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

課程:5G垂直應用網路

實驗三
URLLC垂直應用網路實驗

副教授：吳俊興

助教：林原進

國立高雄大學 資訊工程學系

Ouline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
  - 調整參數
- 執行程式暨測試
  - 實驗-Split Mode
  - 實驗-Duplication Mode
- 總結

實驗目的

- 建置以MR-DC支援uRLLC的垂直應用網路
- 調整應用及網路參數來分析及量測PRP對網路可靠度及時延的影響

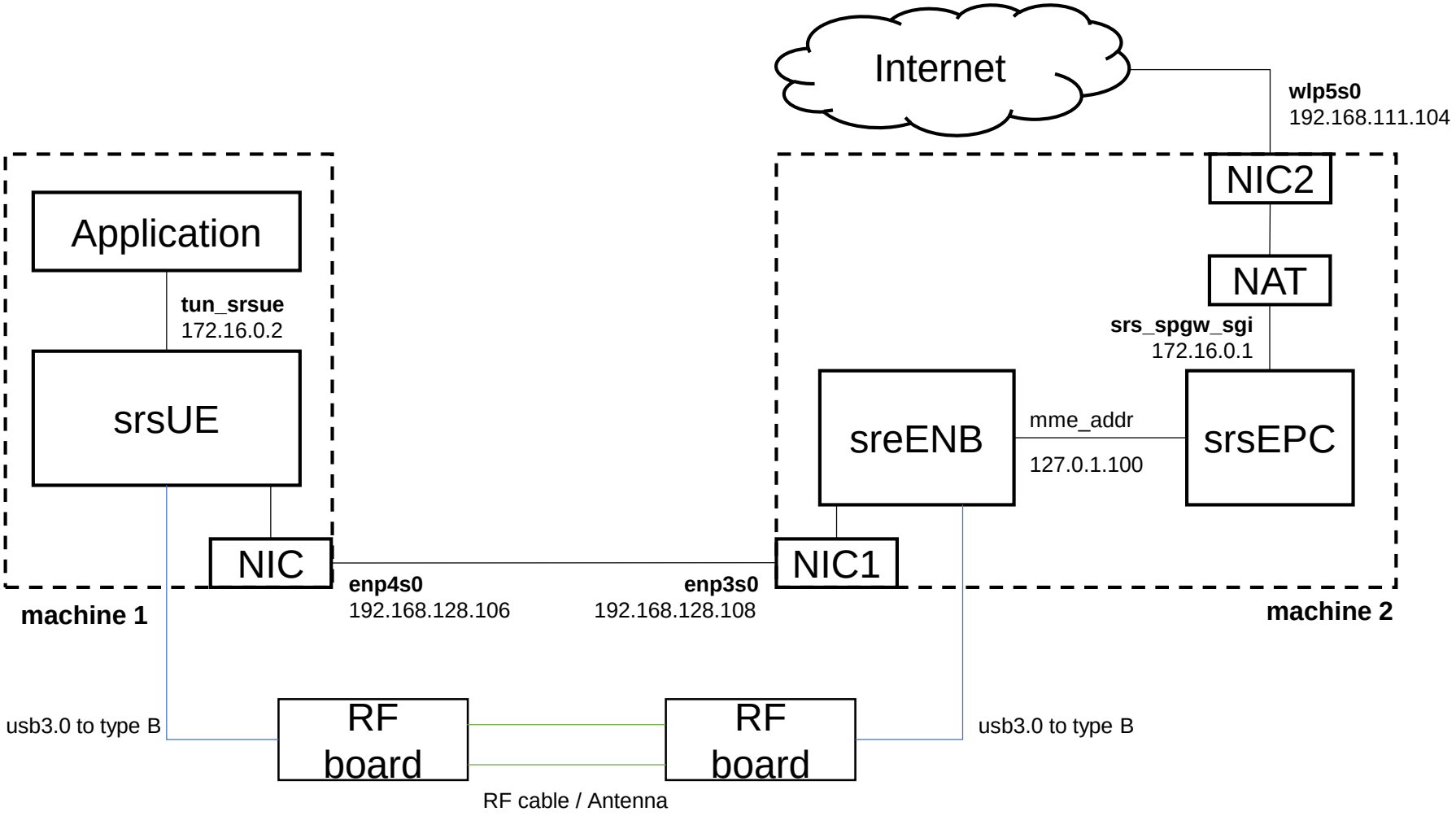
實驗內容

- 利用開源碼小基站建置uRLLC實驗環境
- 調整參數量測不同情境下網路效能傳輸的表現

Ouline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
  - 調整參數
- 執行程式暨測試
  - 實驗-Split Mode
  - 實驗-Duplication Mode
- 總結

srsLTE Small Cell 實驗架構



實驗環境

軟硬體環境 – 硬體

名稱	規格	數量	目的
EPC+eNB	電腦型號 ： ASUS NB M580V	1	啟動 MME,S-GW,P-GW
	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

軟硬體環境 – 軟體

名稱	軟體	版本
EPC+eNB	OS : Ubuntu	Ubuntu 16.04
		Kernel : 4.15.0-041500-lowlatancy
	srsLTE	srsLTE 19.12 d045213fb9cbf98c83c06d7c17197a9dcbfddacf
UE	OS : Ubuntu	Ubuntu 16.04
		Kernel : 4.15.0-041500-lowlatancy
	srsLTE	srsLTE 19.12 d045213fb9cbf98c83c06d7c17197a9dcbfddacf

