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

實驗二
eMBB垂直應用網路實驗

副教授：吳俊興

助教：胡詠翔

國立高雄大學 資訊工程學系

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 環境設定
- Stage 2. eMBMS 啟動與測試
- Stage 3. eMBMS 觀測
- Stage 4. Physical Layer參數調整
- Stage 5. eMBMS應用
- 總結及問題

實驗目的

- 建置行動寬頻暨MBMS服務應用網路
- 調整網路架構及參數來分析及量測其對網路傳輸效能的影響

實驗內容

- 了解srsLTE在如何進行eMBMS的設定
- 對eMBMS的封包進行觀測
  - 觀測SGI_MB上的封包
  - 觀測M1上的封包
- 透過UE MAC PDU進一步了解eMBMS的資源分配
  - 了解Transport Channel MCH的排程機制
  - 了解Logical Channel MCCH及MTCH的排程機制
  - 了解MIB及SIB1、SIB2及SIB13的意義
- 透過實際調整SIB參數了解其對上層通道的影響
- 透過影片Multicast串流了解eMBMS的應用

Outline

- 實驗目的及實驗內容
- 背景知識
  - srsLTE與TS23.246的差異
- 實驗環境
- Stage 1. 環境設定
- Stage 2. eMBMS 啟動與測試
- Stage 3. eMBMS 觀測
- Stage 4. Physical Layer參數調整
- Stage 5. eMBMS應用
- 總結及問題

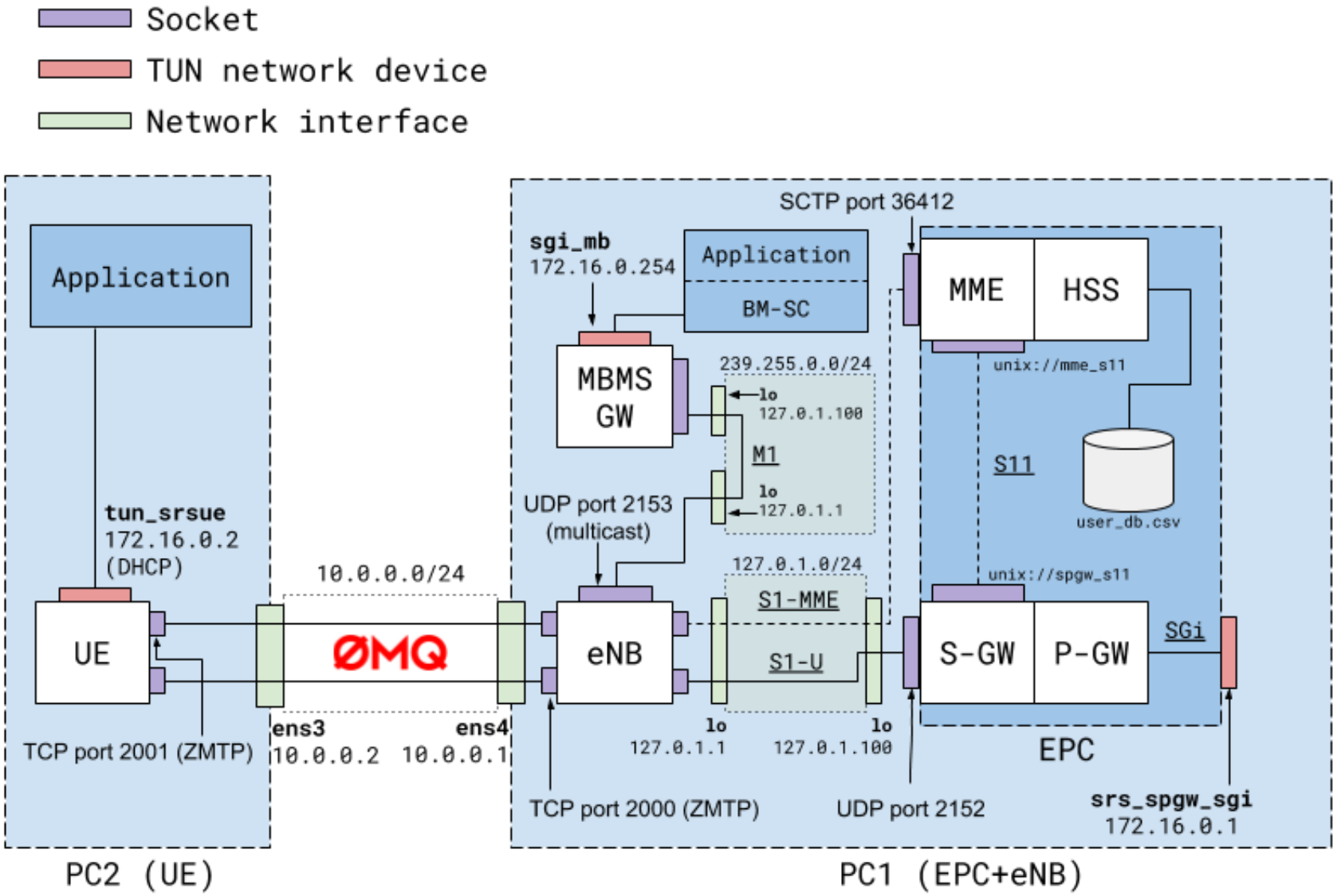
背景知識 - srsLTE與TS23.246的差異

srsLTE的eMBMS	TS23.246的eMBMS
僅有User Plane	有Control Plane和User Plane
eNB透過設定檔資訊加入MBMS-GW的Multicast Group	eNB透過M3得知MBMS-GW的Multicast Group
UE透過設定檔得知MBMS服務的相關資訊	UE透過MBMS的Broadcast Mode或是URL得知服務資訊
eNB會一直配置資源給MCH	在BM-S請求Session Start後eNB才會配置MCH資源
Session在MBMS-GW啟動後便自動開始	Session在BM-SC請求後才會開始

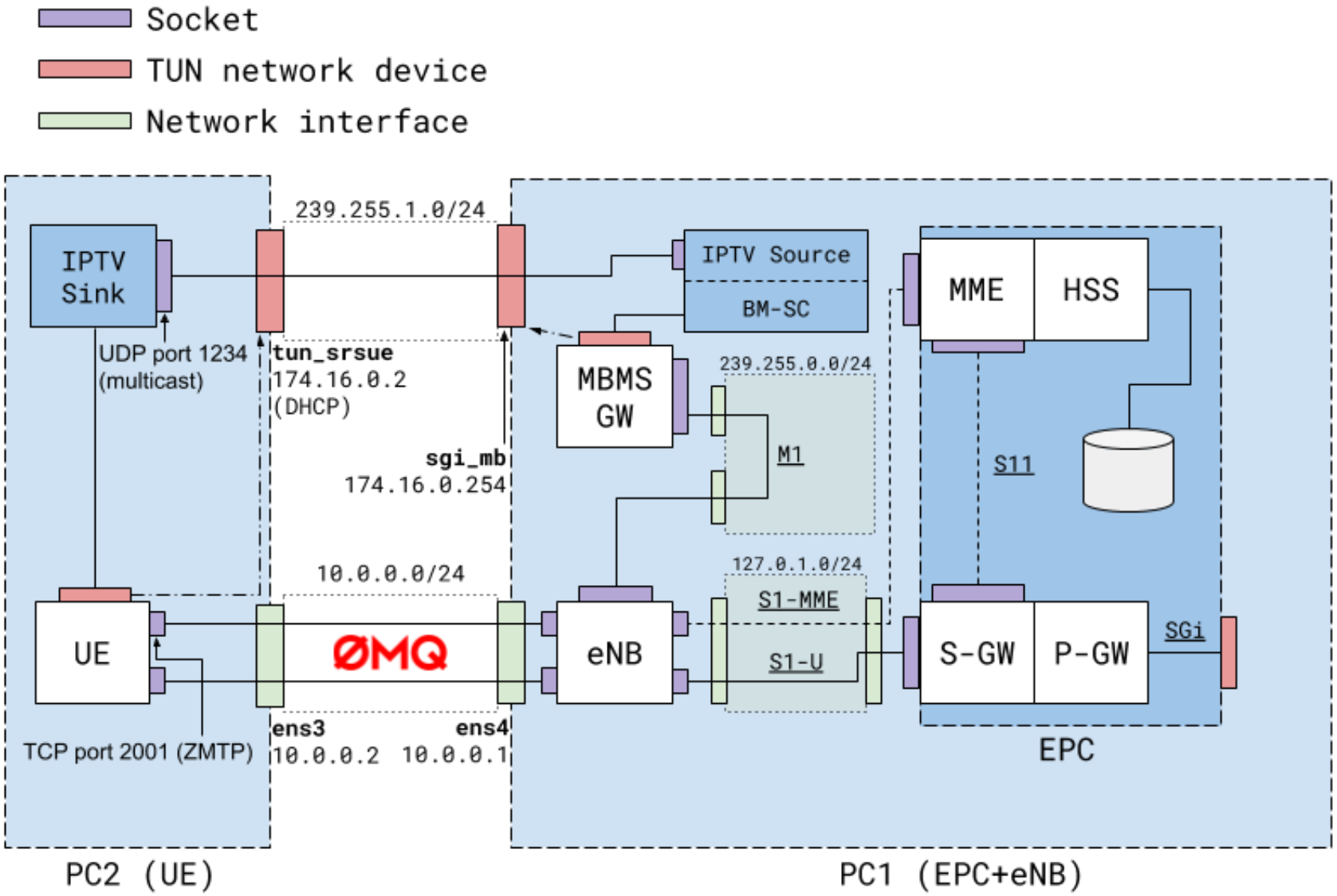
Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
  - 底層架構
  - 應用架構
  - 硬體環境需求
  - 軟體環境需求
- Stage 1. 環境設定
- Stage 2. eMBMS 啟動與測試
- Stage 3. eMBMS 觀測
- Stage 4. Physical Layer參數調整
- Stage 5. eMBMS應用
- 總結及問題

底層架構


Socket
TUN network device
Network interface
Application
tun_srsue
172.16.0.2
(DHCP)
UE
TCP port 2001 (ZMTP)
PC2 (UE)
10.0.0.0/24
ØMQ
ens3
10.0.0.2
ens4
10.0.0.1
SCTP port 36412
sgi_mb
172.16.0.254
Application
BM-SC
MBMS
GW
239.255.0.0/24
lo
127.0.1.100
M1
lo
127.0.1.1
UDP port 2153
(multicast)
127.0.1.0/24
S1-MME
S1-U
eNB
lo
127.0.1.1
lo
127.0.1.100
TCP port 2000 (ZMTP)
UDP port 2152
MME
HSS
unix://mme_s11
S11
user_db.csv
unix://spgw_s11
S-GW
P-GW
SGi
EPC
srs_spgw_sgi
172.16.0.1
PC1 (EPC+eNB)

應用架構


Socket
TUN network device
Network interface
239.255.1.0/24
IPTV Sink
UDP port 1234
(multicast)
tun_srsue
174.16.0.2
(DHCP)
sgi_mb
174.16.0.254
IPTV Source
BM-SC
MBMS GW
239.255.0.0/24
M1
MME
HSS
S11
10.0.0.0/24
127.0.1.0/24
S1-MME
S1-U
UE
ØMQ
eNB
S-GW
P-GW
SGi
TCP port 2001 (ZMTP)
ens3
10.0.0.2
ens4
10.0.0.1
EPC
PC2 (UE)
PC1 (EPC+eNB)

軟硬體環境–硬體

*eNB與UE以乙太網路相連接

名稱	規格	數量	目的
EPC+ eNB	電腦型號：ASUS VivoMini UN65H	1	啟動 MME,HSS,S-GW,P-GW,eNB
	乙太網路卡	2	讓 eNB 透過 ZeroMQ 與 UE 溝通
UE	電腦型號：ASUS NB M580V	1	模擬 UE
	乙太網路卡	1	讓 UE 透過 ZeroMQ 與 eNB 溝通

軟硬體環境–軟體

名稱	軟體	版本
EPC+ eNB	OS : Ubuntu	Ubuntu 20.04
	srsLTE	srsLTE 20.04.1 c892ae56be5302eaee5ca00e270efc7a5ce6fbb2
UE	OS : Ubuntu	Ubuntu 20.04
	srsLTE	srsLTE 20.04.1 c892ae56be5302eaee5ca00e270efc7a5ce6fbb2
影片檔	-	任何格式的影片檔均可，長度最好大於10分鐘

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 環境設定
  - Step1 eNB 設定
  - Step2 MBMS-GW 設定
  - Step3 UE 設定
  - Step4 安裝測試軟體
  - Step5 安裝Wireshark
- Stage 2. eMBMS 啟動與測試
- Stage 3. eMBMS 觀測
- Stage 4. Physical Layer參數調整
- Stage 5. eMBMS應用
- 總結及問題

Step1-1 eNB設定(enb_files)

在eNB的終端機輸入指令vim /etc/srslte/enb.conf 編輯eNB的相關設定，並將[enb_files]下的SIB設定檔參數sib_config改為/etc/srslte/sib.conf.mbsfn

```
#####################################################################
# eNB configuration files
#
# sib_config:  SIB1, SIB2 and SIB3 configuration file
# note: when enabling mbms, use the sib.conf.mbsfn configuration file which incl
udes SIB13
# rr_config:   Radio Resources configuration file
# drb_config:  DRB configuration file
#####################################################################
[enb_files]
sib_config = /etc/srslte/sib.conf.mbsfn
rr_config  = rr.conf
drb_config = drb.conf
```

Step1-2 eNB設定(scheduler)

將eNB設定檔/etc/srslte/enb.conf 內有關無線電資源排程的設定參數，亦即[scheduler]下的參數改為下圖所示之內容

```
#####################################################################
# Scheduler configuration options
#
# max_aggr_level:    Optional maximum aggregation level index (l=log2(L) can be 0, 1, 2 or 3)
# pdsch_mcs:         Optional fixed PDSCH MCS (ignores reported CQIs if specified)
# pdsch_max_mcs:     Optional PDSCH MCS limit
# pusch_mcs:         Optional fixed PUSCH MCS (ignores reported CQIs if specified)
# pusch_max_mcs:     Optional PUSCH MCS limit
# min_nof_ctrl_symbols: Minimum number of control symbols
# max_nof_ctrl_symbols: Maximum number of control symbols
#
#####################################################################
[scheduler]
#max_aggr_level   = -1
#pdsch_mcs        = -1
#pdsch_max_mcs    = -1
#pusch_mcs        = -1
pusch_max_mcs    = 16
min_nof_ctrl_symbols = 2
max_nof_ctrl_symbols = 2
```

Step1-3 eNB設定(embms)

將eNB設定檔/etc/srslte/enb.conf 內有關eMBMS的設定參數，亦即[embms]下的參數改為下圖所示之內容。其中m1u_multiaddr為MBMS-GW與eNB的Multicast Group，m1u_if_addr則為綁定該Multicast Group之介面位置

```
#####################################################################
# eMBMS configuration options
#
# enable:               Enable MBMS transmission in the eNB
# m1u_multiaddr:        Multicast addres the M1-U socket will register to
# m1u_if_addr:          Address of the inteferface the M1-U interface will listen for multicast packets.
#
#####################################################################
[embms]
enable = true
m1u_multiaddr = 239.255.0.1
m1u_if_addr = 127.0.1.1
```

