kernel 版本

重新開機後在終端機輸入指令，確認版本

uname -r



2. 安裝相關套件

2.1. 一般套件

開啟終端機並輸入

sudo apt-get install cmake libfftw3-dev libboost-all-dev libconfig++-dev libsctp-dev

2.2. RF Front-end Driver

開啟終端機並依序輸入

sudo add-apt-repository ppa:ettusresearch/uhd
sudo apt-get update
sudo apt-get install libuhd-dev libuhd003 uhd-host
python3 /lib/uhd/utils/uhd_images_downloader.py

2.3. mbed TLS

開啟終端機並依序輸入

wget https://tls.mbed.org/download/start/mbedtls-2.6.0-apache.tgz
tar zxvf mbedtls-2.6.0-apache.tgz
sudo mv ~/Donwload/mbedtls-2.6.0 /usr/local
cd cd /usr/local/mbedtls-2.6.0
cmake .
make
make test
cmake -DENABLE_TESTING=Off .
cmake -DUSE_SHARED_MBEDTLS_LIBRARY=On .
sudo make install library

2.4. srsGUI

開啟終端機並依序輸入

sudo apt-get install libboost-system-dev libboost-test-dev libboost-thread-dev libqwt-dev libqt4-dev
git clone https://github.com/srsLTE/srsGUI.git
cd ~/srsgui
mkdir build
cd build
cmake ../
make
make test

2.5. srsLTE

開啟終端機並依序輸入

git clone https://github.com/nukcsie2066/nukxDC.git
cd srsLTE
mkdir build
cd build
cmake ../
make
make test
sudo make install

```
ue@ue-X580VD: ~/Desktop/srsLTE/build
ue@ue-X580VD:~$ cd ~/Desktop/srsLTE/
ue@ue-X580VD:~/Desktop/srsLTE$ mkdir build
ue@ue-X580VD:~/Desktop/srsLTE$ cd build/
ue@ue-X580VD:~/Desktop/srsLTE/build$ cmake ../
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- CMAKE_SYSTEM: Linux-4.15.0-51-generic
-- CMAKE_SYSTEM_PROCESSOR: x86_64
-- CMAKE_CXX_COMPILER: /usr/bin/c++
-- Build type not specified: defaulting to Release.
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Looking for pthread_create
-- Looking for pthread_create - not found
```

```
ue@ue-X580VD: ~/Desktop/srsLTE/build
ue@ue-X580VD:~$ cd ~/Desktop/srsLTE/
ue@ue-X580VD:~/Desktop/srsLTE$ mkdir build
ue@ue-X580VD:~/Desktop/srsLTE$ cd build/
ue@ue-X580VD:~/Desktop/srsLTE/build$ cmake ../
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- CMAKE_SYSTEM: Linux-4.15.0-51-generic
-- CMAKE_SYSTEM_PROCESSOR: x86_64
-- CMAKE_CXX_COMPILER: /usr/bin/c++
-- Build type not specified: defaulting to Release.
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Looking for pthread_create
-- Looking for pthread_create - not found
```

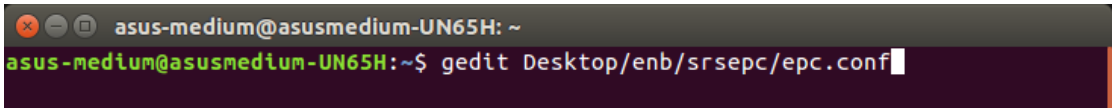
```
ue@ue-X580VD: ~/Desktop/srsLTE/build
ue@ue-X580VD:~/Desktop/srsLTE/build$ sudo make install
[sudo] password for ue:
[  1%] Built target rrc_asn1
[  2%] Built target srslte_asn1
-- Generating build_info.h
[  2%] Built target gen_build_info
[  7%] Built target srslte_common
[  8%] Built target arch_select
[  9%] Built target srslte_enb
[ 10%] Built target srslte_agc
[ 11%] Built target srslte_ch_estimation
[ 12%] Built target srslte_phy_common
[ 17%] Built target srslte_fec
[ 17%] Built target srslte_mimo
[ 22%] Built target srslte_phch
[ 24%] Built target srslte_sync
[ 27%] Built target srslte_utils
[ 28%] Built target srslte_channel
[ 29%] Built target srslte_dft
[ 30%] Built target srslte_io
[ 32%] Built target srslte_modem
[ 33%] Built target srslte_resampling
[ 34%] Built target srslte_scrambling
[ 35%] Built target srslte_ue
[ 35%] Built target srslte_phy
[ 35%] Built target refsignal_ul_test_all
```