下載及安裝Kernel

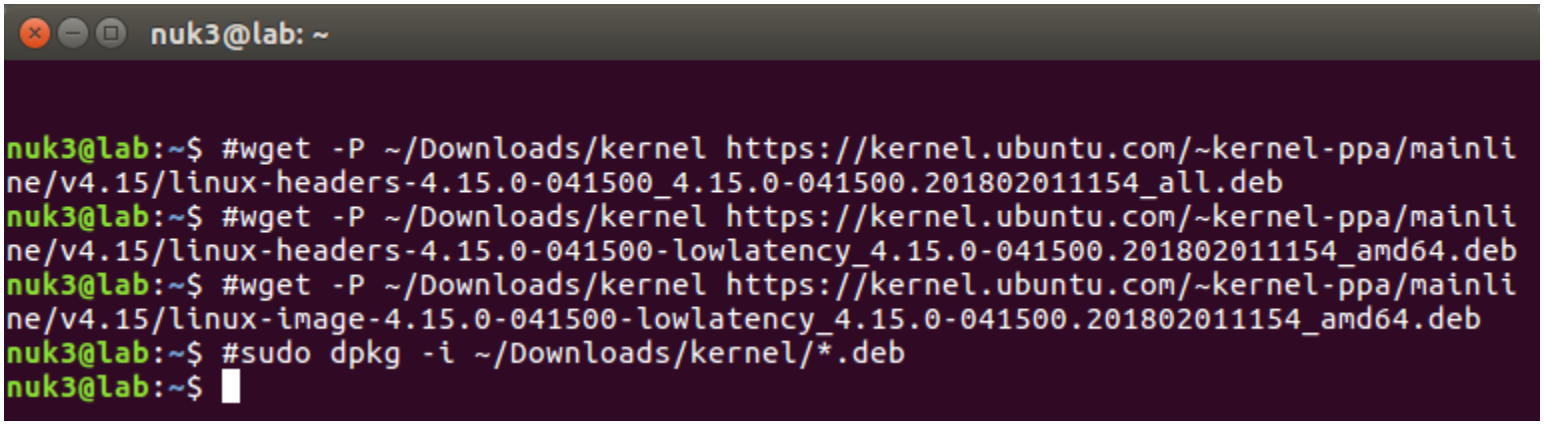
在主機上開啟一個新的終端機，並且輸入

$ wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.15/linux-headers-4.15.0-041500_4.15.0-041500.201802011154_all.deb

$ wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.15/linux-headers-4.15.0-041500-lowlatency_4.15.0-041500.201802011154_amd64.deb

$ wget -P ~/Downloads/kernel https://kernel.ubuntu.com/~kernel-ppa/mainline/v4.15/linux-image-4.15.0-041500-lowlatency_4.15.0-041500.201802011154_amd64.deb

$ sudo dpkg -i ~/Downloads/kernel/*.deb



修改開機選單和設定

在終端機輸入以下指令

$sudo gedit /etc/default/grub

```
nuk3@lab: ~
nuk3@lab:~$ sudo gedit /etc/default/grub
[sudo] password for nuk3:
```

"GRUB_HIDDEN_TIMEOUT=0" 改成
"#GRUB_HIDDEN_TIMEOUT=0"

```
Open ▾   *grub
/etc/default   Save
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
#GRUB_HIDDEN_TIMEOUT=0
#GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=4
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```

更新剛才的設定

在終端機輸入以下指令

$sudo update-grub2

然後終端機輸入以下指令，重啟電腦

$sudo reboot

然後在開機選單選剛才安裝的lowlatency kernel

檢查CPU的效能設定

重新開機後在終端機輸入，確認Kernel版本

$uname -a

```
nuk2@nuk2-X580VD: ~
nuk2@nuk2-X580VD:~$ uname -a
Linux nuk2-X580VD 4.15.0-041500-lowlatency #201802011154 SMP PREEMPT Thu Feb 1 1
1:58:07 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
nuk2@nuk2-X580VD:~$
```

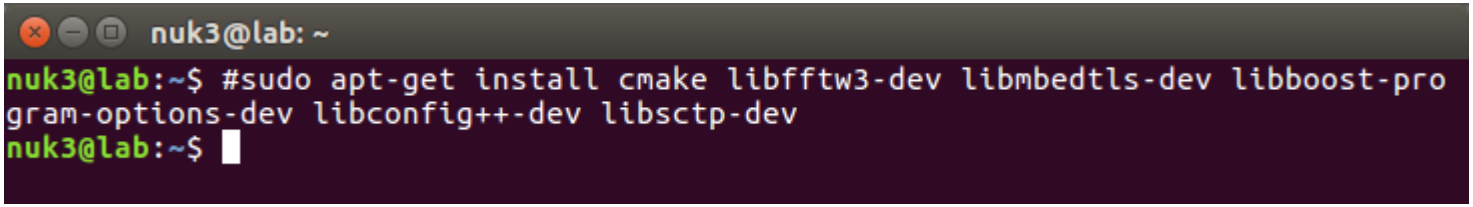
Ouline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
  - 調整參數
- 執行程式暨測試
  - 實驗-Split Mode
  - 實驗-Duplication Mode
- 總結

Required Library

- 在終端機輸入以下指令

$sudo apt-get install cmake libfftw3-dev libmbedtls-dev libboost-program-options-dev libconfig++-dev libsctp-dev



Install UHD

根據所使用的硬體安裝不同的套件

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

ref: http://files.ettus.com/manual/page_install.html

Download and Build

在終端機輸入以下指令

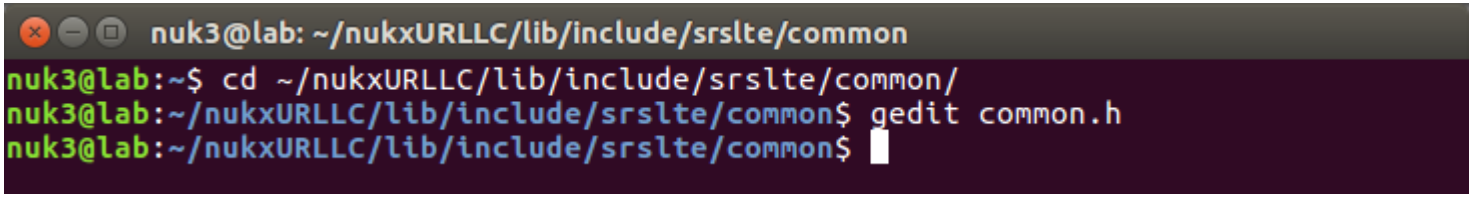
```
$git clone https://github.com/Nukicslab/nukxURLLC.git
$cd nukxURLLC
$mkdir build
$cd build
$cmake ../
$make
$make test
$sudo make install
$srslte_install_configs.sh user
```