Step1-4 eNB設定(rf)

為了使用ZeroMQ傳送無線電訊號的Samples，我們須將eNB設定檔/etc/srslte/enb.conf 內有關無線電實體介面的設定參數，亦即[rf]下的參數改為下圖所示之內容。其中device_name 為指定無線電介面要使用 ZeroMQ，device_args 則指定了ZeroMQ的RX與TX連接阜內容

```
[rf]
#dl_earfcn = 3400
tx_gain = 80
rx_gain = 40

#device_name = auto
#time_adv_nsamples = auto

# Example for ZMQ-based operation with TCP transport for I/Q samples
device_name = zmq
device_args = fail_on_disconnect=true,tx_port=tcp://*:2000,rx_port=tcp://10.0.0.
2:2001,id=enb,base_srate=23.04e6
```

Step1-5 eNB設定(expert)

由於我們使用了ZeroMQ作為無線電介面傳送無線電訊號的Samples，因此須將[expert]內的nof_phy_threads內容設定為1，以避免在PHY層使用多個執行緒所造成的race condition

```
[expert]
#pusch_max_its         = 8 # These are half iterations
#pusch_8bit_decoder    = false
nof_phy_threads       = 1
#metrics_period_secs   = 1
#metrics_csv_enable    = false
#metrics_csv_filename  = /tmp/enb_metrics.csv
#pregenerate_signals   = false
#tx_amplitude          = 0.6
#link_failure_nof_err  = 50
#rrc_inactivity_timer  = 60000
#max_prach_offset_us   = 30
#eea_pref_list = EEA0, EEA2, EEA1
#eia_pref_list = EIA2, EIA1, EIA0
```

Step1-6 eNB設定(sib.conf.msbfn)

執行cd /path/to/srslte/srsenb/以將目前的工作目錄移動到srsLTE原始碼目錄下的srsenb目錄，並使用sudo cp ./sib.conf.mbsfn.example /etc/srslte/sib.conf.mbsfn來將eMBMS SIB的設定檔複製到srsLTE的設定目錄下，可使用cat /etc/srslte/sib.conf.mbsfn 來檢視設定檔的正確性，確認sib13存在且sib1的si_mapping_info有13存在即可。

```
sib1 =
{
    intra_freq_reselection = "Allowed";
    q_rx_lev_min = -65;
    //p_max = 3;
    cell_barred = "NotBarred"
    si_window_length = 20;
    sched_info =
    (
        {
            si_periodicity = 16;
            si_mapping_info = [13]; // comm
            // Leave empty or commented to
        }
    );
    system_info_value_tag = 0;
};
```

```
sib13 =
{
    mbsfn_notification_config =
    {
        mbsfn_notification_repetition_coeff = "n2";
        mbsfn_notification_offset = 0;
        mbsfn_notification_sf_index = 1;
    };
    mbsfn_area_info_list_size = 1;
    mbsfn_area_info_list =
    {
        non_mbsfn_region_length = "s2";
        mcch_repetition_period = "rf64";
        mcch_modification_period = "rf512";
        signalling_mcs = "n2";
        mbsfn_area_id = 1;
        notification_indicator = 0;
        mcch_offset = 0;
        sf_alloc_info = 32;
    };
};
```

Step2 MBMS-GW設定

在 MBMS-GW 上 的 終 端 機 內 輸 入 以 下 指 令 vim /etc/srslte/mbms.conf 編輯MBMS-GW的相關設定，並將 [mbms_gw] 下 參 數 改 為 如 下 圖 所 示 ， 其 中 m1u_multi_addr 需 與 eNB 設 定 檔 中 的 Multicast Group (m1u_multiaddr)一致

```
#####################################################################
# MBMS-GW configuration
#
# name:             MBMS-GW name
# sgi_mb_if_name:   SGi-mb TUN interface name
# sgi_mb_if_addr:   SGi-mb interface IP address
# sgi_mb_if_mask:   SGi-mb interface IP mask
# m1u_multi_addr:   Multicast group for eNBs (TODO this should be setup with M2/M3)
# m1u_multi_if:     IP of local interface for multicast traffic
# m1u_multi_ttl:    TTL for M1-U multicast traffic
#
#####################################################################
[mbms_gw]
name = srsmbmsgw01
sgi_mb_if_name = sgi_mb
sgi_mb_if_addr = 172.16.0.254
sgi_mb_if_mask = 255.255.255.255
m1u_multi_addr = 239.255.0.1
m1u_multi_if   = 127.0.1.200
m1u_multi_ttl  = 1
```

Step3-1 UE設定(rf)

為了使用ZeroMQ傳送無線電訊號的Samples，我們須將UE設定檔內有關無線電實體介面的設定參數，亦即[rf]下的參數改為下圖所示之內容。在UE的終端機上面輸入vim /etc/srslte/ue.conf以編輯UE的設定檔，並將[rf]下的參數改為如圖所示之內容

```
[rf]
dl_earfcn = 3400
freq_offset = 0
tx_gain = 80
#rx_gain = 40

#nof_carriers = 1
#nof_antennas = 1

# For best performance in 2x2 MIMO and >= 15 MHz use the following device_args settings:
#     USRP B210: num_recv_frames=64,num_send_frames=64

# For best performance when BW<5 MHz (25 PRB), use the following device_args settings:
#     USRP B210: send_frame_size=512,recv_frame_size=512

#device_args = auto
#time_adv_nsamples = auto
#continuous_tx     = auto

# Example for ZMQ-based operation with TCP transport for I/Q samples
device_name = zmq
device_args = tx_port=tcp://*:2001,rx_port=tcp://10.0.0.1:2000,id=ue,base_srate=23.04e6
```

Step3-2 UE設定(rrc)

為了進行eMBMS的傳輸，我們須將UE設定檔(/etc/srslte/ue.conf)內[rrc]下的mbms_service_id設為0以啟用UE端的eMBMS功能，其設定結果如下圖所示

```
#####################################################################
# RRC configuration
#
# ue_category:        Sets UE category (range 1-5). Default: 4
# release:            UE Release (8 to 10)
# feature_group:      Hex value of the featureGroupIndicators field in the
#                     UECapabilityInformation message. Default 0xe6041000
# mbms_service_id:    MBMS service id for autostarting MBMS reception
#                     (default -1 means disabled)
# mbms_service_port: Port of the MBMS service
#####################################################################
[rrc]
#ue_category       = 4
#release           = 8
#feature_group     = 0xe6041000
mbms_service_id    = 0
#mbms_service_port = 4321
```

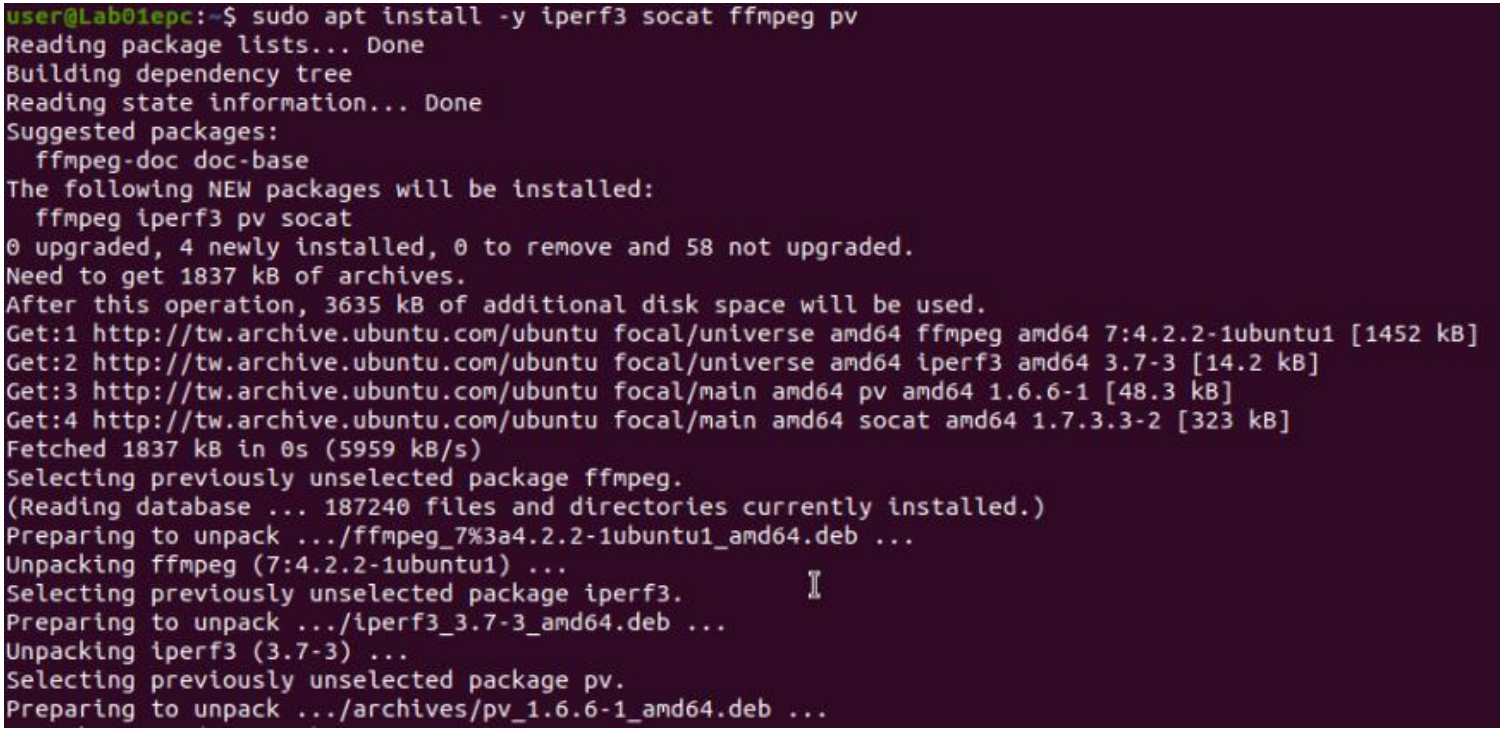
Step3-3 UE設定(phy)

根據srsLTE官方的建議，我們還須將UE設定檔(/etc/srslte/ue.conf)內[phy]下的參數調整成如下圖所示的內容，總共調整了snr_estim_alg、nof_phy_hreads、interpolate_subframe_enabled三項設定

```
[phy]
#rx_gain_offset      = 62
#prach_gain          = 30
#cqi_max             = 15
#cqi_fixed           = 10
#snr_ema_coeff       = 0.1
snr_estim_alg       = empty
#pdsch_max_its       = 8    # These are half iterations
#pdsch_meas_evm      = false
nof_phy_threads     = 1
#equalizer_mode      = mmse
#correct_sync_error  = false
#sfo_ema             = 0.1
#sfo_correct_period  = 10
#sss_algorithm       = full
#estimator_fil_auto  = false
#estimator_fil_stddev  = 1.0
#estimator_fil_order  = 4
#snr_to_cqi_offset   = 0.0
interpolate_subframe_enabled = true
#pdsch_csi_enabled  = true
#pdsch_8bit_decoder = false
#force_ul_amplitude = 0
```

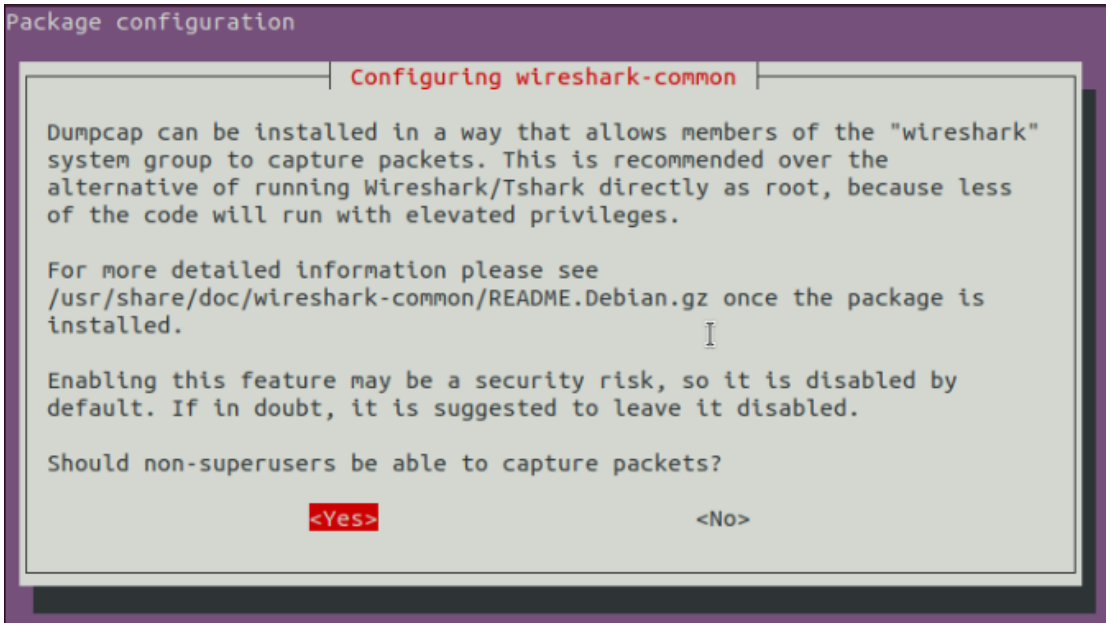
Step4 安裝測試軟體

後續的實驗會使用到iperf3、socat、ffmpeg及pv進行測試，請y在MBMS-GW及UE上的終端機分別輸入sudo apt install -y iperf3 socat ffmpeg pv以安裝此四套軟體



Step5 安裝Wireshark

本實驗中將使用Wireshark進行MBMS-GW的IP封包以及UE MAC PDU的觀測，請在MBMS-GW及UE上分別輸入 sudo add-apt-repository ppa:wireshark-dev/stable; sudo apt update; sudo apt-get -y install wireshark 以安裝Wireshark，如出現以下畫面請選擇 “yes”



Stage 1 Check List

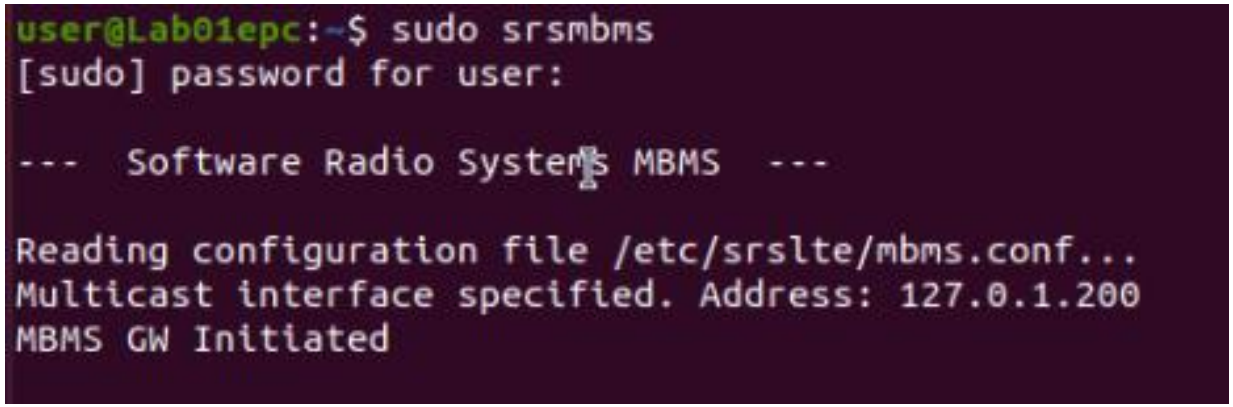
	項目	內容
	eNB上的 /etc/srslte/enb.conf	確認設定是否與投影片上提到的一致
	eNB上的 /etc/srslte/sib.cong.mbsfn	確認檔案是否存在，確認 sib13 存在且 sib1 的 si_mapping_info 有13存在即可
	MBMS-GW上的 /etc/srslte/mbms.conf	確認設定是否與投影片上提到的一致
	UE上的 /etc/srslte/ue.conf	確認設定是否與投影片上提到的一致
	測試軟體	確認iperf3、socat及ffmpeg是否安裝
	Wireshark	輸入wireshark –version確認Wireshark是否正確安裝