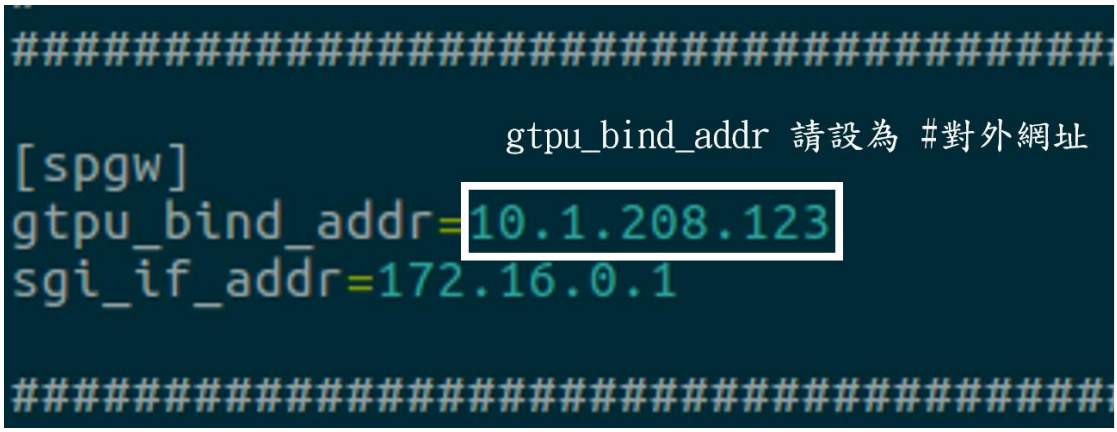
四、 設定 srsLTE

1. 設定 EPC

在 EPC 開一個新的終端機輸入指令

cd ~/path/to/srsLTE/srsepc
gedit epc.conf





2. 設定 eNB

2.1. 修改 conf 檔

在 MeNB 和 SeNB 開啟一個新的終端機輸入指令

gedit /path/to/srsLTE/srsenb/enb.conf

```
[enb]
enb_id = 0x19B
cell_id = 0x01
phy_cell_id = 1
tac = 0x0007
mcc = 001
mnc = 01
mme_addr = 127.0.0.1
gtp_bind_addr = 127.0.0.1
s1c_bind_addr = 127.0.0.1
n_prb = 25
#tm = 4
#nof_ports = 2
```

將這三個addr 都設為 127.0.0.1

```
[expert]
#pdsch_max_its          = 4
#nof_phy_threads        = 2
#pregenerate_signals    = false
#tx_amplitude           = 0.6
#link_failure_nof_err   = 50
rrc_inactivity_timer = 6000000
#max_prach_offset_us    = 30
#enable_mbsfn = false
x2_bind_addr = 192.168.128.101
x2_senb_addr = 192.168.128.100
```

x2_bind_addr
設為自己內網的網址

x2_senb_addr
設為另一個eNB內網的網址

2.2. 修改 lwaap_entity.h 檔案

開啟終端機輸入指令

```
gedit /path/to/srsLTE/lib/include/srslte/upper/lwaap_entity.h
```

```
#define WLAN_IF "eth0"
#define UE_MAC0 0x10
#define UE_MAC1 0x7b
#define UE_MAC2 0x44
#define UE_MAC3 0x23
#define UE_MAC4 0x07
#define UE_MAC5 0xba

#define ETH_TYPE_WIFI    0x9e65
#define LWAAP_HEADER_LEN 1
```