```
nuk3@lab: ~/nukxURLLC/build
nuk3@lab:~$ git clone https://github.com/Nukicslab/nukxURLLC.git
Cloning into 'nukxURLLC'...
remote: Enumerating objects: 1026, done.
remote: Counting objects: 100% (1026/1026), done.
remote: Compressing objects: 100% (747/747), done.
remote: Total 1026 (delta 263), reused 1026 (delta 263), pack-reused 0
Receiving objects: 100% (1026/1026), 7.98 MiB | 2.15 MiB/s, done.
Resolving deltas: 100% (263/263), done.
Checking connectivity... done.
nuk3@lab:~$ cd nukxURLLC/
nuk3@lab:~/nukxURLLC$ mkdir build
nuk3@lab:~/nukxURLLC$ cd build/
nuk3@lab:~/nukxURLLC/build$ #cmake ../
nuk3@lab:~/nukxURLLC/build$ #make
nuk3@lab:~/nukxURLLC/build$ #make test
nuk3@lab:~/nukxURLLC/build$ #sudo make install
nuk3@lab:~/nukxURLLC/build$ #srslte_install_configs.sh user
nuk3@lab:~/nukxURLLC/build$
```

Edit File

$cd ~/nukxURLLC/lib/include/srslte/common/

$gedit common.h



Common define - NUK

- 執行eNB、UE的主機：
  - 請把 #define NUK 改成 true

```
/*****************************************************************************
                                INCLUDES
*****************************************************************************/

#include <memory>
#include <stdint.h>
#include <string.h>

/*****************************************************************************
                                DEFINES
*****************************************************************************/
#define NUK true
#define NUK_UE true
#define NUK_JIN_DEBUG false
```

Common define - eNB

執行eNB的主機：

請把 #define NUK_UE 改成 false

common.h (~/nukxURLLC/lib/include/srslte/common) - gedit

Open Save

```
 * and at http://www.gnu.org/licenses/.
 *
 */

#ifndef SRSLTE_COMMON_H
#define SRSLTE_COMMON_H

/*****************************************************************************
                              INCLUDES
*****************************************************************************/

#include <memory>
#include <stdint.h>
#include <string.h>

/*****************************************************************************
                              DEFINES
*****************************************************************************/
#define NUK true
#define NUK_UE false
#define NUK_JIN_DEBUG false
```

Common define - UE

執行UE的主機：

請把 #define NUK_UE 改成 true

common.h (~/nukxURLLC/lib/include/srslte/common) - gedit

Open Save

```
 * and at http://www.gnu.org/licenses/.
 *
 */

#ifndef SRSLTE_COMMON_H
#define SRSLTE_COMMON_H

/*****************************************************************************
                              INCLUDES
*****************************************************************************/

#include <memory>
#include <stdint.h>
#include <string.h>

/*****************************************************************************
                              DEFINES
*****************************************************************************/
#define NUK true
#define NUK_UE true
#define NUK_JIN_DEBUG false
```

Re Compile

完成後，請重新編譯：

$cd ~/nukxURLLC/build

$make

$sudo make install

```
nuk3@lab: ~/nukxURLLC/build
nuk3@lab:~$ cd ~/nukxURLLC/build/
nuk3@lab:~/nukxURLLC/build$ #make
nuk3@lab:~/nukxURLLC/build$ #sudo make install
nuk3@lab:~/nukxURLLC/build$
```

Edit UE proxy address

$cd ~/nukxURLLC/srsue/proxy

$gedit urllc.h

將enb_ip 和ue_ip設定成正確的IP位址後，編譯程式

$gcc urllc.c -lpthread -o proxy

urllc.h (~/nukxURLLC/srsue/proxy) - gedit

Open Save

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/select.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/un.h>
#include <arpa/inet.h>
#include <errno.h>
#include <fcntl.h>
#include <pthread.h>
#include <signal.h>

#define ENB_PORT 8888
#define enb_ip "192.168.128.108"
#define ue_ip "192.168.128.106"
```

Configure File - enb.conf

$cd ~/.config/srslte

- 執行eNB的主機：

  $ gedit enb.conf

  將x2ap_myaddr 改成自己的ip位址

  將x2ap_neiaddr 改成UE的ip位址

  確認 mme,gtp,s1c 位址無誤

```
#######################################################################
[enb]
enb_id = 0x19B
name = MeNB
cell_id = 0x01
phy_cell_id = 1
tac = 0x0007
mcc = 001
mnc = 01
mme_addr = 127.0.1.100
gtp_bind_addr = 127.0.1.1
s1c_bind_addr = 127.0.1.1
n_prb = 50
#tm = 4
#nof_ports = 2
x2ap_myaddr = 192.168.128.108
x2ap_neiaddr = 192.168.128.106
```

Configure File - epc.conf

執行EPC的主機：

$ gedit epc.conf

確認 mme,gtpu 位址無誤

gtpu_bind_addr若是需連到外網，則將此ip位址改成主機連外網的ip位址

```
..
################################
[mme]
mme_code = 0x1a
mme_group = 0x0001
tac = 0x0007
mcc = 001
mnc = 01
mme_bind_addr = 127.0.1.100
apn = srsapn
dns_addr = 8.8.8.8
encryption_algo = EEA0
integrity_algo = EIA1
paging_timer = 2
```

```
########################################

[spgw]
gtpu_bind_addr    = 127.0.1.100
sgi_if_addr       = 172.16.0.1
sgi_if_name       = srs
max_paging_queue = 100
```

Checking IP address

- 請在終端機上輸入

$ifconfig

```
nuk3@lab: ~
nuk3@lab:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 10:7b:44:23:09:42
          inet addr:192.168.128.108  Bcast:192.168.128.255  Mask:255.255.255.0
          inet6 addr: fe80::6547:50d:92d9:4524/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14689 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8785 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10967003 (10.9 MB)  TX bytes:731282 (731.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:4580 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4580 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:661096 (661.0 KB)  TX bytes:661096 (661.0 KB)

nuk3@lab:~$
```

eNB PDU 封包流程

- DownLink : EPC->eNB->UE
  - srsenb::gtpu::handle_gtpu_s1u_rx_packet()
    ->srsenb::pdcp::write_sdu()
    ->srslte::pdcp::write_sdu()
    ->srslte::pdcp_entity_lte::write_sdu()
    ->srsenb::rlc::write_sdu() //在此介面分流