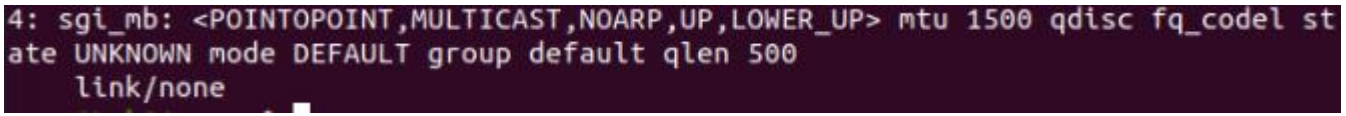
Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 環境設定
- Stage 2. eMBMS 啟動與測試
  - Step1 啟動 MBMS-GW
  - Step2 啟動 EPC
  - Step3 啟動 eNB
  - Step4 啟動 UE
  - Step5 連線測試
  - Step6 路由設定
  - Step7 Multicast 測試
- Stage 3. eMBMS 觀測
- Stage 4. Physical Layer參數調整
- Stage 5. eMBMS應用
- 總結及問題

Step1 啟動MBMS-GW

在MBMS-GW的終端機中輸入sudo srsmbms以啟動srsLTE的MBMS-GW，若啟動成功則畫面應如下圖所示，亦可輸入ip link show以顯示目前的網路介面，應會有一名為sgi_mb的TUN裝置存在





Step2 啟動EPC

在EPC的終端機中輸入sudo srsepc以啟動srsLTE的MME、HSS、S-GW及P-GW，若啟動成功則畫面應如下圖所示，亦可輸入ip link show以顯示目前的網路介面，應會有一名為srs_spgw_sgi的TUN裝置存在

```
user@Lab01epc:~$ sudo srsepc
[sudo] password for user:

Built in Release mode using commit c892ae56b on branch HEAD.


---  Software Radio Systems EPC  ---

Reading configuration file /etc/srslte/epc.conf...
HSS Initialized.
MME S11 Initialized
MME GTP-C Initialized
MME Initialized. MCC: 0xf001, MNC: 0xff01
SPGW GTP-U Initialized.
SPGW S11 Initialized.
SP-GW Initialized.
```

```
6: srs_spgw_sgi: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_co
del state UNKNOWN mode DEFAULT group default qlen 500
    link/none
```

Step3 啟動eNB

在eNB的終端機中輸入sudo srsenb以啟動srsLTE的eNB，若啟動成功則畫面應如下圖左所示，而於此同時EPC上的終端機也應顯示如下圖右的S1連線建立訊息

```
user@Lab01epc:~$ sudo srsenb
[sudo] password for user:
---  Software Radio Systems LTE eNodeB  ---

Reading configuration file /etc/srslte/enb.conf...

Built in Release mode using commit c892ae56b on branch HEAD.

Opening 1 channels in RF device=zmq with args=fail_on_disconnect=true,tx_port=tc
p://*:2000,rx_port=tcp://10.0.0.2:2001,id=enb,base_srate=23.04e6
CHx base_srate=23.04e6
CHx id=enb
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
CH0 rx_port=tcp://10.0.0.2:2001
CH0 tx_port=tcp://*:2000
CH0 fail_on_disconnect=true
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Setting frequency: DL=2685.0 Mhz, UL=2565.0 MHz for cc_idx=0

==== eNodeB started ===
Type <t> to view trace
```

```
user@Lab01epc:~$ sudo srsepc
[sudo] password for user:

Built in Release mode using commit c892ae56b on branch HEAD.


---  Software Radio Systems EPC  ---

Reading configuration file /etc/srslte/epc.conf...
HSS Initialized.
MME S11 Initialized
MME GTP-C Initialized
MME Initialized. MCC: 0xf001, MNC: 0xff01
SPGW GTP-U Initialized.
SPGW S11 Initialized.
SP-GW Initialized.
Received S1 Setup Request.
S1 Setup Request - eNB Name: srsenb01, eNB id: 0x19ba
S1 Setup Request - MCC:001, MNC:01, PLMN: 61712
S1 Setup Request - TAC 0, B-PLMN 0
S1 Setup Request - Paging DRX v128
Sending S1 Setup Response
```

Step4 啟動UE

在UE的終端機中輸入sudo srsue以啟動srsLTE的UE，若啟動成功請成功連接上MBMS-GW則畫面應如下圖左所示，而於此同時EPC與eNB應會顯示如下圖右的連接訊息

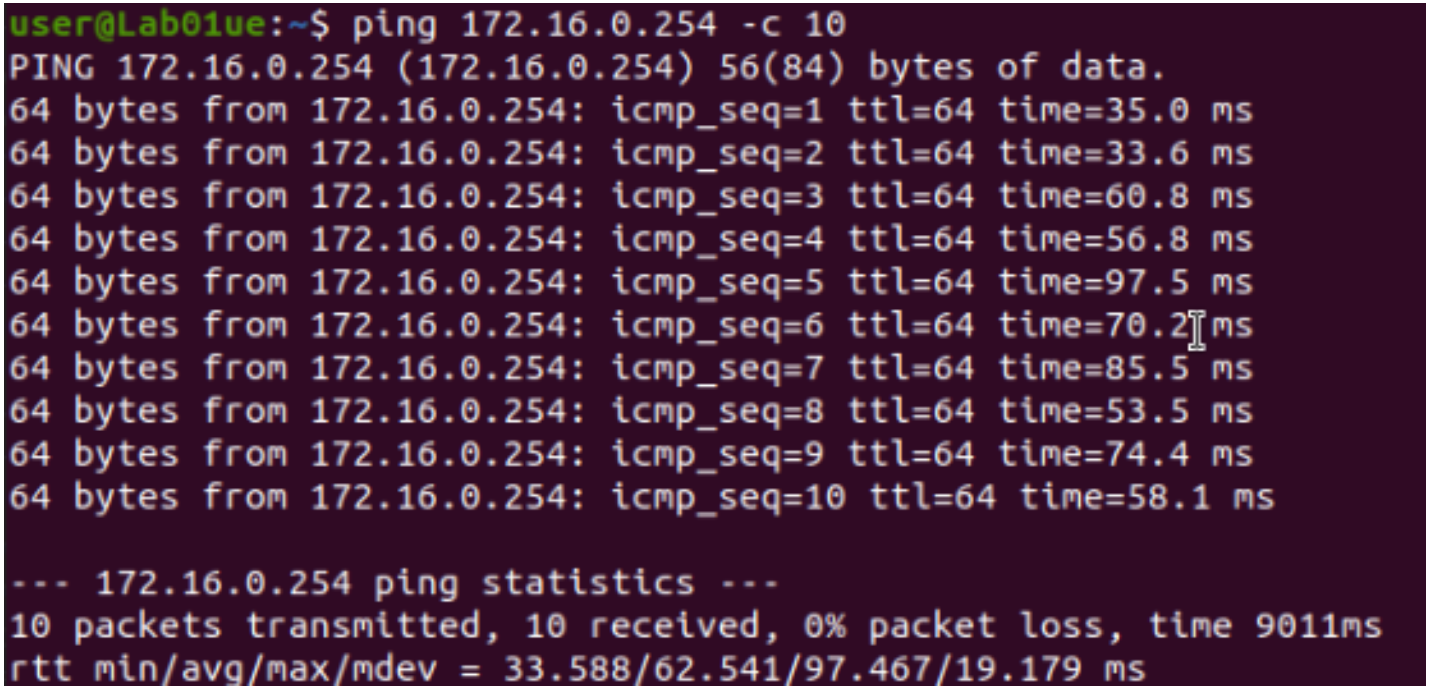
```
user@Lab01ue:~$ sudo srsue
[sudo] password for user:
Reading configuration file /etc/srslte/ue.conf...

Built in Release mode using commit c892ae56b on branch HEAD.

Opening 1 channels in RF device=zmq with args=tx_port=tcp://*:2001,rx_port=tcp:
e_srate=23.04e6
CHx base_srate=23.04e6
CHx id=ue
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
CH0 rx_port=tcp://10.0.0.1:2000
CH0 tx_port=tcp://*:2001
Waiting PHY to initialize ... done!
Attaching UE...
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
.
Found Cell:  Mode=FDD, PCI=1, PRB=50, Ports=1, CFO=-0.2 KHz
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Found PLMN:  Id=00101, TAC=7
Random Access Transmission: seq=42, ra-rnti=0x2
Random Access Complete.     c-rnti=0x46, ta=0
RRC Connected
MBMS service started. Service id=0, port=4321, lcid=1
Network attach successful. IP: 172.16.0.2
Software Radio Systems LTE (srsLTE)
```

```
Authentication Response -- IMSI 001010123456789
UE Authentication Accepted.
Generating KeNB with UL NAS COUNT: 0
Downlink NAS: Sending NAS Security Mode Command.
UL NAS: Received Security Mode Complete
Security Mode Command Complete -- IMSI: 00101012345
6789
Getting subscription information -- QCI 7
Sending Create Session Request.
Creating Session Response -- IMSI: 1010123456789
Creating Session Response -- MME control TEID: 1
Received GTP-C PDU. Message type: GTPC_MSG_TYPE_CRE
ATE_SESSION_REQUEST
SPGW: Allocated Ctrl TEID 1
SPGW: Allocated User TEID 1
SPGW: Allocate UE IP 172.16.0.2
Received Create Session Response
Create Session Response -- SPGW control TEID 1
Create Session Response -- SPGW S1-U Address: 127.0
.1.100
SPGW Allocated IP 172.16.0.2 to IMSI 00101012345678
9
Adding attach accept to Initial Context Setup Reque
st
Sent Initial Context Setup Request. E-RAB id 5
Received Initial Context Setup Response
E-RAB Context Setup. E-RAB id 5
E-RAB Context -- eNB TEID 0x460003; eNB GTP-U Addre
ss 127.0.1.1
UL NAS: Received Attach Complete
Unpacked Attached Complete Message. IMSI 1010123456
789
Unpacked Activate Default EPS Bearer message. EPS B
earer id 5
Received GTP-C PDU. Message type: GTPC_MSG_TYPE_MOD
IFY_BEARER_REQUEST
Sending EMM Information
```

```
user@Lab01epc:~$ sudo srsenb
[sudo] password for user:
---  Software Radio Systems LTE eNodeB  ---

Reading configuration file /etc/srslte/enb.conf...

Built in Release mode using commit c892ae56b on bra
nch HEAD.

Opening 1 channels in RF device=zmq with args=fail_
on_disconnect=true,tx_port=tcp://*:2000,rx_port=tcp
://10.0.0.2:2001,id=enb,base_srate=23.04e6
CHx base_srate=23.04e6
CHx id=enb
Current sample rate is 1.92 MHz with a base rate of
 23.04 MHz (x12 decimation)
CH0 rx_port=tcp://10.0.0.2:2001
CH0 tx_port=tcp://*:2000
CH0 fail_on_disconnect=true
Current sample rate is 11.52 MHz with a base rate o
f 23.04 MHz (x2 decimation)
Current sample rate is 11.52 MHz with a base rate o
f 23.04 MHz (x2 decimation)
Setting frequency: DL=2685.0 Mhz, UL=2565.0 MHz for
 cc_idx=0

==== eNodeB started ===
Type <t> to view trace
RACH:  tti=181, preamble=42, offset=0, temp_crnti=0
x46
User 0x46 connected
```

Step5 連線測試

在UE的終端機中輸入ping 172.16.0.254 -c 10以測試UE與MBMS-GW中間的連接是否出現問題，成功則畫面應如下圖所示可以成功收到MBMS-GW的回應



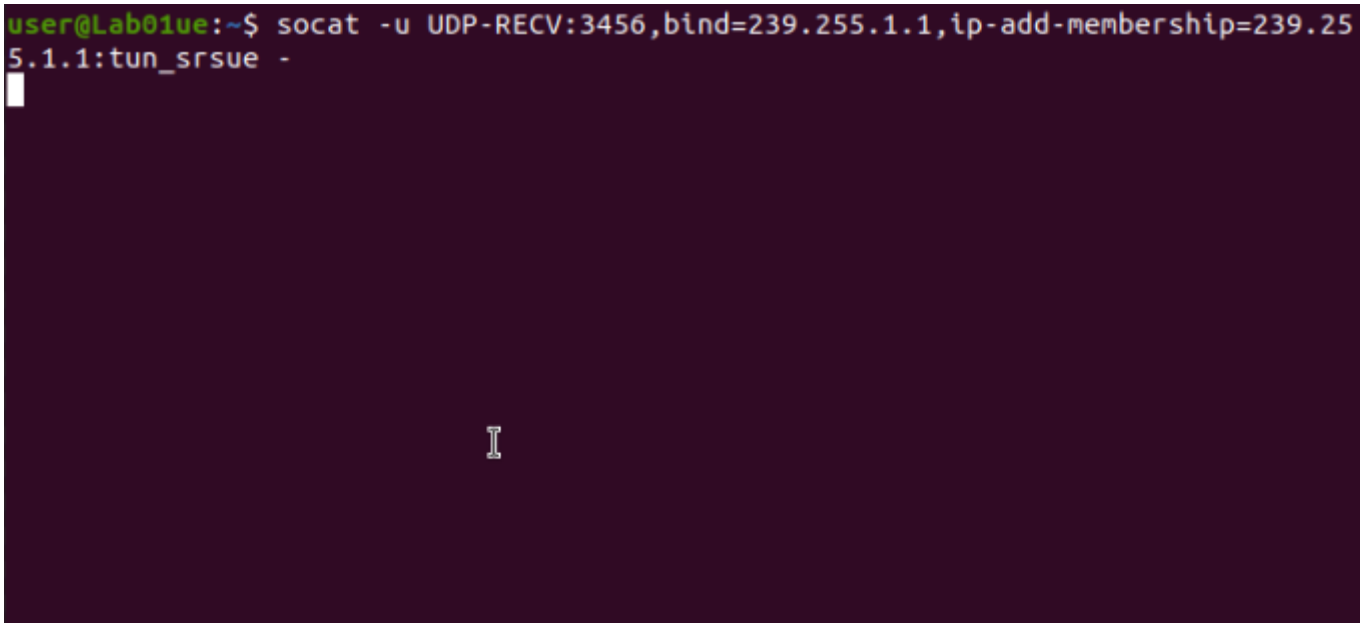
Step6 路由設定

在 MBMS-GW 的終端機中輸入 sudo ip route add 239.255.1.0/24 dev sgi_mb 以確保後續所測試的Multicast封包會由sgi_mb介面進入MBMS-GW，可使用 ip route show 來檢查設定是否正確，正確設定應如下圖所示

```
user@Lab01epc:~$ ip route show
default via 10.0.2.2 dev ens3 proto dhcp metric 100
10.0.0.0/24 dev ens4 proto kernel scope link src 10.0.0.1 metric 101
10.0.2.0/24 dev ens3 proto kernel scope link src 10.0.2.15 metric 100
169.254.0.0/16 dev ens4 scope link metric 1000
172.16.0.0/24 dev srs_spgw_sgi proto kernel scope link src 172.16.0.1
239.255.1.0/24 dev sgi_mb scope link
```