WLAN_IF 設為 eNB 內網
網卡名稱

UE_MAC 請設為UE 內網
網卡Mac_addr

3. 設定 UE

在 UE 開啟終端機並輸入指令

gedit /path/to/srsue/hdr/upper/lwaap.h

4. 重新編譯

在 EPC、MeNB、SeNB、UE 開啟終端機輸入指令

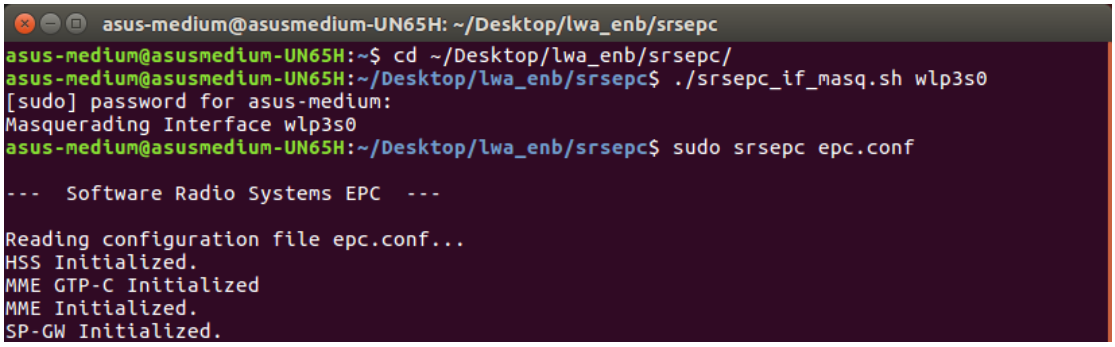
cd /path/to/srsLTE/build
make
sudo make install
sudo ldconfig

五、 執行 srsLTE

1. 執行 EPC

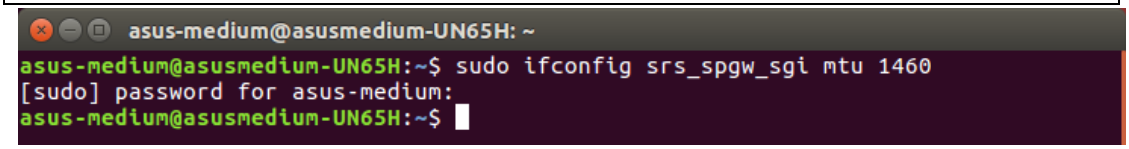
在 EPC 開一個新的終端機輸入指令

cd ~/path/to/srsLTE/srsepc
./srsepc_if_masq.sh enp4s0 #enp4s0 是本例使用的對外網卡名稱
sudo srsepc epc.conf