- DownLink : EPC->eNB->UE
  - proxy->srslte::pdcp::run_thread() //接收eNB封包後轉傳到PDCP層
  - srslte::pdcp::write_pdu()
    -> srslte::pdcp_entity_lte::write_pdu()

Ouline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
  - 調整參數
- 執行程式暨測試
  - 實驗-Split Mode
  - 實驗-Duplication Mode
- 總結

enb.conf::enb.n_prb

- srsLTE目前支援：6,15,25,50,75,100
  n_prb對應到的不同頻寬(spectrum bandwidth)

```
# eNB configuration
#
# enb_id:          20-bit eNB identifier.
# cell_id:         8-bit cell identifier.
# tac:             16-bit Tracking Area Code.
# mcc:             Mobile Country Code
# mnc:             Mobile Network Code
# mme_addr:        IP address of MME for S1 connnection
# gtp_bind_addr:   Local IP address to bind for GTP connection
# s1c_bind_addr:   Local IP address to bind for S1AP connection
# n_prb:           Number of Physical Resource Blocks (6,15,25,50,75,100)
```

enb.conf::rf.dl_earfcn

- dl_earfcn對應到MHz (spectrum frequency)
  參考網站: https://www.sqimway.com/lte_band.php
- tx_gain和rx_gain調整接收的放大比率
  越高數值並不代表就會有越高的吞吐量(throughput)

```
# RF configuration
#
# dl_earfcn: EARFCN code for DL
# tx_gain: Transmit gain (dB).
# rx_gain: Optional receive gain (dB). If disabled, AGC if enabled
```

- eNB的dl_earfcn需要跟UE一致
  以避免無法連線情況

Spectrum Frequency

LTE frequency band 36.101 (Rel 16 Jun 2020)

Sort Increasing Display

☑ Earfcn ☐ UMTS band ☑ Geographical area

Band	Name	Downlink (MHz) Low	Downlink (MHz) Middle	Downlink (MHz) High	Bandwidth DL/UL (MHz)	Uplink (MHz) Low	Uplink (MHz) Middle	Uplink (MHz) High	Duplex spacing (MHz)	Geographical area	3GPP release
			Earfcn				Earfcn				
1	2100	2110 0	**2140** **300**	2170 599	60	1920 18000	**1950** **18300**	1980 18599	190	Global	8
2	1900 PCS	1930 600	**1960** **900**	1990 1199	60	1850 18600	**1880** **18900**	1910 19199	80	NAR	8
3	1800+	1805 1200	**1842.5** **1575**	1880 1949	75	1710 19200	**1747.5** **19575**	1785 19949	95	Global	8
4	AWS-1	2110 1950	**2132.5** **2175**	2155 2399	45	1710 19950	**1732.5** **20175**	1755 20399	400	NAR	8
5	850	869 2400	**881.5** **2525**	894 2649	25	824 20400	**836.5** **20525**	849 20649	45	NAR	8
6	UMTS only	875 2650	**880** **2700**	885 2749	10	830 20650	**835** **20700**	840 20749	45	APAC	8
7	2600	2620 2750	**2655** **3100**	2690 3449	70	2500 20750	**2535** **21100**	2570 21449	120	EMEA	8
8	900 GSM	925 3450	**942.5** **3625**	960 3799	35	880 21450	**897.5** **21625**	915 21799	45	Global	8
9	1800	1844.9 3800	**1862.5** **3975**	1879.9 4149	35	1749.9 21800	**1767.5** **21975**	1784.9 22149	95	APAC	8
10	AWS-1+	2110 4150	**2140** **4450**	2170 4749	60	1710 22150	**1740** **22450**	1770 22749	400	NAR	8
		1475.9	**1486**	1495.9		1427.9	**1438**	1447.9			

enb.conf::log

- srsLTE支援不同層log的設定
  可依據需求調整logging level
  方便在log檔案中找出傳遞的資訊

```
# Logging layers: rf, phy, phy_lib, mac, rlc, pdcp, rrc, gtpu, s1ap, all
# Logging levels: debug, info, warning, error, none
```

Ouline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
  - 調整參數
- 執行程式暨測試
  - 實驗-Split Mode
  - 實驗-Duplication Mode
- 總結

UE Attachment

- #若是正確設定 epc.conf 和 enb.conf
  UE 成功連接後會出現以下畫面，
  若是失敗請參閱log file 依據錯誤訊息修正。

```
nuk2@nuk2-X580VD: ~/.config/srslte
[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 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.

[NUK] UE is aggregation node
[NUK] UE start thread
Waiting PHY to initialize ... done!
Attaching UE...
Searching cell in DL EARFCN=300, f_dl=2140.0 MHz, f_ul=1950.0 MHz
.
Found Cell:  Mode=FDD, PCI=1, PRB=25, Ports=1, CFO=-0.0 KHz
Found PLMN:  Id=00101, TAC=7
Could not find Home PLMN Id=00000, trying to connect to PLMN Id=00101
Random Access Transmission: seq=31, ra-rnti=0x2
Random Access Complete.     c-rnti=0x46, ta=1
RRC Connected
Network attach successful. IP: 172.16.0.2
Software Radio Systems LTE (srsLTE)
```

傳輸連結測試

執行UE的主機：

$ ping “enb的ip位址”

```
nuk2@nuk2-X580VD: ~
nuk2@nuk2-X580VD:~$ ping 192.168.128.108 -c 4
PING 192.168.128.108 (192.168.128.108) 56(84) bytes of data.
64 bytes from 192.168.128.108: icmp_seq=1 ttl=64 time=0.202 ms
64 bytes from 192.168.128.108: icmp_seq=2 ttl=64 time=0.134 ms
64 bytes from 192.168.128.108: icmp_seq=3 ttl=64 time=0.143 ms
64 bytes from 192.168.128.108: icmp_seq=4 ttl=64 time=0.128 ms

--- 192.168.128.108 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3090ms
rtt min/avg/max/mdev = 0.128/0.151/0.202/0.033 ms
nuk2@nuk2-X580VD:~$
```