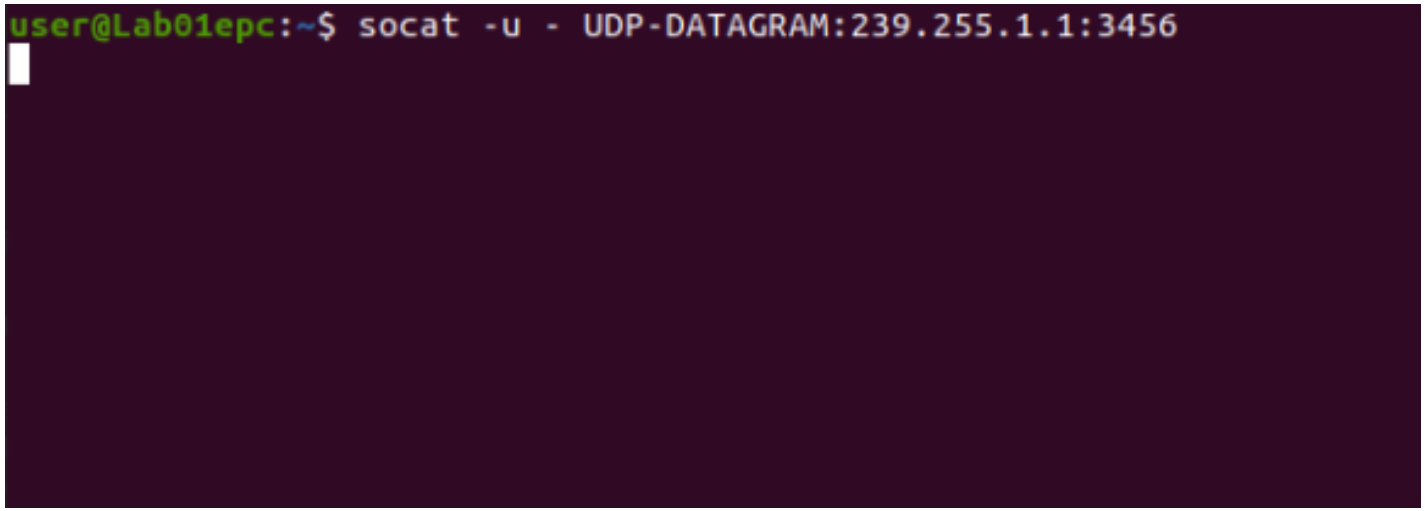
Step7-1 Multicast 測試(接收)

在 UE 的終端機中輸入 socat -u UDP-RECV:3456, bind=239.255.1.1,ip-add-membership=239.255.1.1:tun_srsue - 此指令會開啟一個 UDP socket 並加入 Multicast Group 239.255.1.1，並將所收到的資料重新導向至螢幕，當執行這個指令時沒有輸出任何東西屬正常現象，請先將其放置



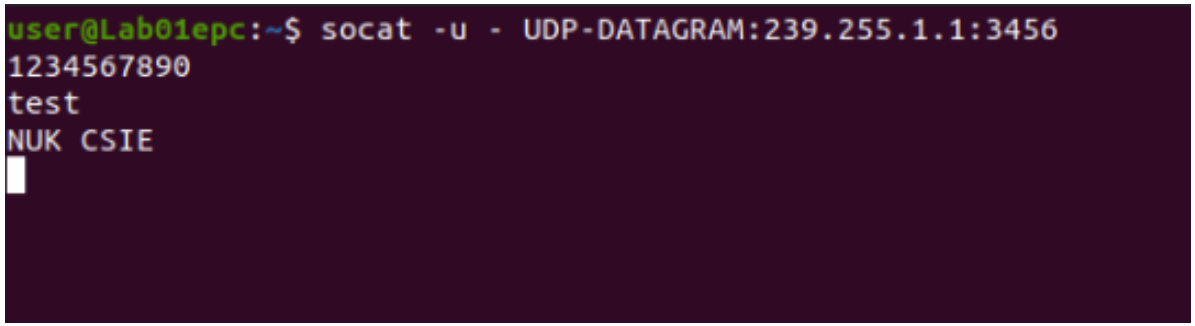
Step7-2 Multicast 測試(傳送)

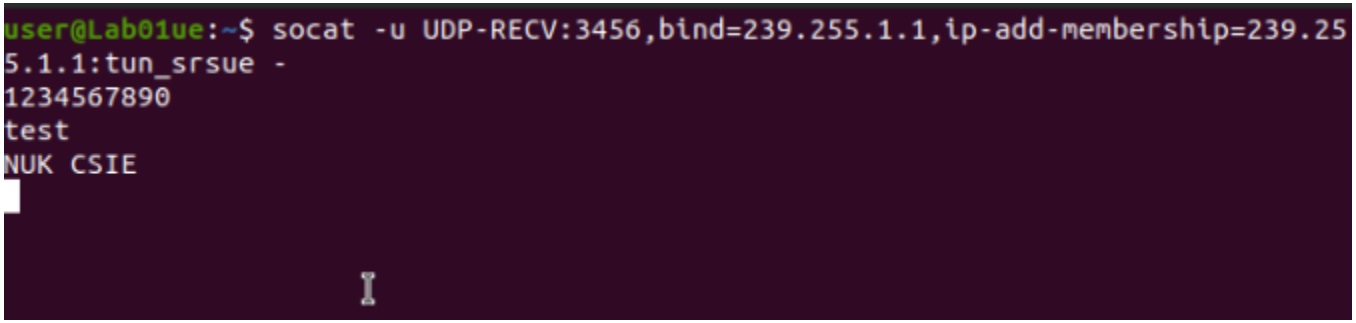
在MBMS-GW的終端機中輸入以下指令以進行資料傳送 socat -u - UDP-DATAGRAM:239.255.1.1:3456，此指令會開啟一個UDP socket並將由鍵盤輸入的任何東西導向至該socket。值得注意的是在Multicast中傳送者不須加入Multicast Group即可傳送資料



Step7-3 Multicast 測試(結果)

接下來請在MBMS-GW中的傳送端上打上任意字串後按下Enter，經過一小段時間若看到UE上的接收端收到相同字串則代表由MBMS-GW向UE進行Multicast的能力正常運作





Stage 2 Check List

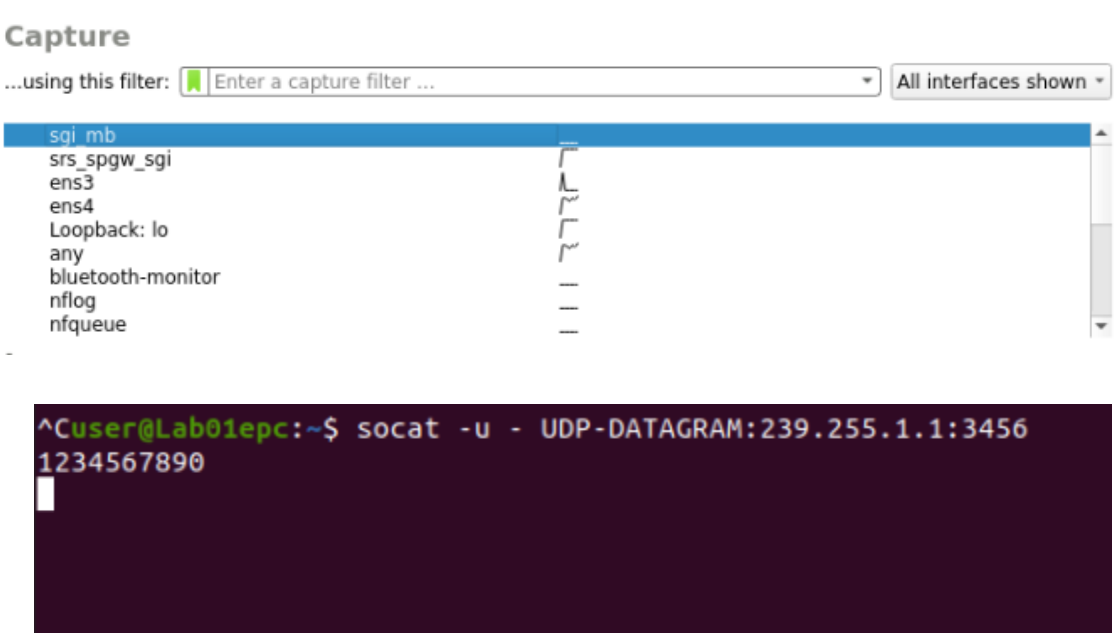
	項目	內容
	MBMS-GW	srsmbms正常啟動且出現一名為sgi_mb的TUN裝置
	EPC	srsepc正常啟動且出現一名為srs_spgw_sgi的TUN裝置
	eNB	srsenb正常啟動且成功與EPC建立S1連線
	UE	srsue正常啟動且成功連接上eNB並取得MBMS Service ID
	Multicast	從MBMS-GW可以正常的對UE已經加入的Multicast Group 進行 Multicast

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 環境設定
- Stage 2. eMBMS 啟動與測試
- Stage 3. eMBMS 觀測
  - Step1 觀測EPC封包
  - Step2 導出UE MAC PDU
  - Step3 觀測導出的PDU
  - Step4 觀測MIB及SIB
  - Step5 觀測MCH
  - Step6 觀測eMBMS的影響
- Stage 4. Physical Layer參數調整
- Stage 5. eMBMS應用
- 總結及問題

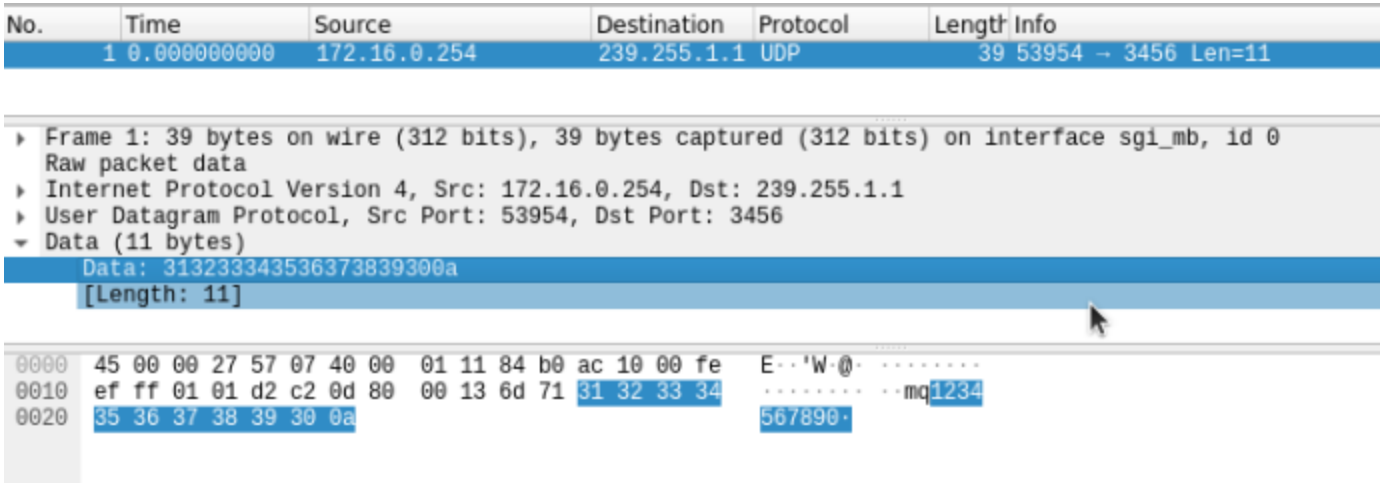
Step1-1 觀測EPC封包(sgi_mb)

請在MBMS-GW中開啟Wireshark並選擇sgi_mb介面進行觀測，當觀測開始後再開啟一個終端機並輸入socat -u - UDP-DATAGRAM:239.255.1.1:3456開啟傳送端，成功開起傳送端後請輸入1234567890並按下Enter



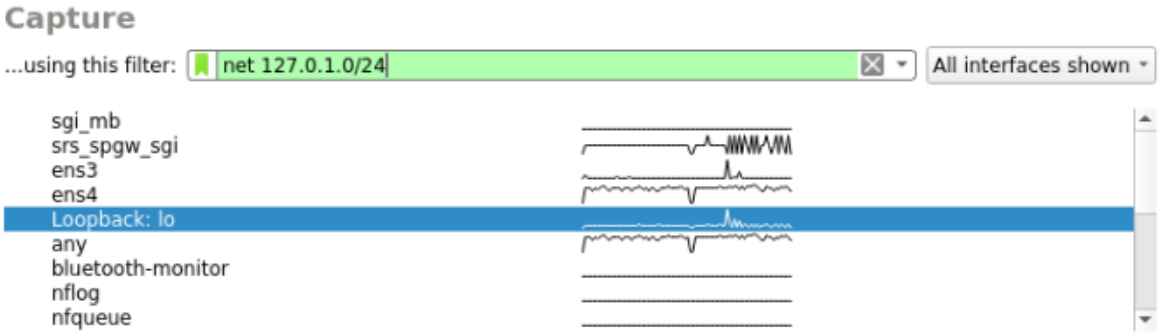
Step1-2 觀測EPC封包(sgi_mb結果)

接下來應會在MBMS-GW上的Wireshark內看到一個UDP封包，檢視其內容可以發現為剛剛所傳送的字串，這顯示了前面設定的路由規則有正常運作，傳送至239.255.1.1的封包確實被轉送至sgi_mb介面中



Step1-3 觀測EPC封包(M1)