在 EPC 開一個新的終端機輸入指令

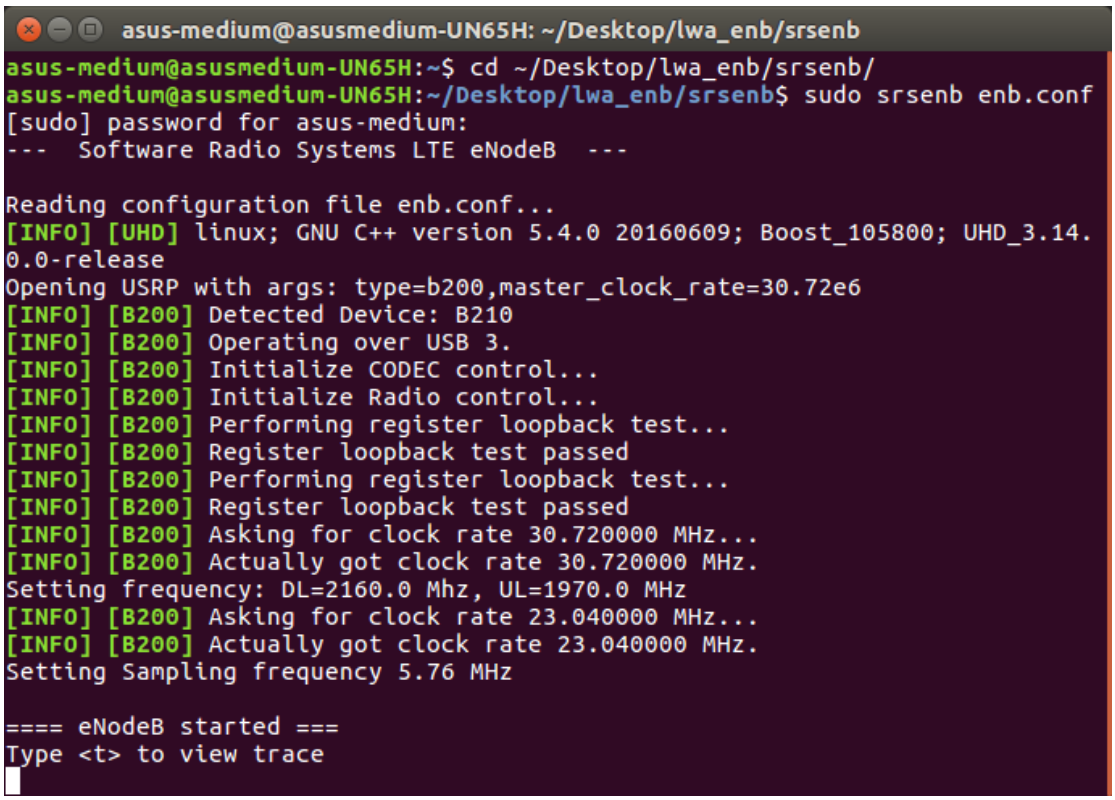
sudo ifconfig srs_spgw_sgi mtu 1460



2. 執行 eNB

在 eNB 再開一個新的終端機輸入

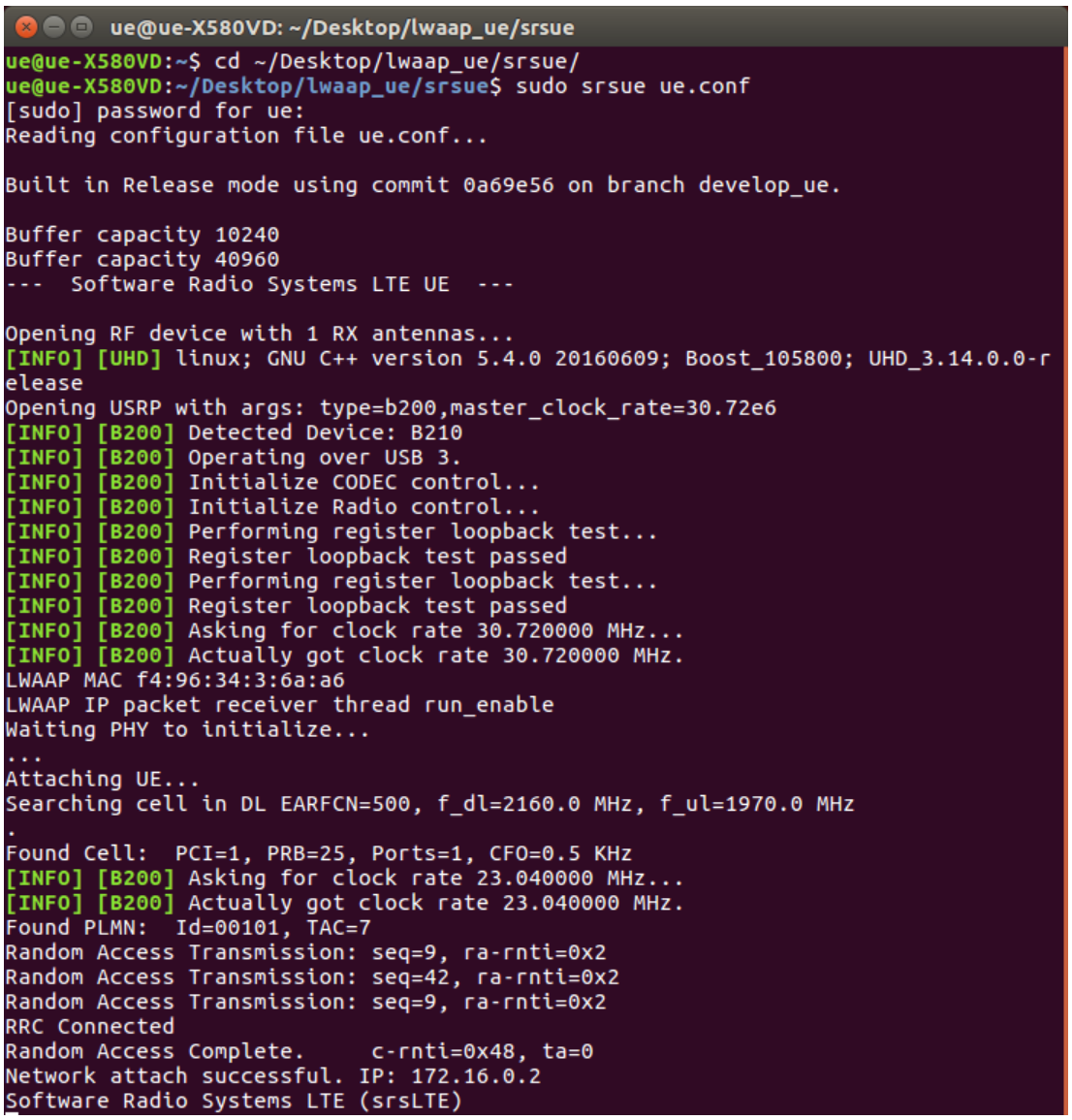
cd ~/path/to/srsLTE/srsenb
sudo srsenb enb.conf