ICMP回應正常，即代表設定無誤且通道順暢

Ping指令測試

執行UE的主機：

$ping 172.16.0.1

```
nuk2@nuk2-X580VD: ~
nuk2@nuk2-X580VD:~$ ping 172.16.0.1 -c 5
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=31.9 ms
64 bytes from 172.16.0.1: icmp_seq=2 ttl=64 time=31.1 ms
64 bytes from 172.16.0.1: icmp_seq=3 ttl=64 time=30.8 ms
64 bytes from 172.16.0.1: icmp_seq=4 ttl=64 time=30.1 ms
64 bytes from 172.16.0.1: icmp_seq=5 ttl=64 time=29.7 ms

--- 172.16.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 29.782/30.759/31.931/0.778 ms
nuk2@nuk2-X580VD:~$
```

ICMP回應正常，即代表設定無誤且順利連線到EPC

開啟 proxy程式

UE在終端機輸入:

$ cd ~/path/to/nukxURLLC/srsue/proxy
$ gcc urllc.c -lpthread -o proxy
$ ./proxy

```
nuk3@lab: ~/nukxURLLC/srsue/proxy
nuk3@lab:~$ cd ~/nukxURLLC/srsue/proxy/
nuk3@lab:~/nukxURLLC/srsue/proxy$ gcc urllc.c -lpthread -o proxy
nuk3@lab:~/nukxURLLC/srsue/proxy$ #./proxy
nuk3@lab:~/nukxURLLC/srsue/proxy$
```

此程式用來接收eNB轉傳的封包，
再將原本的內容轉至srsue中

Ouline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
  - 調整參數
- 執行程式暨測試
  - 實驗-Split Mode
  - 實驗-Duplication Mode
- 總結

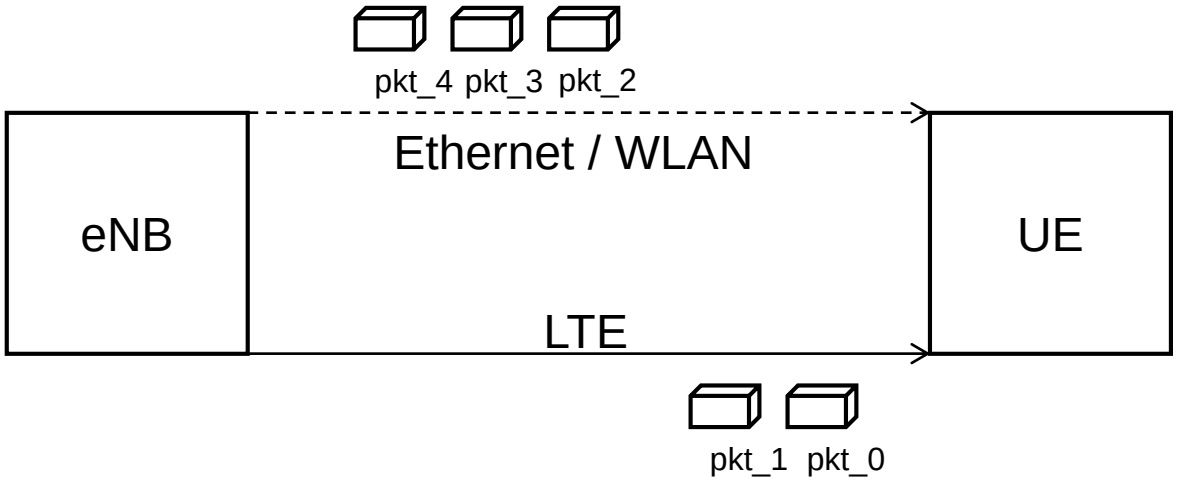
Split_mode

- 在執行eNB的主機上：
  - 輸入 ‘s’，按下’enter’，將會啟動split_mode

```
-- Actually got clock rate 23.040000 MHz.
-- Performing timer loopback test... pass
-- Performing timer loopback test... pass
Setting frequency: DL=2140.0 Mhz, UL=1950.0 MHz
Setting Sampling frequency 1.92 MHz

==== eNodeB started ===
Type <t> to view trace
RACH:  tti=7221, preamble=47, offset=1, temp_crnti=0x46
User 0x46 connected
s
[NUK] Split_mode is ON
[NUK] set ratio 2(M) : 3(S)
```