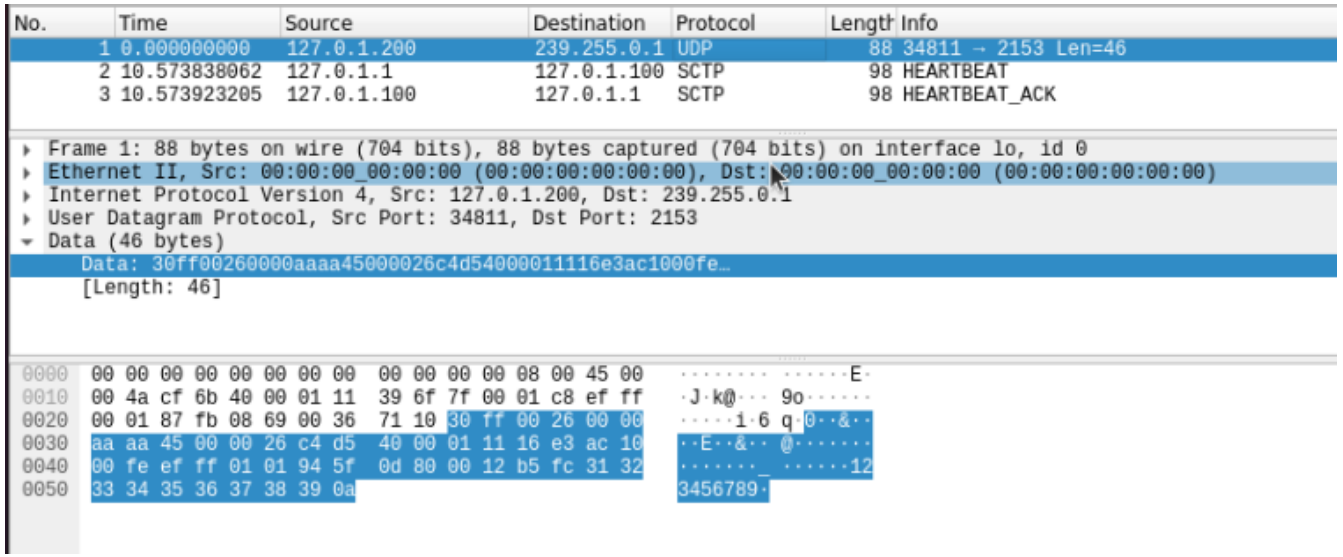
將上一個步驟的Wireshark關閉後重新開啟Wireshark並選擇lo介面，擷取過濾器則輸入net 127.0.1.0/24以檢視eNB與MBMS-GW間的M1介面訊息，開始擷取封包後請重新開啟Multicast傳送者並傳送123456789




```
user@Lab01epc:~$ socat -u - UDP-DATAGRAM:239.255.1.1:3456
1234567890
test
NUK CSIE
^Cuser@Lab01epc:~$ socat -u - UDP-DATAGRAM:239.255.1.1:3456
1234567890
^Cuser@Lab01epc:~$ socat -u - UDP-DATAGRAM:239.255.1.1:3456
123456789
```

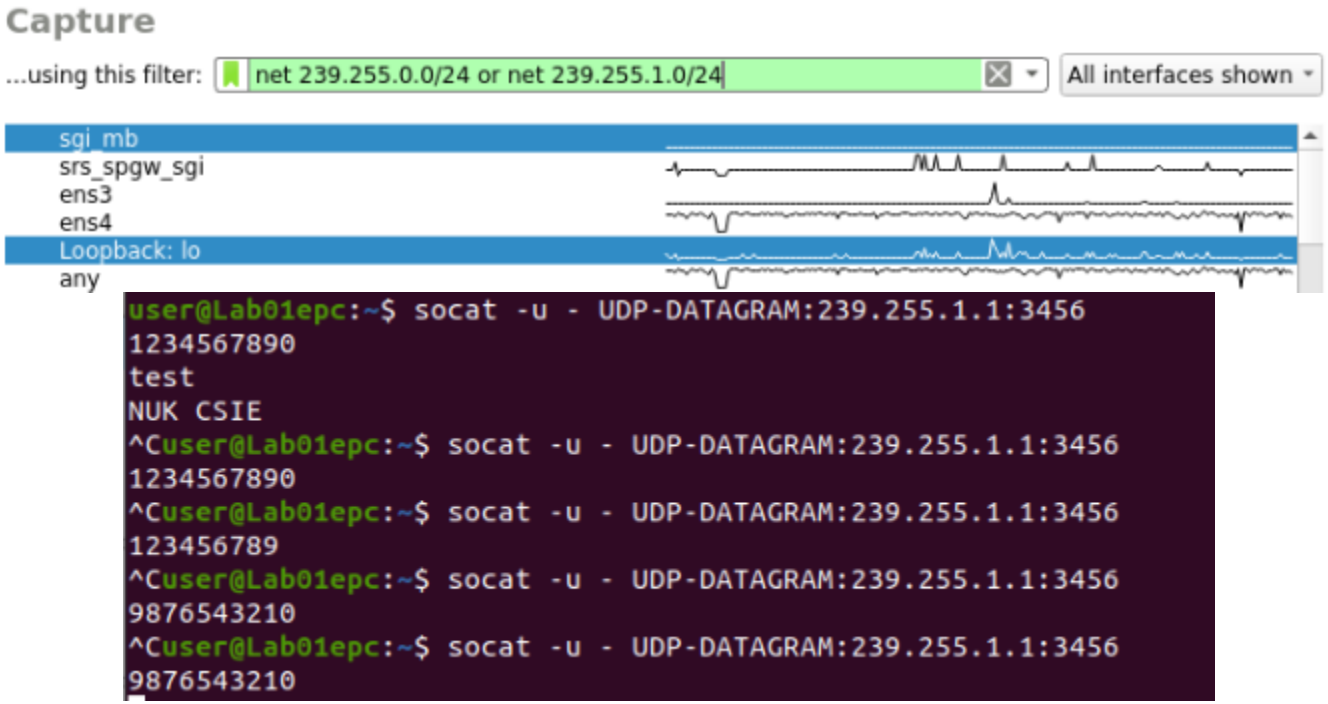
Step1-4 觀測EPC封包(M1結果)

接下來應會在MBMS-GW的Wireshark內看到一UDP封包，其內容為剛剛所傳送的字串加上一Wireshark無法辨認的Header，由eMBMS的架構判斷該封包應為以GTP封裝過的封包，但是因其目標連接阜並非標準GTP之連接阜，故Wireshark無法判讀其內容。



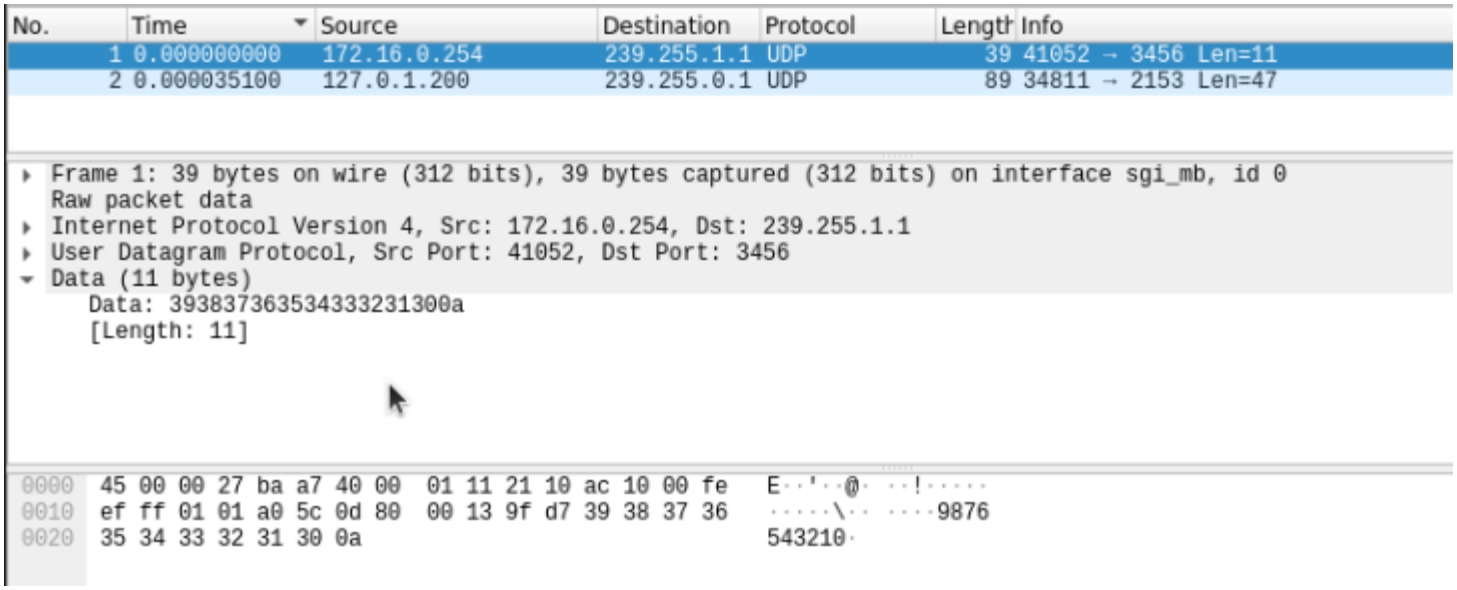
Step1-5 觀測EPC封包(流向)

將上一個步驟的Wireshark關閉後重新開啟Wireshark並選擇sgi_mb與lo介面，擷取過濾器則輸入net 239.255.0.0/24 or net 239.255.1.0/24以檢視SGI_MB及M1介面的封包，開始擷取封包後請重新開啟Multicast傳送者並傳送9876543210



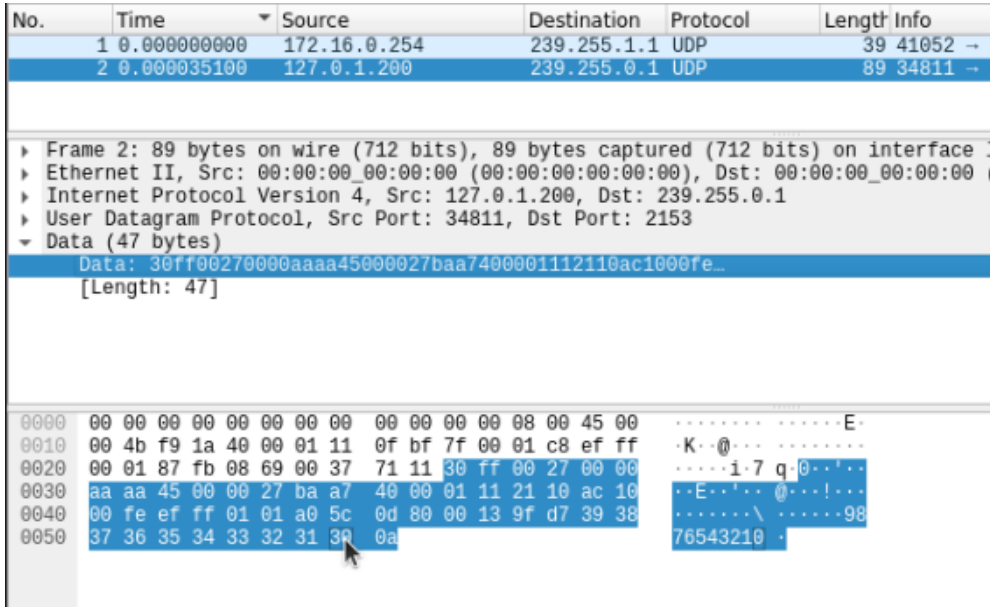
Step1-6 觀測EPC封包(流向結果)

接下來應會在MBMS-GW的Wireshark內看到兩個UDP封包，案照時間排序後可發現第一個封包為Multicast傳送者送給sgi_mb介面的封包，而第二個封包為MBMS-GW以GTP封裝後透過M1介面的Multicast Group傳送至eNB的封包，以此可以確定資料確實是由MBMS-GW傳送至eNB。



Step1-7 觀測EPC封包(GTP)

參考TS 29.060中的GTP Header後我們可以發現Wireshark無法辨認的Header確實為GTP的Header，其版本為GTPv1、被封裝的封包長度為39 bytes(0x27)且其TEID為0x0000AAAA



Bits

Octets	8	7	6	5	4	3	2	1
1	Version			PT	(*)	E	S	PN
2	Message Type							
3	Length (1st Octet)							
4	Length (2nd Octet)							
5	Tunnel Endpoint Identifier (1st Octet)							
6	Tunnel Endpoint Identifier (2nd Octet)							
7	Tunnel Endpoint Identifier (3rd Octet)							
8	Tunnel Endpoint Identifier (4th Octet)							
9	Sequence Number (1st Octet)[1) 4)]							
10	Sequence Number (2nd Octet)[1) 4)]							
11	N-PDU Number[2) 4)]							
12	Next Extension Header Type[3) 4)]							

Step2-1 導出UE MAC PDU(mkfifo)

為了能及時的觀測到UE MAC層的PDU，須先建立一個Named Pipe(fifo)作為PDU擷取時的暫存區。在UE上開啟一個新的終端機並輸入mkfifo /tmp/ue.pcap.pipe以建立一個Named Pipe，可以使用file /tmp/ue.pcap.pipe來檢查其檔案型態是否為fifo(named pipe)

```
user@Lab01ue:~$ sudo mkfifo /tmp/ue.pcap.pipe
[sudo] password for user:
user@Lab01ue:~$ file /tmp/ue.pcap.pipe
/tmp/ue.pcap.pipe: fifo (named pipe)
user@Lab01ue:~$ /
```

Step2-2 導出UE MAC PDU(設定)

在UE的終端機輸入sudo vim /etc/srslte/ue.conf以編輯UE的設定檔，並將[pcap]下的設定值改為如下圖所示之內容，編輯完後請重啟eNB及UE

```
# add an entry with DLT=147, Payload Protocol=mac-lte-framed.
# For more information see: https://wiki.wireshark.org/MAC-LTE
# NAS-layer packets are dissected with DLT=148, and
# Payload Protocol = nas-eps.
#
# enable:       Enable MAC layer packet captures (true/false)
# filename:     File path to use for MAC packet captures
# nas_enable:   Enable NAS layer packet captures (true/false)
# nas_filename: File path to use for NAS packet captures
###################################################################
[pcap]
enable = true
filename = /tmp/ue.pcap.pipe
nas_enable = false
nas_filename = /tmp/nas.pcap
```

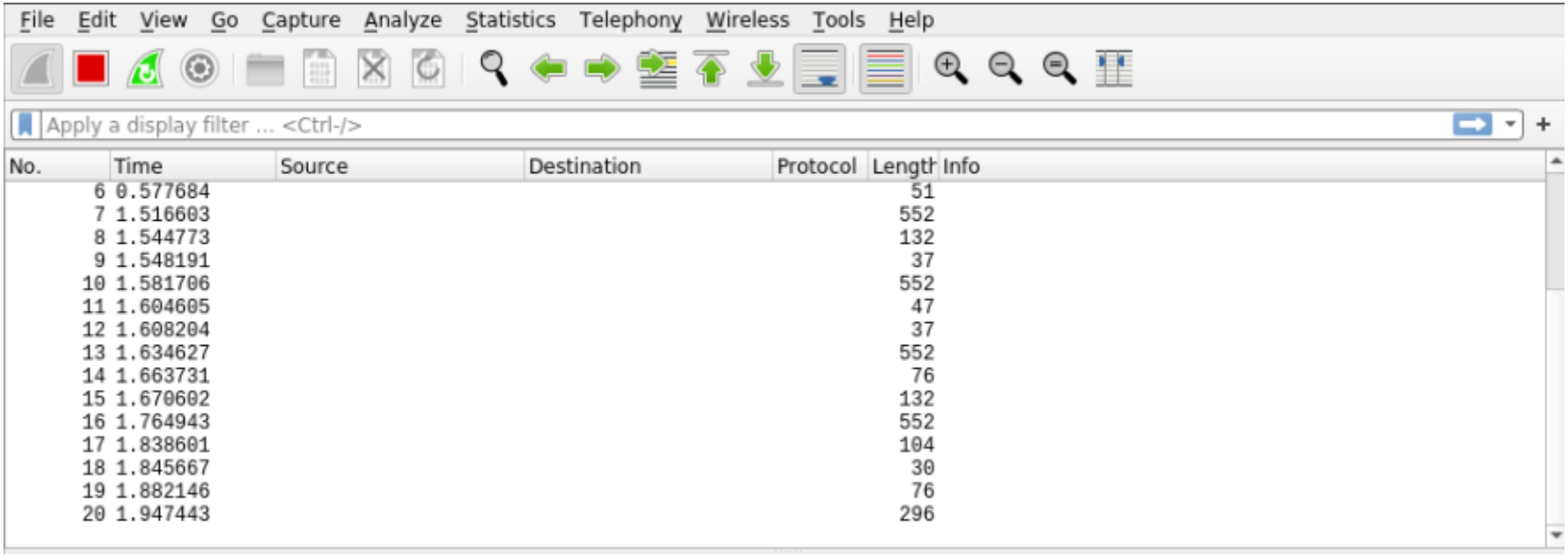
Step3-1 觀測導出的PDU (重啟)