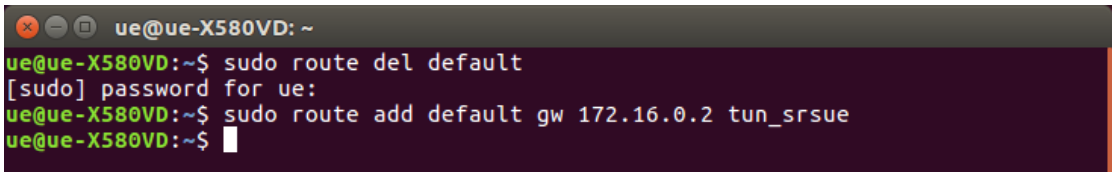


3. 執行 UE

在 UE 開一個新的終端機輸入

cd ~/path/to/srsLTE/srsue
sudo srsue　ue.conf
sudo route del default
sudo route add default gw 172.16.0.2 tun_srsue


ue@ue-X580VD: ~/Desktop/lwaap_ue/srsue
ue@ue-X580VD:~$ cd ~/Desktop/lwaap_ue/srsue/
ue@ue-X580VD:~/Desktop/lwaap_ue/srsue$ sudo srsue ue.conf
[sudo] password for ue:
Reading configuration file ue.conf...
Built in Release mode using commit 0a69e56 on branch develop_ue.
Buffer capacity 10240
Buffer capacity 40960
--- Software Radio Systems LTE UE ---
Opening RF device with 1 RX antennas...
[INFO] [UHD] linux; GNU C++ version 5.4.0 20160609; Boost_105800; UHD_3.14.0.0-release
Opening USRP with args: type=b200,master_clock_rate=30.72e6
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 30.720000 MHz...
[INFO] [B200] Actually got clock rate 30.720000 MHz.
LWAAP MAC f4:96:34:3:6a:a6
LWAAP IP packet receiver thread run_enable
Waiting PHY to initialize...
...
Attaching UE...
Searching cell in DL EARFCN=500, f_dl=2160.0 MHz, f_ul=1970.0 MHz
.
Found Cell: PCI=1, PRB=25, Ports=1, CFO=0.5 KHz
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
Found PLMN: Id=00101, TAC=7
Random Access Transmission: seq=9, ra-rnti=0x2
Random Access Transmission: seq=42, ra-rnti=0x2
Random Access Transmission: seq=9, ra-rnti=0x2
RRC Connected
Random Access Complete.   c-rnti=0x48, ta=0
Network attach successful. IP: 172.16.0.2
Software Radio Systems LTE (srsLTE)


ue@ue-X580VD: ~
ue@ue-X580VD:~$ sudo route del default
[sudo] password for ue:
ue@ue-X580VD:~$ sudo route add default gw 172.16.0.2 tun_srsue
ue@ue-X580VD:~$