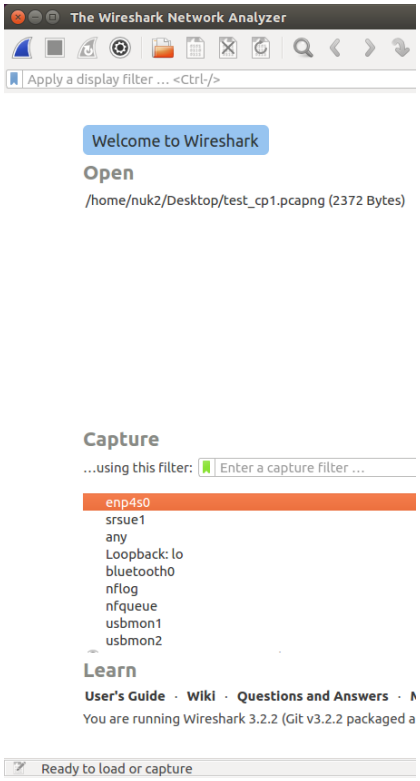
Split_ratio

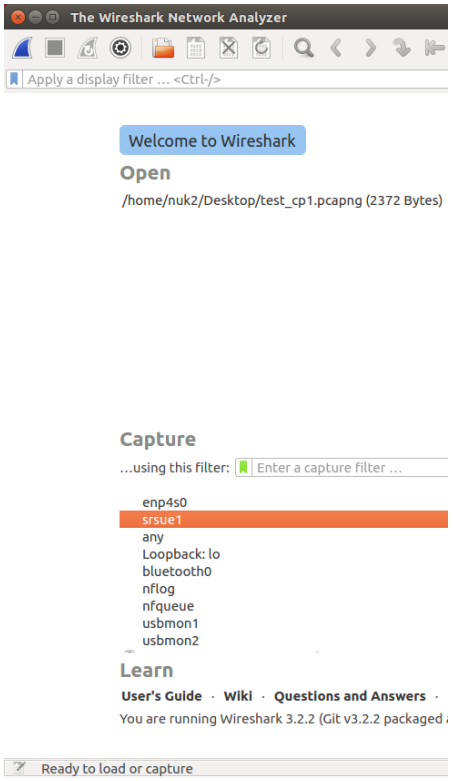
- 分流比例為2:3，如下圖所示



Wireshark

- 在執行UE主機上，開啟兩個Wireshark視窗
  - 其一選取srsue interface
  - 另一選取連結switch網卡介面





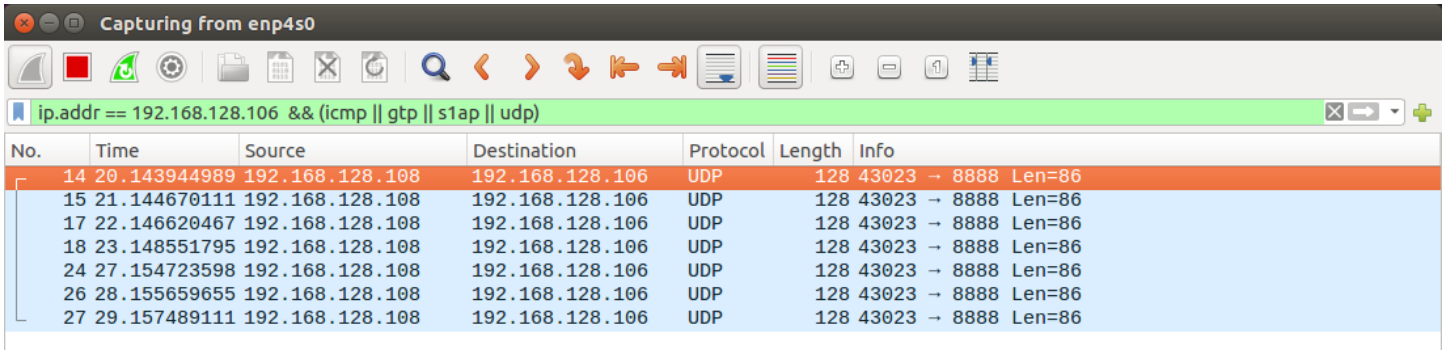
Monitor Seq.

在執行epc主機上:

$ ping 172.16.0.2 -c 10

從UE的wireshark中看到

UE在網卡上收到來自enb的UDP封包，
內含從EPC發出的ICMP request，
接著透過proxy程式轉送給srs介面



No.	Time	Source	Destination	Protocol	Length	Info
14	20.143944989	192.168.128.108	192.168.128.106	UDP	128	43023 → 8888 Len=86
15	21.144670111	192.168.128.108	192.168.128.106	UDP	128	43023 → 8888 Len=86
17	22.146620467	192.168.128.108	192.168.128.106	UDP	128	43023 → 8888 Len=86
18	23.148551795	192.168.128.108	192.168.128.106	UDP	128	43023 → 8888 Len=86
24	27.154723598	192.168.128.108	192.168.128.106	UDP	128	43023 → 8888 Len=86
26	28.155659655	192.168.128.108	192.168.128.106	UDP	128	43023 → 8888 Len=86
27	29.157489111	192.168.128.108	192.168.128.106	UDP	128	43023 → 8888 Len=86

Monitor Seq.(Cont.)

從UE的wireshark中看到：
UE在srs裡解析封包後，
再透過srs介面將ICMP reply回傳至EPC

Capturing from srsue1

Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.16.0.1	172.16.0.2	ICMP	84	Echo (ping) request id=0x2525, seq=1/256, ttl=64 (reply in 2)
2	0.000024793	172.16.0.2	172.16.0.1	ICMP	84	Echo (ping) reply id=0x2525, seq=1/256, ttl=64 (request in 1)
3	1.000656027	172.16.0.1	172.16.0.2	ICMP	84	Echo (ping) request id=0x2525, seq=2/512, ttl=64 (reply in 4)
4	1.000666952	172.16.0.2	172.16.0.1	ICMP	84	Echo (ping) reply id=0x2525, seq=2/512, ttl=64 (request in 3)
5	2.002638552	172.16.0.1	172.16.0.2	ICMP	84	Echo (ping) request id=0x2525, seq=3/768, ttl=64 (reply in 6)
6	2.002649596	172.16.0.2	172.16.0.1	ICMP	84	Echo (ping) reply id=0x2525, seq=3/768, ttl=64 (request in 5)
7	3.004573293	172.16.0.1	172.16.0.2	ICMP	84	Echo (ping) request id=0x2525, seq=4/1024, ttl=64 (reply in 8)
8	3.004606646	172.16.0.2	172.16.0.1	ICMP	84	Echo (ping) reply id=0x2525, seq=4/1024, ttl=64 (request in 7)
9	4.009615132	172.16.0.1	172.16.0.2	ICMP	84	Echo (ping) request id=0x2525, seq=5/1280, ttl=64 (reply in 10)
10	4.009629469	172.16.0.2	172.16.0.1	ICMP	84	Echo (ping) reply id=0x2525, seq=5/1280, ttl=64 (request in 9)
11	5.011869391	172.16.0.1	172.16.0.2	ICMP	84	Echo (ping) request id=0x2525, seq=6/1536, ttl=64 (reply in 12)
12	5.011884555	172.16.0.2	172.16.0.1	ICMP	84	Echo (ping) reply id=0x2525, seq=6/1536, ttl=64 (request in 11)
13	6.012793634	172.16.0.1	172.16.0.2	ICMP	84	Echo (ping) request id=0x2525, seq=7/1792, ttl=64 (reply in 14)
14	6.012807897	172.16.0.2	172.16.0.1	ICMP	84	Echo (ping) reply id=0x2525, seq=7/1792, ttl=64 (request in 13)
15	7.010716699	172.16.0.1	172.16.0.2	ICMP	84	Echo (ping) request id=0x2525, seq=8/2048, ttl=64 (reply in 16)
16	7.010727143	172.16.0.2	172.16.0.1	ICMP	84	Echo (ping) reply id=0x2525, seq=8/2048, ttl=64 (request in 15)
17	8.011878854	172.16.0.1	172.16.0.2	ICMP	84	Echo (ping) request id=0x2525, seq=9/2304, ttl=64 (reply in 18)
18	8.011890780	172.16.0.2	172.16.0.1	ICMP	84	Echo (ping) reply id=0x2525, seq=9/2304, ttl=64 (request in 17)
19	9.013519434	172.16.0.1	172.16.0.2	ICMP	84	Echo (ping) request id=0x2525, seq=10/2560, ttl=64 (reply in 20)
20	9.013529537	172.16.0.2	172.16.0.1	ICMP	84	Echo (ping) reply id=0x2525, seq=10/2560, ttl=64 (request in 19)