按順序重啟eNB及UE後會發現UE停在如下圖所示的畫面，這是因為Named Pipe的Consumer還沒啟動，因此UE在寫入Name Pipe時被block所導致，屬正常現象繼續操作即可

```
user@Lab01ue:~$ sudo srsue
Reading configuration file /etc/srslte/ue.conf...

Built in Release mode using commit c892ae56b on branch HEAD.

Opening 1 channels in RF device=zmq with args=tx_port=tcp://*:2001,rx_port=tcp://10.0.0.1:2000,id=ue,bas
e_srate=23.04e6
CHx base_srate=23.04e6
CHx id=ue
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
CH0 rx_port=tcp://10.0.0.1:2000
CH0 tx_port=tcp://*:2001
```

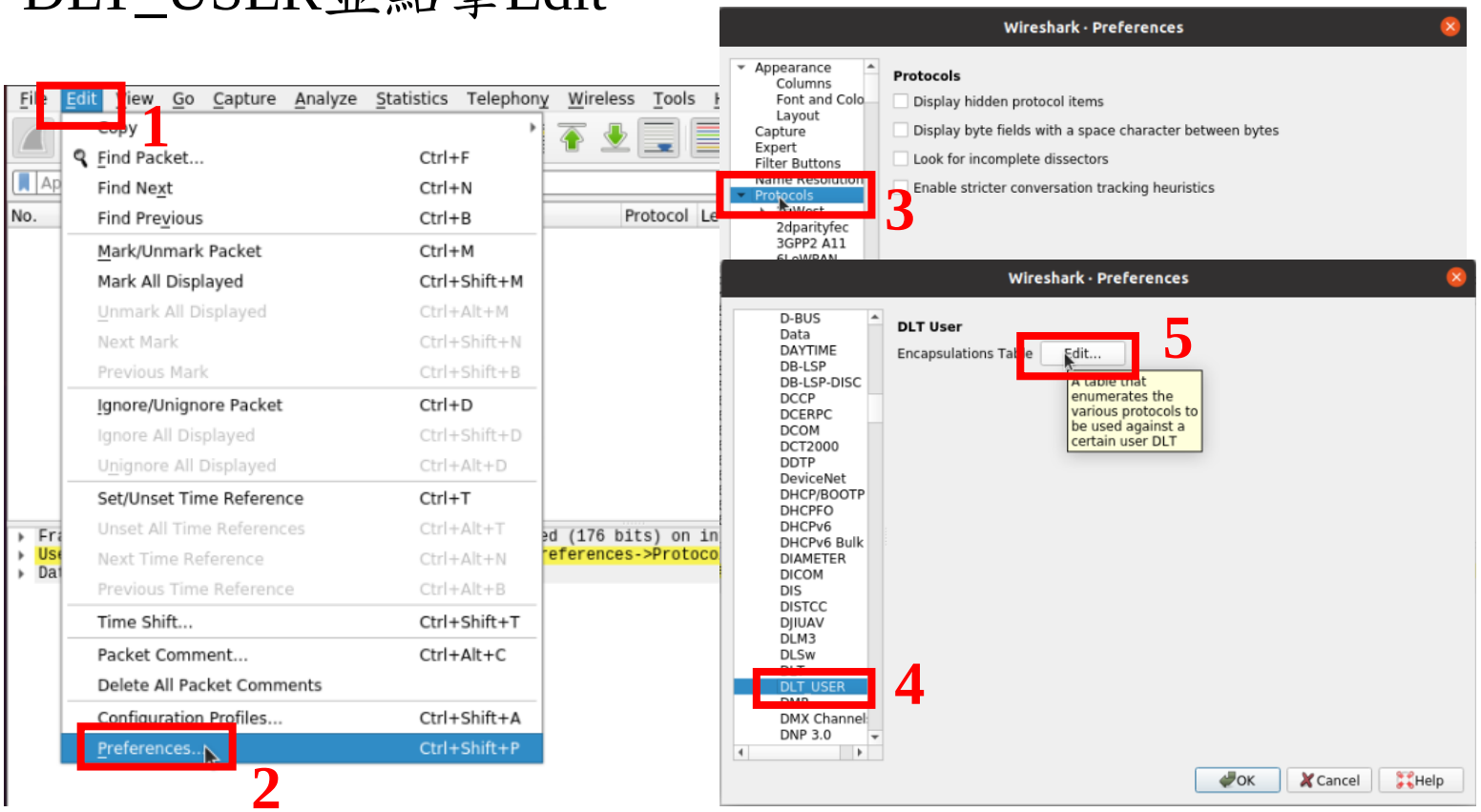
Step3-2 觀測導出的PDU (開啟)

在UE的終端機輸入sudo wireshark -k -i /tmp/ue.pcap.pipe以啟動Wireshark並立及觀測從UE MAC層所導出的PDU，Wireshark啟動後UE應會繼續先前的連線動作，並且應可在Wireshark的介面上看到如圖所示未被辨認的封包



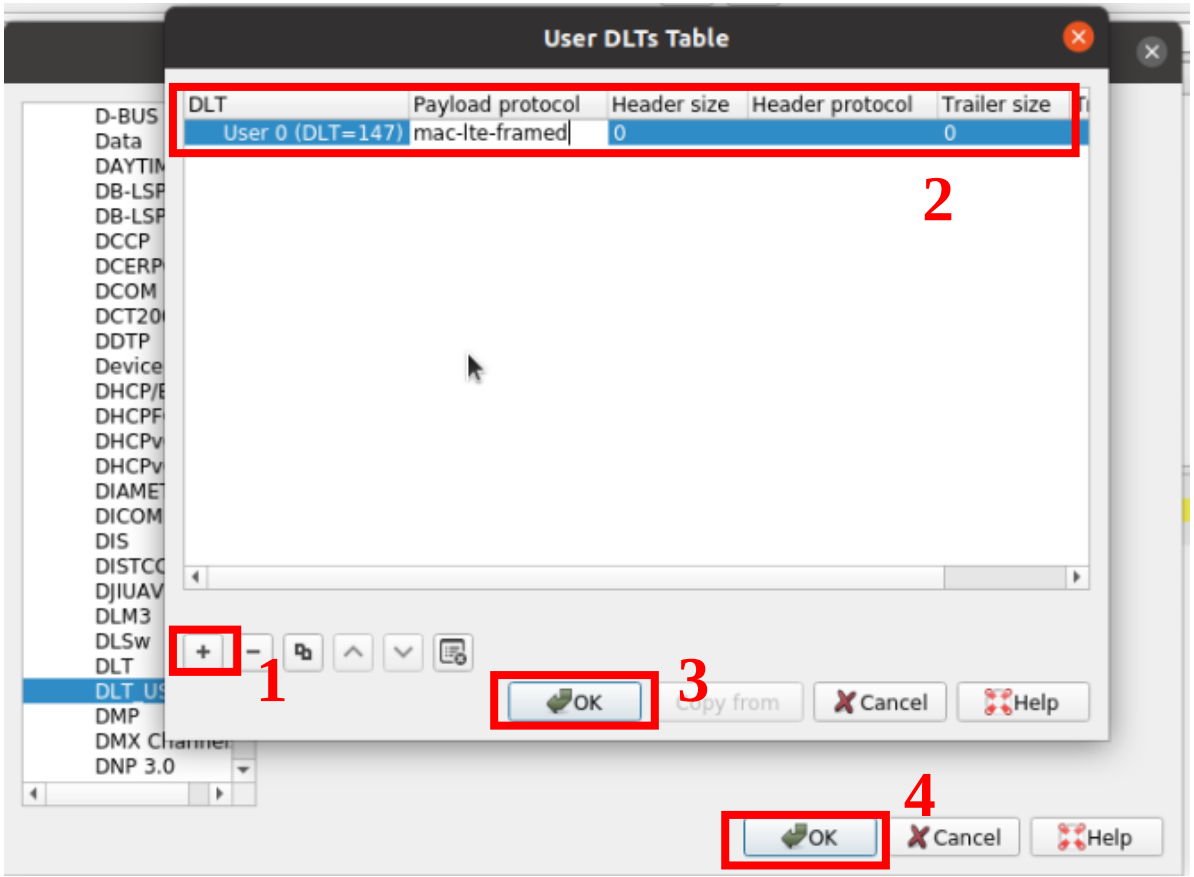
Step3-3 觀測導出的PDU (設定選單)

在 Wireshark 依 序 開 啟 Edit->Preferences->Protocols->DLT_USER並點擊Edit



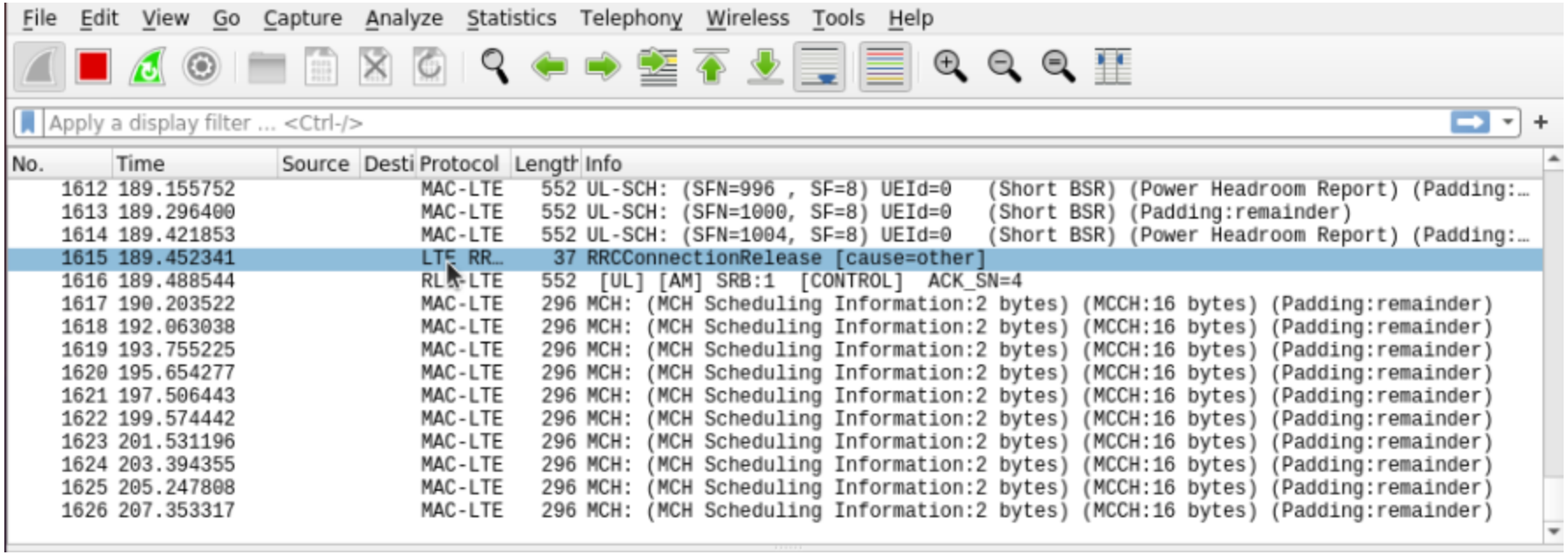
Step3-4 觀測導出的PDU (新增設定)

於彈出視窗中新增一筆DLT=147且Payload protocol為mac-lte-framed之自訂訊框解析器，然後按下OK儲存



Step3-5 觀測導出的PDU (解析結果)

設定完成後回到Wireshark的主介面，應會看到剛剛無法解析的封包現在都已順利被解析，其中Protocol部分會顯示該訊框為MAC-LTE、LTE-RRC或是RLC-LTE，而Info部分則會顯示該封包所屬的Transport Channel及相關資訊



Step3-6 觀測導出的PDU (注意事項)

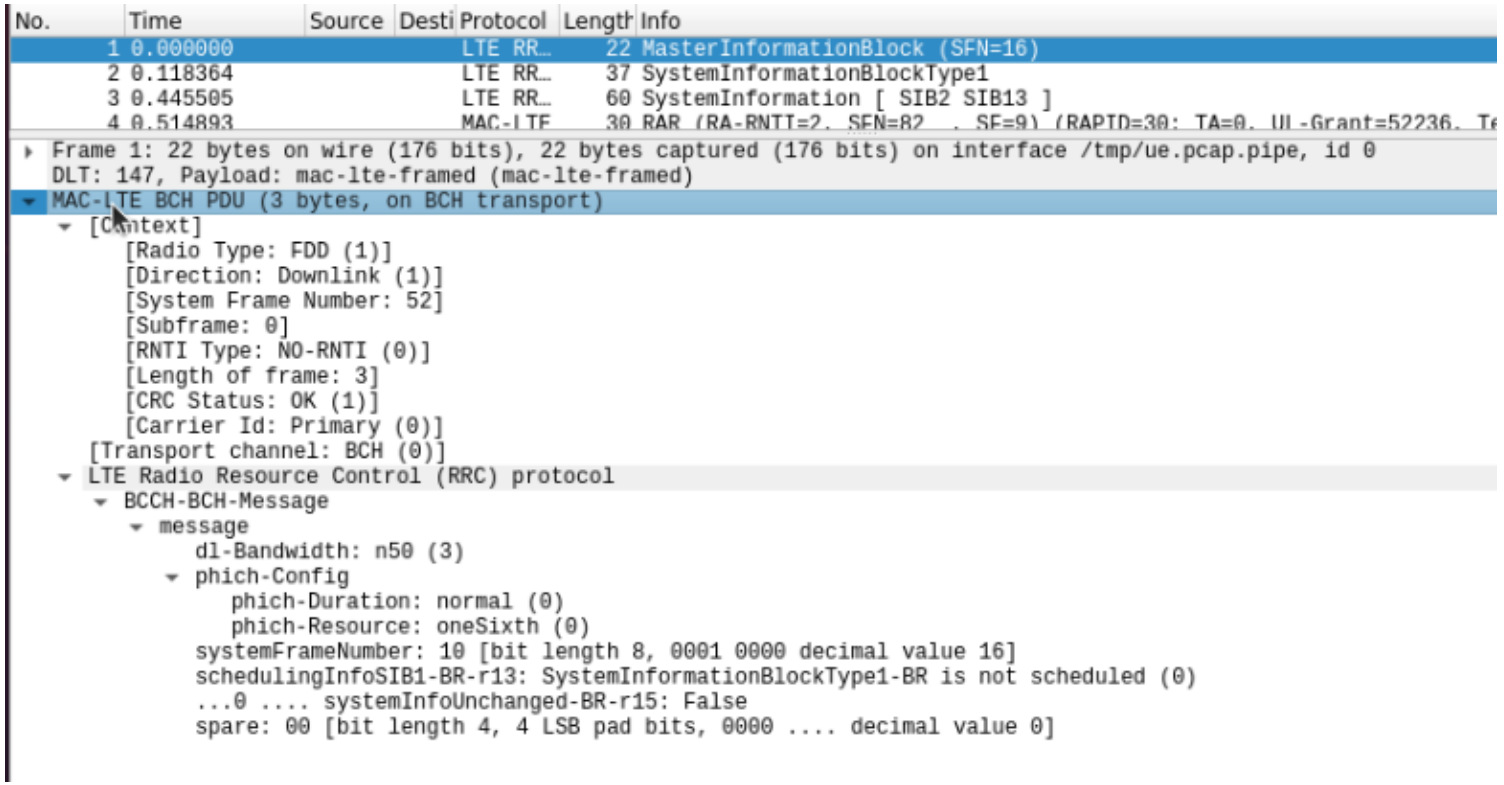
若使用Wireshark進行UE MAC PDU的觀測須注意以下兩點

1. UE會等Wireshark開啟後才會順利動作
2. 停止觀測或是關閉Wireshark視窗會導致srsUE crash

```
CH0 rx_port=tcp://10.0.0.1:2000
CH0 tx_port=tcp://*:2001
Waiting PHY to initialize ... done!
Attaching UE...
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
.
Found Cell:  Mode=FDD, PCI=1, PRB=50, Ports=1, CFO=-0.2 KHz
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Found PLMN:  Id=00101, TAC=7
Random Access Transmission: seq=21, ra-rnti=0x2
Random Access Complete.     c-rnti=0x46, ta=0
RRC Connected
Network attach successful. IP: 172.16.0.8
MBMS service started. Service id=0, port=4321, lcid=1
Software Radio Systems LTE (srsLTE)
srsLTE crashed... backtrace saved in './srsLTE.backtrace.crash'...
---  exiting  ---
srsLTE crashed... backtrace saved in './srsLTE.backtrace.crash'...
```