六、 測試

1. 互通測試

根據 EPC 設定，EPC 預設 ip 為 172.16.0.1

UE 在終端機輸入

```
ping 172.16.0.1 -c 10
```

若是有收到回覆，則代表平台建置成功。

```
ue@ue-X580VD: ~
ue@ue-X580VD:~$ ping 172.16.0.1 -c 10
PING 172.16.0.1 (172.16.0.1) 56(84) bytes of data.
64 bytes from 172.16.0.1: icmp_seq=1 ttl=64 time=179 ms
64 bytes from 172.16.0.1: icmp_seq=2 ttl=64 time=16.0 ms
64 bytes from 172.16.0.1: icmp_seq=3 ttl=64 time=14.0 ms
64 bytes from 172.16.0.1: icmp_seq=4 ttl=64 time=11.8 ms
64 bytes from 172.16.0.1: icmp_seq=5 ttl=64 time=18.0 ms
64 bytes from 172.16.0.1: icmp_seq=6 ttl=64 time=15.8 ms
64 bytes from 172.16.0.1: icmp_seq=7 ttl=64 time=12.8 ms
64 bytes from 172.16.0.1: icmp_seq=8 ttl=64 time=9.98 ms
64 bytes from 172.16.0.1: icmp_seq=9 ttl=64 time=16.9 ms
64 bytes from 172.16.0.1: icmp_seq=10 ttl=64 time=13.8 ms

--- 172.16.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9012ms
rtt min/avg/max/mdev = 9.983/30.951/179.973/49.728 ms
ue@ue-X580VD:~$
```

2. 調配封包傳送比例

在 MeNB 執行 enb.conf 的終端機輸入”t”並按下”enter”，可以調整比例

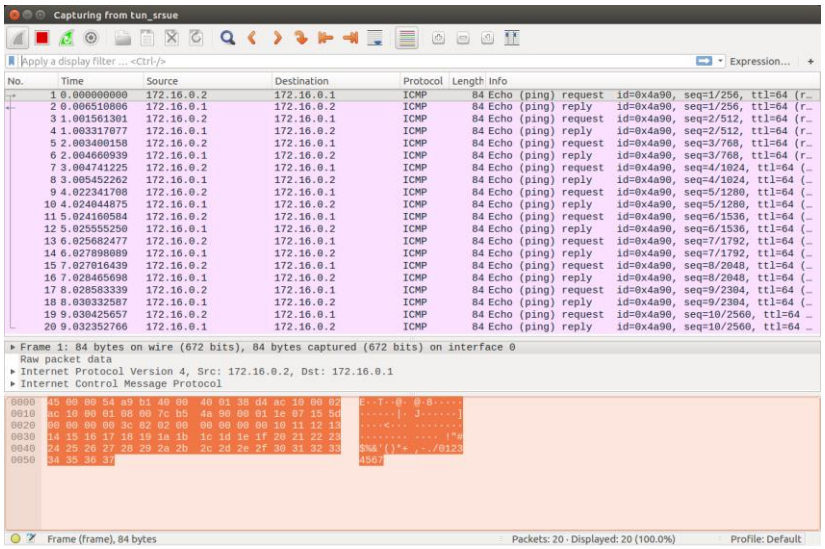
```
asus-medium@asusmedium-UN65H: ~/Desktop/enb/srsenb
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 30.720000 MHz...
[INFO] [B200] Actually got clock rate 30.720000 MHz.
Setting frequency: DL=2160.0 Mhz, UL=1970.0 MHz
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
Setting Sampling frequency 5.76 MHz

==== eNodeB started ===
Type <t> to view trace
RACH:  tti=6101, preamble=0, offset=0, temp_crnti=0x46
Data LCID 3
Data LCID ---------------------------------- 3
LWAAP TX MAC 78:24:af:4:55:3
LWAAP add user rnti=0x46
User 0x46 connected
r
Enter lwa ratio:1 1
```