UE Aggregate

```
nuk2@nuk2-X580VD: ~/.config/srslte
RRC Connected
Received RRC Connection Release
RRC IDLE
S-TMSI match in paging message
Random Access Transmission: seq=31, ra-rnti=0x2
Random Access Complete.     c-rnti=0x49, ta=1
RRC Connected
[NUK] Received packets
[NUK] UE1 aggregate pdu : 0
[NUK] Received packets
[NUK] UE1 aggregate pdu : 1
[NUK] Received packets
[NUK] UE1 aggregate pdu : 2
[NUK] Received packets
[NUK] UE1 aggregate pdu : 3
[NUK] Received packets
[NUK] UE1 aggregate pdu : 4
[NUK] Received packets
[NUK] UE1 aggregate pdu : 5
[NUK] Received packets
[NUK] UE1 aggregate pdu : 6
```

srsue終端機顯示，收到從proxy轉傳的封包

UE Proxy

```
nuk2@nuk2-X580VD: ~/Desktop/test
nuk2@nuk2-X580VD:~/Desktop/test$ ./urllc
Please enter Ctrl + C to terminate process

receive data len: 86
OK

receive data len: 86
OK

receive data len: 86
OK

receive data len: 86
OK

receive data len: 86
OK
```

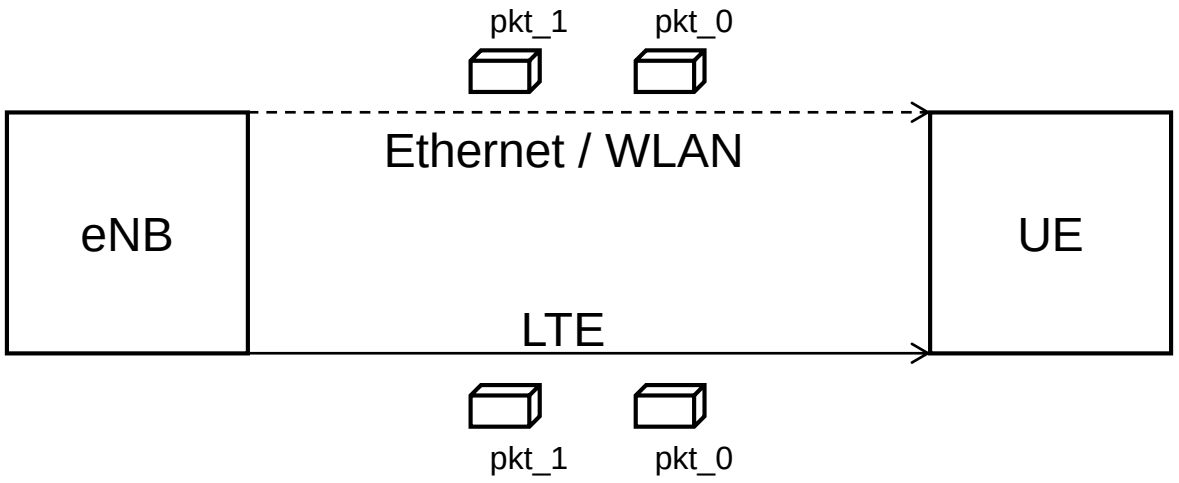
proxy終端機顯示，收到從eNB轉傳的封包

Ouline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
  - 調整參數
- 執行程式暨測試
  - 實驗-Split Mode
  - 實驗-Duplication Mode
- 總結

Duplication_mode

- 在執行eNB的主機上：
  - 輸入 ‘d’，按下’enter’，將會啟動duplication_mode
- Duplication mode 和 Split mode 並不會同時存在