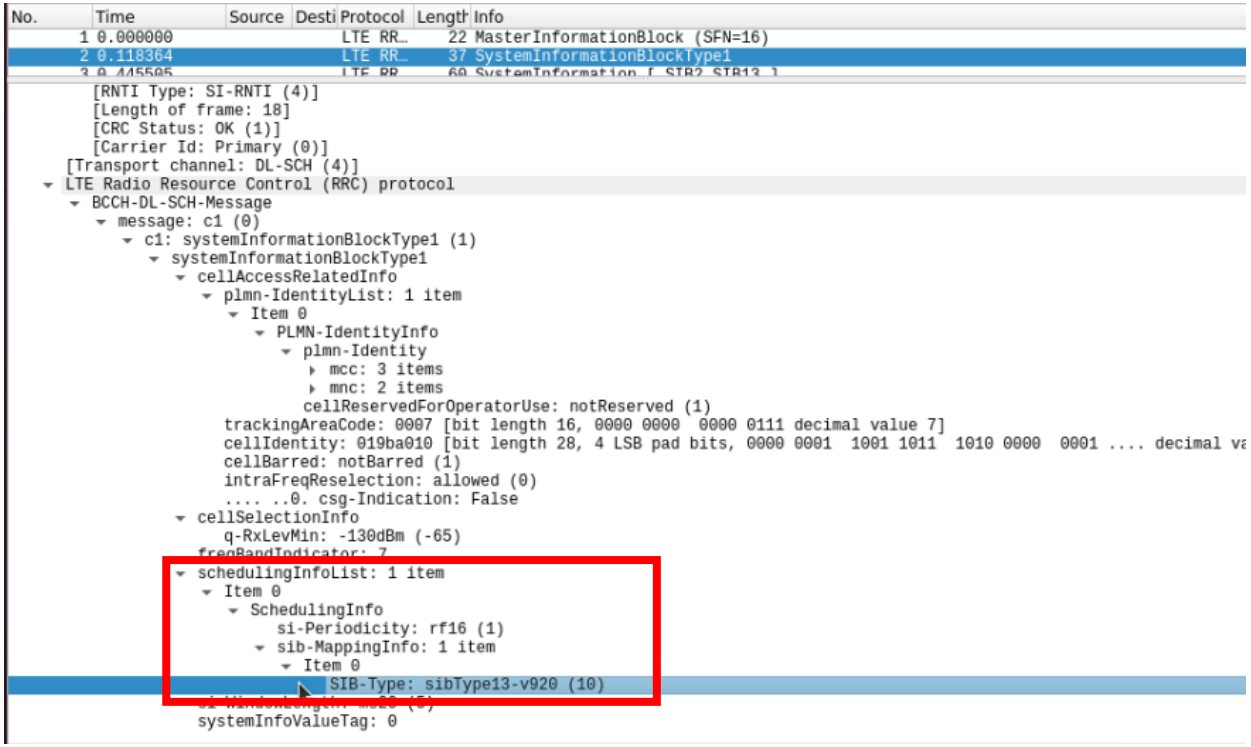
Step4-1 觀測MIB及SIB(MIB)

UE MAC PDU的第一個訊框為Master Information Block (MIB)，其中包含了PHY層的重要設定以及SIB1的排程資訊，為LTE系統中最重要的資訊塊



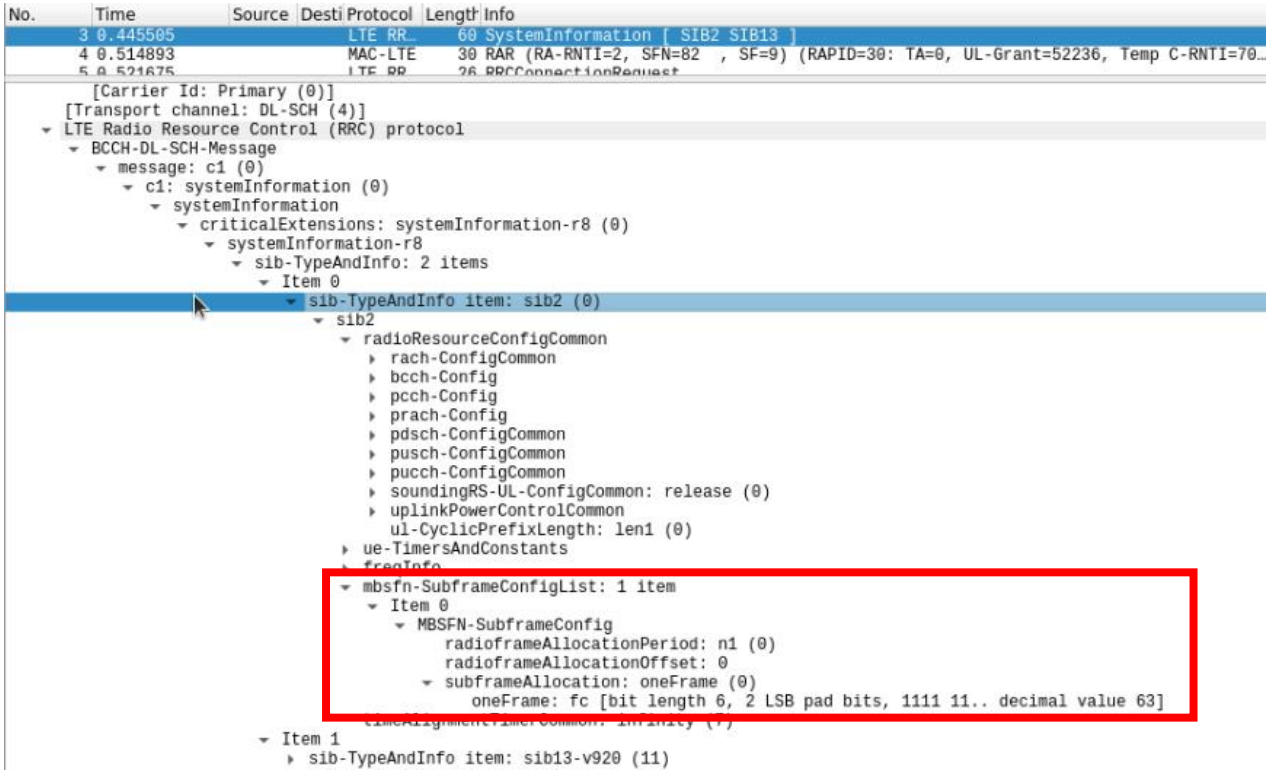
Step4-2 觀測MIB及SIB(SIB1)

System Information Block Type1(SIB1)為最基本且最重要的SIB，其內涵資訊包括該eNB的MCC與MNC、其他SIB的排程資訊(如週期、Window長度等等)，本實驗中需確認SIB13是否在SIB1的排程資訊內。



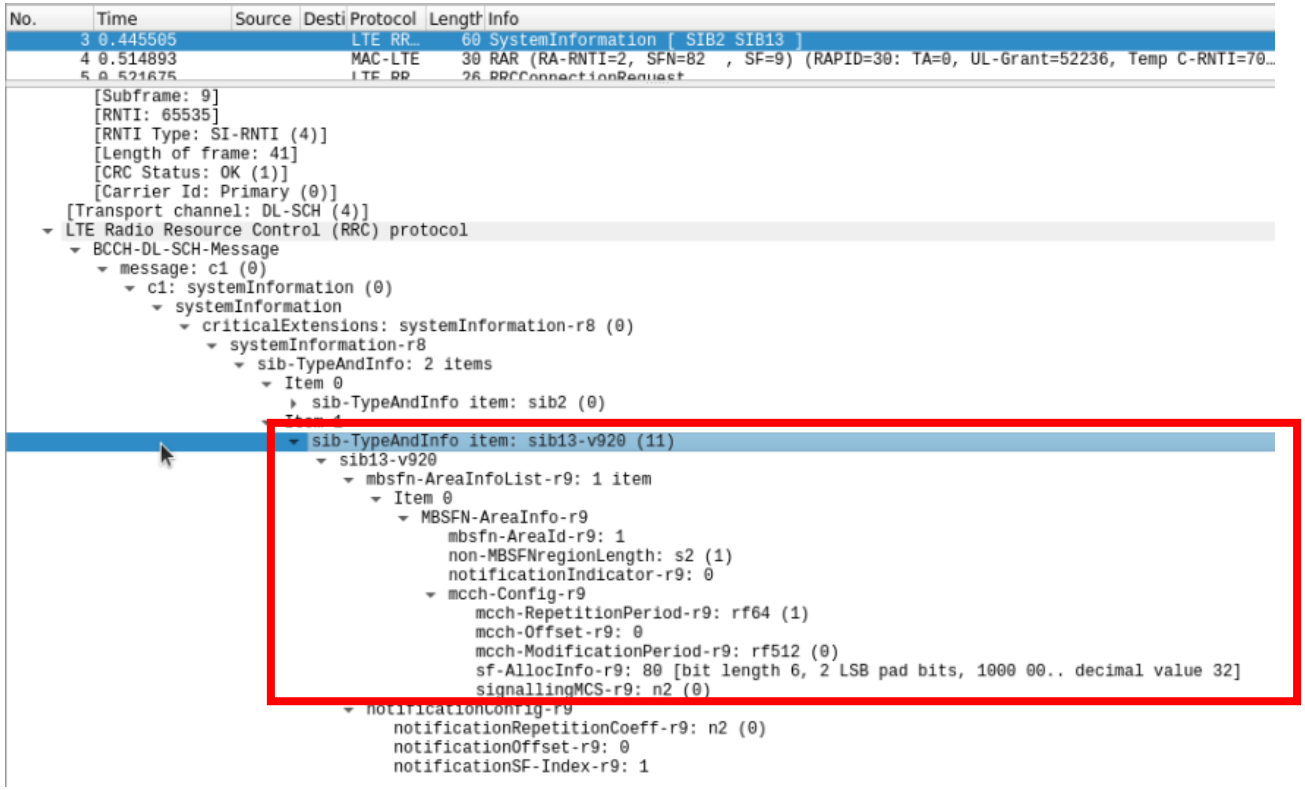
Step4-3 觀測MIB及SIB(SIB2)

SIB2包含一些共享Transport Channel的設定資訊，如BCCH、PCCH等。本實驗中eMBMS資訊傳遞所使用的MCH設定也在SIB2內，包括Radio Frame及Sub-Frame的位置分配等



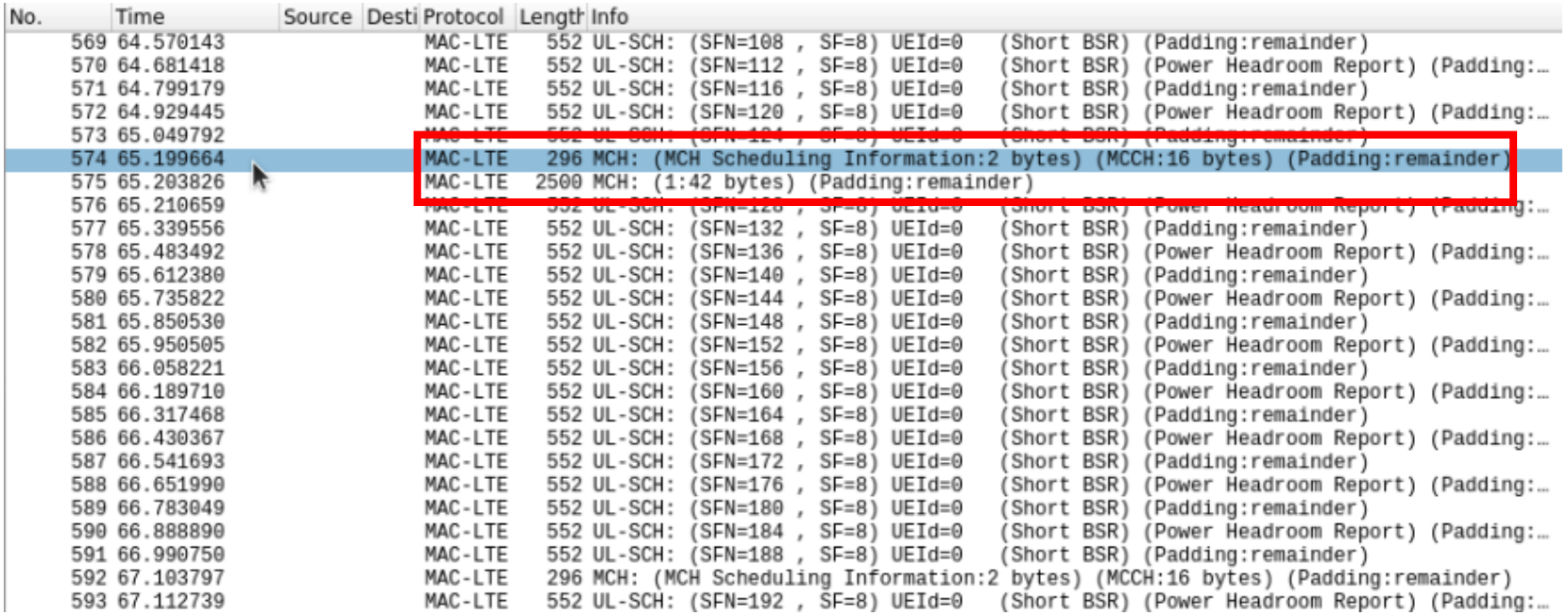
Step4-4 觀測MIB及SIB(SIB13)