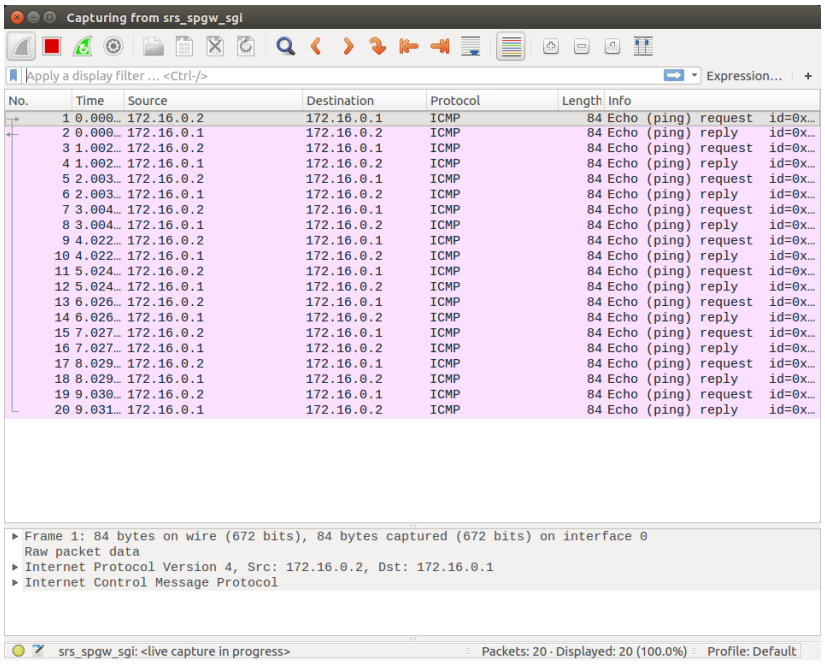
3. Wireshark 介面查看

接著利用 Wireshark 軟體觀看封包傳遞流程。

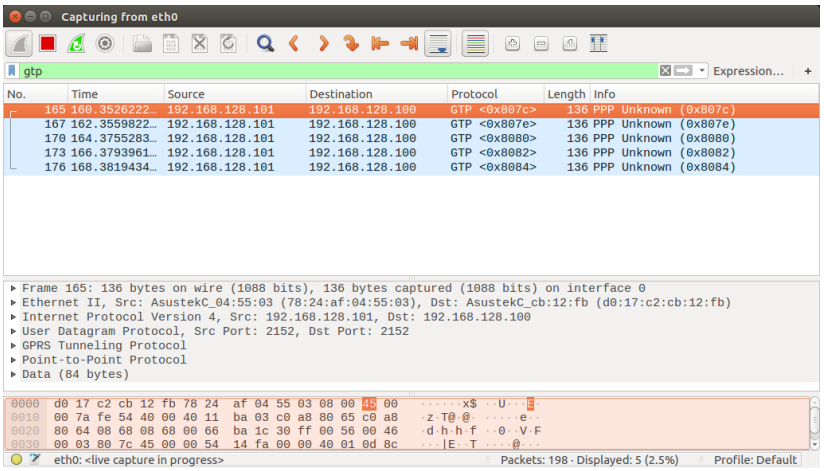
UE 端:



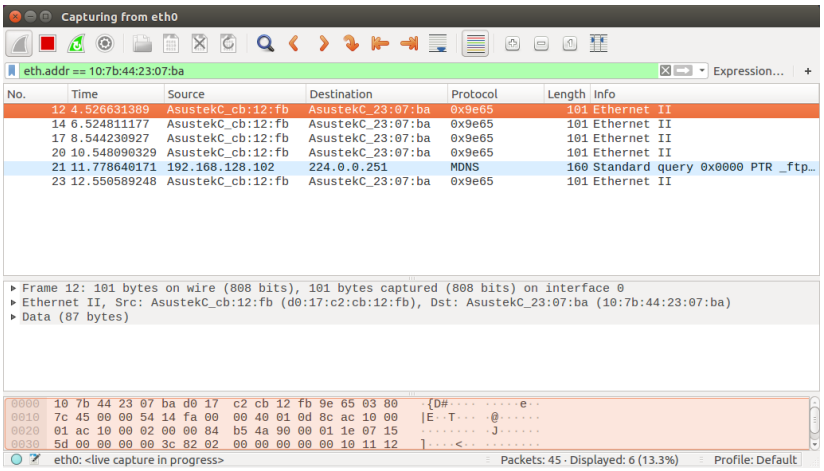
MeNB 端:



SeNB 端收封包:


Capturing from eth0
gtp
Expression...
No. Time Source Destination Protocol Length Info
165 160.3526222… 192.168.128.101 192.168.128.100 GTP <0x807c> 136 PPP Unknown (0x807c)
167 162.3559822… 192.168.128.101 192.168.128.100 GTP <0x807e> 136 PPP Unknown (0x807e)
170 164.3755283… 192.168.128.101 192.168.128.100 GTP <0x8080> 136 PPP Unknown (0x8080)
173 166.3793961… 192.168.128.101 192.168.128.100 GTP <0x8082> 136 PPP Unknown (0x8082)
176 168.3819434… 192.168.128.101 192.168.128.100 GTP <0x8084> 136 PPP Unknown (0x8084)
Frame 165: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface 0
Ethernet II, Src: AsustekC_04:55:03 (78:24:af:04:55:03), Dst: AsustekC_cb:12:fb (d0:17:c2:cb:12:fb)
Internet Protocol Version 4, Src: 192.168.128.101, Dst: 192.168.128.100
User Datagram Protocol, Src Port: 2152, Dst Port: 2152
GPRS Tunneling Protocol
Point-to-Point Protocol
Data (84 bytes)
0000 d0 17 c2 cb 12 fb 78 24 af 04 55 03 08 00 45 00
0010 00 7a fe 54 40 00 40 11 ba 03 c0 a8 80 65 c0 a8
0020 80 64 08 68 08 68 00 66 ba 1c 30 ff 00 56 00 46
0030 00 03 80 7c 45 00 00 54 14 fa 00 00 40 01 0d 8c
eth0: <live capture in progress>
Packets: 198 · Displayed: 5 (2.5%)
Profile: Default

SeNB 端送封包:


Capturing from eth0
eth.addr == 10:7b:44:23:07:ba
Expression...
No. Time Source Destination Protocol Length Info
12 4.526631389 AsustekC_cb:12:fb AsustekC_23:07:ba 0x9e65 101 Ethernet II
14 6.524811177 AsustekC_cb:12:fb AsustekC_23:07:ba 0x9e65 101 Ethernet II
17 8.544230927 AsustekC_cb:12:fb AsustekC_23:07:ba 0x9e65 101 Ethernet II
20 10.548090329 AsustekC_cb:12:fb AsustekC_23:07:ba 0x9e65 101 Ethernet II
21 11.778640171 192.168.128.102 224.0.0.251 MDNS 160 Standard query 0x0000 PTR _ftp…
23 12.550589248 AsustekC_cb:12:fb AsustekC_23:07:ba 0x9e65 101 Ethernet II
Frame 12: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface 0
Ethernet II, Src: AsustekC_cb:12:fb (d0:17:c2:cb:12:fb), Dst: AsustekC_23:07:ba (10:7b:44:23:07:ba)
Data (87 bytes)
0000 10 7b 44 23 07 ba d0 17 c2 cb 12 fb 9e 65 03 80
0010 7c 45 00 00 54 14 fa 00 00 40 01 0d 8c ac 10 00
0020 01 ac 10 00 02 00 00 84 b5 4a 90 00 01 1e 07 15
0030 5d 00 00 00 00 3c 82 02 00 00 00 00 00 10 11 12
eth0: <live capture in progress>
Packets: 45 · Displayed: 6 (13.3%)
Profile: Default