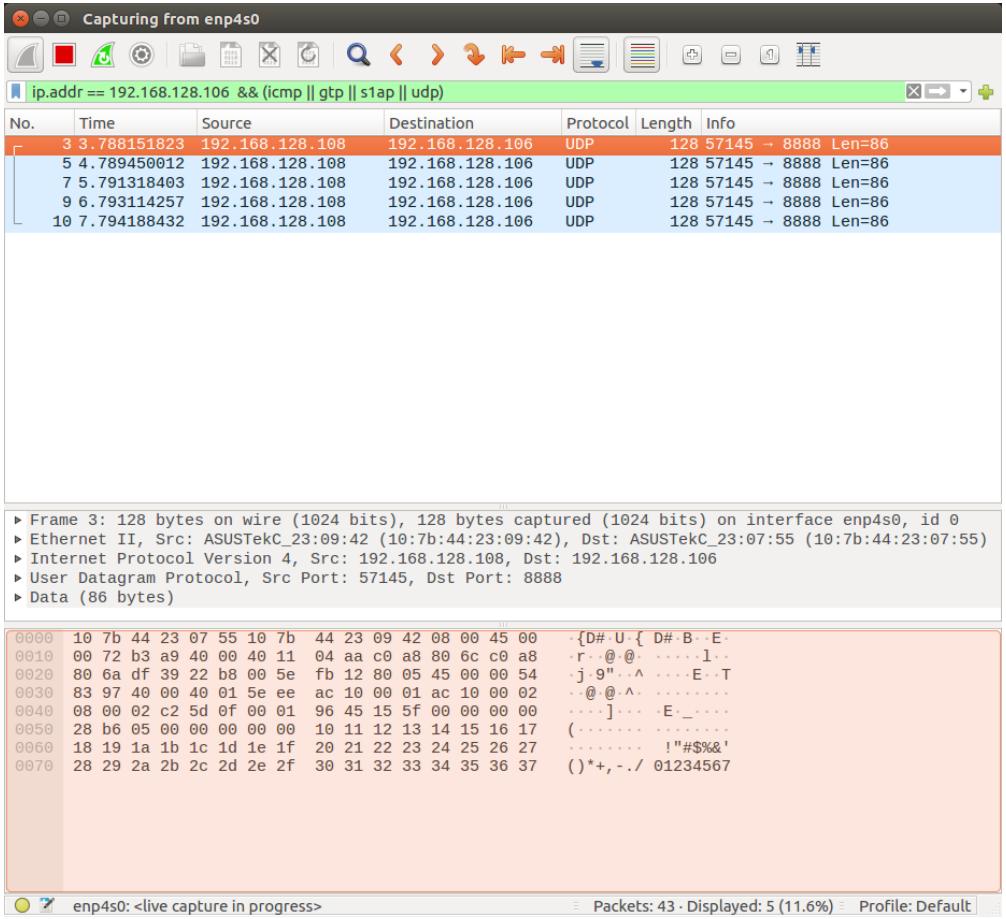
```
nuk3@lab: ~
Reading configuration file /home/nuk3/.config/srslte/enb.conf...
Opening 1 RF devices with 1 RF channels...
Opening USRP with args: type=b200,master_clock_rate=23.04e6
-- Detected Device: B210
-- Operating over USB 3.
-- Initialize CODEC control...
-- Initialize Radio control...
-- Performing register loopback test... pass
-- Performing register loopback test... pass
-- Performing CODEC loopback test... pass
-- Performing CODEC loopback test... pass
-- Asking for clock rate 23.040000 MHz...
-- Actually got clock rate 23.040000 MHz.
-- Performing timer loopback test... pass
-- Performing timer loopback test... pass
Setting frequency: DL=2140.0 Mhz, UL=1950.0 MHz
Setting Sampling frequency 5.76 MHz
[NUK] This eNB is MeNB

==== eNodeB started ===
Type <t> to view trace
d
[NUK] Duplication_mode is ON
```

開啟後，終端機畫面會顯示狀態

Wireshark - Switch

- 在執行epc主機上:
  $ ping 172.16.0.2 -c 10

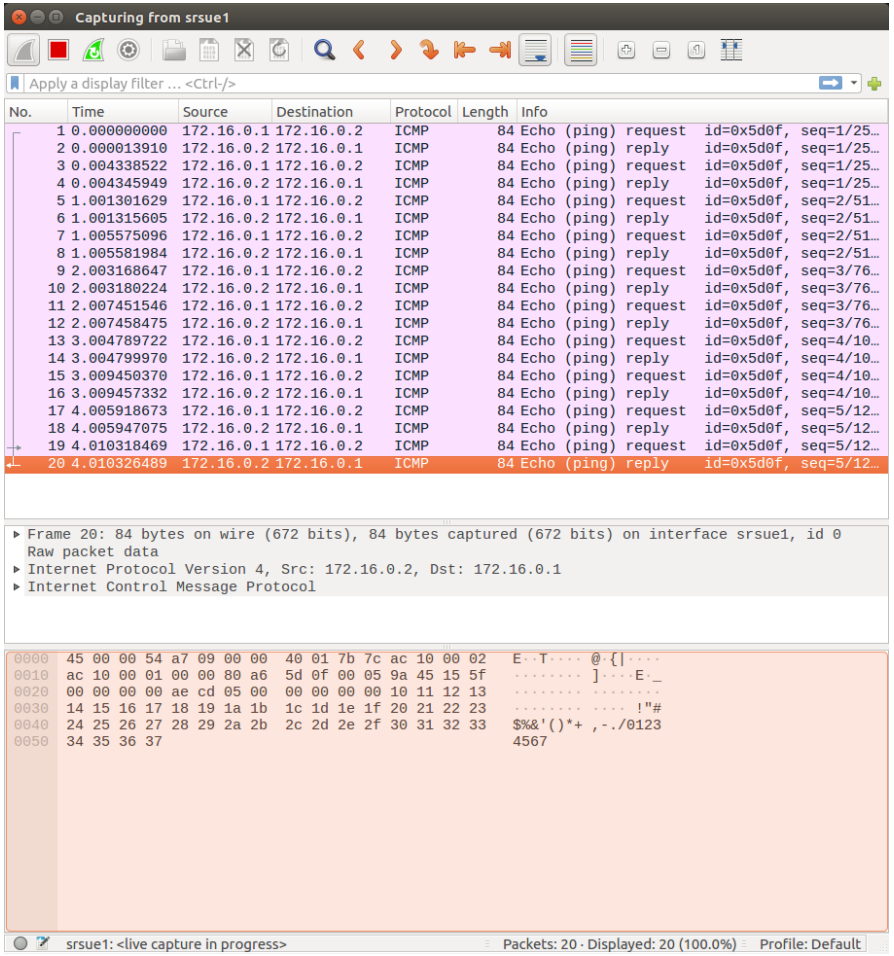
從UE的wireshark中看到：
icmp封包透過Switch封裝成
UDP封包傳向UE



Wireshark - srs

- 在執行epc主機上:
  $ ping 172.16.0.2 -c 10

從UE的wireshark中看到：
icmp封包透過RF路徑傳向UE
同時聚合不同路徑的icmp封包



Ouline

- 實驗目的及實驗內容
- srsLTE 實驗環境
  - srsLTE Small Cell 架構
  - 軟硬體環境
- srsLTE 網路實驗平台建置
  - 安裝所需套件
  - 安裝 srsLTE 網路環境
  - 調整參數
- 執行程式暨測試
  - 實驗-Split Mode
  - 實驗-Duplication Mode
- 總結

Summary

- 在兩台主機間安裝與設定srsLTE
  - 熟悉srsLTE參數配置
  - 觀察srsenb 與 srsue間的封包傳遞過程
- 以Dual Connectivity(DC)為基礎架構，配合多種模式
  - 探討分流或是重複模式對於效能的影響
  - 觀察不同情境下流量的變化