SIB13為eMBMS專用的SIB，其內容包含MCCH所占用的Radio Frame位置及Sub-Frame位置等



Step5-1 觀測MCH(產生測試流量)

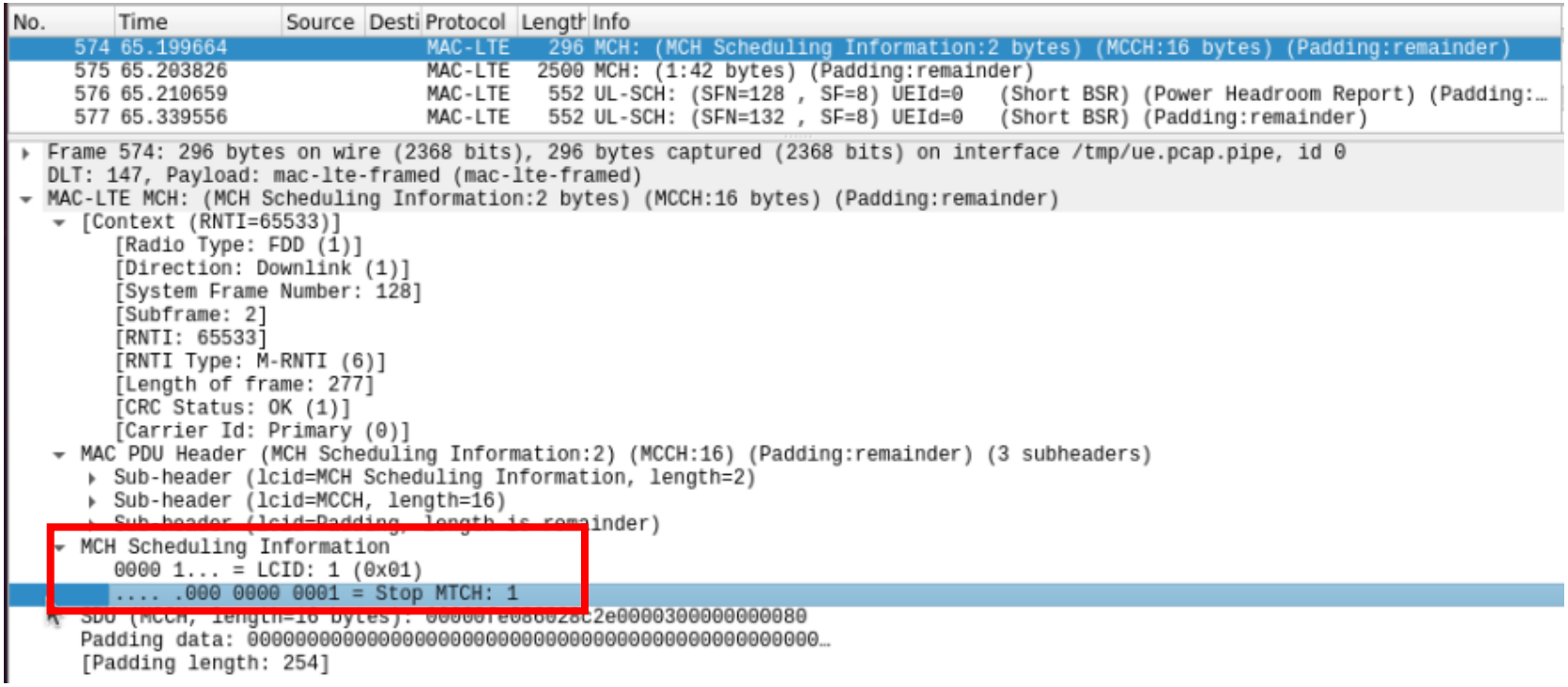
在MBMS-GW的終端機輸入echo 1234567890 | socat -u - UDP-DATAGRAM:239.255.1.1:3456以產生測試的封包，在UE端的Wireshark內應可看到兩個連續的MCH訊框



No.	Time	Source	Desti	Protocol	Length	Info
569	64.570143			MAC-LTE	552	UL-SCH: (SFN=108 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
570	64.681418			MAC-LTE	552	UL-SCH: (SFN=112 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…
571	64.799179			MAC-LTE	552	UL-SCH: (SFN=116 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
572	64.929445			MAC-LTE	552	UL-SCH: (SFN=120 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…
573	65.049792			MAC-LTE	552	UL-SCH: (SFN=124 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
574	65.199664			MAC-LTE	296	MCH: (MCH Scheduling Information:2 bytes) (MCCH:16 bytes) (Padding:remainder)
575	65.203826			MAC-LTE	2500	MCH: (1:42 bytes) (Padding:remainder)
576	65.210659			MAC-LTE	552	UL-SCH: (SFN=128 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…
577	65.339556			MAC-LTE	552	UL-SCH: (SFN=132 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
578	65.483492			MAC-LTE	552	UL-SCH: (SFN=136 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…
579	65.612380			MAC-LTE	552	UL-SCH: (SFN=140 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
580	65.735822			MAC-LTE	552	UL-SCH: (SFN=144 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…
581	65.850530			MAC-LTE	552	UL-SCH: (SFN=148 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
582	65.950505			MAC-LTE	552	UL-SCH: (SFN=152 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…
583	66.058221			MAC-LTE	552	UL-SCH: (SFN=156 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
584	66.189710			MAC-LTE	552	UL-SCH: (SFN=160 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…
585	66.317468			MAC-LTE	552	UL-SCH: (SFN=164 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
586	66.430367			MAC-LTE	552	UL-SCH: (SFN=168 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…
587	66.541693			MAC-LTE	552	UL-SCH: (SFN=172 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
588	66.651990			MAC-LTE	552	UL-SCH: (SFN=176 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…
589	66.783049			MAC-LTE	552	UL-SCH: (SFN=180 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
590	66.888890			MAC-LTE	552	UL-SCH: (SFN=184 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…
591	66.990750			MAC-LTE	552	UL-SCH: (SFN=188 , SF=8) UEId=0 (Short BSR) (Padding:remainder)
592	67.103797			MAC-LTE	296	MCH: (MCH Scheduling Information:2 bytes) (MCCH:16 bytes) (Padding:remainder)
593	67.112739			MAC-LTE	552	UL-SCH: (SFN=192 , SF=8) UEId=0 (Short BSR) (Power Headroom Report) (Padding:…

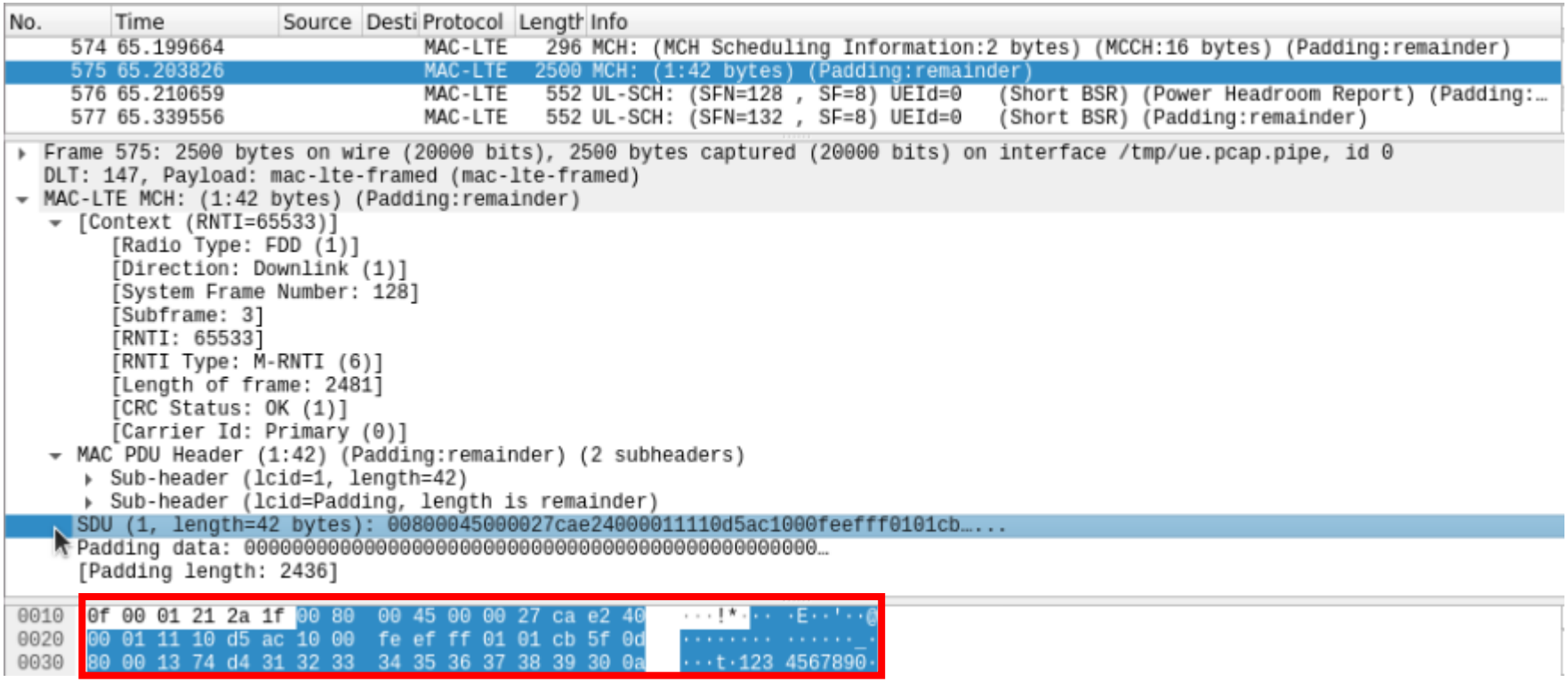
Step5-2 觀測MCH(MCCH)

點擊第一個訊框以顯示其詳細內容，可以發現該訊框所對應之Logical Channel為MCCH，內含MTCH的排程資訊。可以觀察到此訊框表示了緊接著的一個MCH對應到MTCH



Step5-3 觀測MCH(MTCH)

點擊下一個訊框可以發現該訊框所對應之Logical Channel為MTCH，其SDU承載著以00 80 00為內容的上層Header及完整的UDP/IP封包，最上層的內容為剛剛所傳送的字串



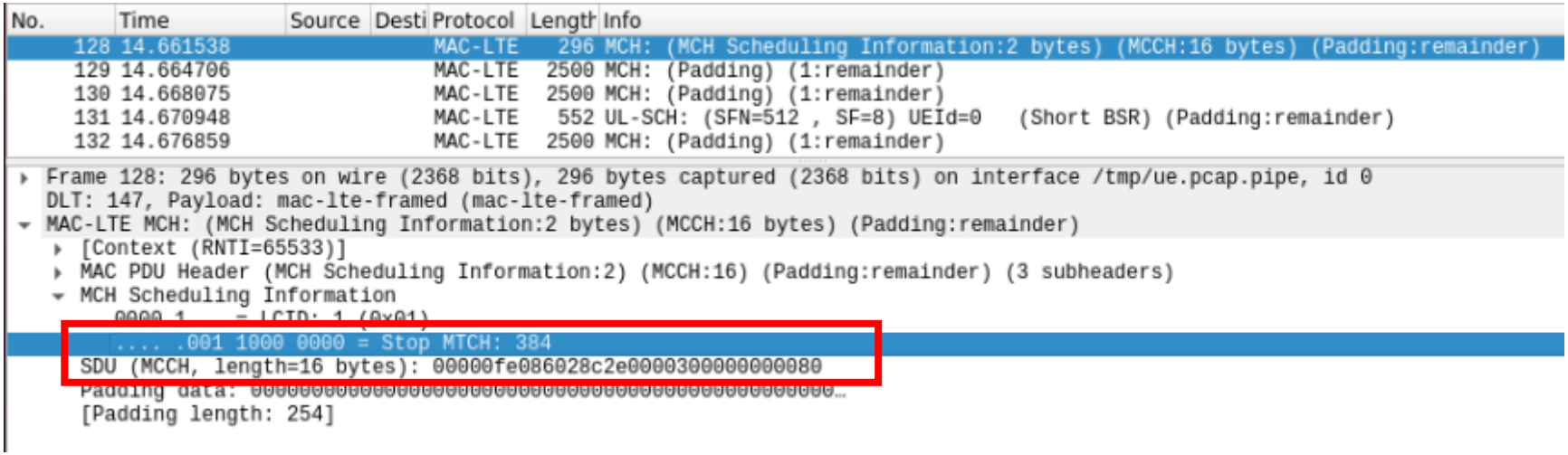
Step5-4 觀測MCH(大流量)

在 MBMS-GW 的 終 端 機 輸 入 dd if=/dev/zero bs=1M count=10 | socat -u - UDP-DATAGRAM:239.255.1.1:3456以傳送10MB的資料到UE，可以在UE端的Wireshark上觀察到隨著資料增加，MCH訊框的數量也隨之上升

```
128 14.661538    MAC-LTE   296 MCH: (MCH Scheduling Information:2 bytes) (MCCH:16 bytes) (Padding:remainder)
129 14.664706    MAC-LTE  2500 MCH: (Padding) (1:remainder)
130 14.668075    MAC-LTE  2500 MCH: (Padding) (1:remainder)
131 14.670948    MAC-LTE   552 UL-SCH: (SFN=512 , SF=8) UEId=0   (Short BSR) (Padding:remainder)
132 14.676859    MAC-LTE  2500 MCH: (Padding) (1:remainder)
133 14.680028    MAC-LTE  2500 MCH: (Padding) (1:remainder)
134 14.685688    MAC-LTE  2500 MCH: (Padding) (1:remainder)
135 14.694806    MAC-LTE  2500 MCH: (Padding) (1:remainder)
136 14.697637    MAC-LTE  2500 MCH: (Padding) (1:remainder)
137 14.700616    MAC-LTE  2500 MCH: (Padding) (1:remainder)
138 14.709482    MAC-LTE  2500 MCH: (Padding) (1:remainder)
139 14.712142    MAC-LTE  2500 MCH: (Padding) (1:remainder)
140 14.714948    MAC-LTE  2500 MCH: (Padding) (1:remainder)
141 14.723613    MAC-LTE  2500 MCH: (Padding) (1:remainder)
142 14.726921    MAC-LTE  2500 MCH: (Padding) (1:remainder)
143 14.729388    MAC-LTE  2500 MCH: (Padding) (1:remainder)
144 14.737641    MAC-LTE  2500 MCH: (Padding) (1:remainder)
145 14.741330    MAC-LTE  2500 MCH: (Padding) (1:remainder)
146 14.744028    MAC-LTE  2500 MCH: (Padding) (1:remainder)
147 14.752553    MAC-LTE  2500 MCH: (Padding) (1:remainder)
148 14.756123    MAC-LTE  2500 MCH: (Padding) (1:remainder)
149 14.758805    MAC-LTE  2500 MCH: (Padding) (1:remainder)
150 14.769362    MAC-LTE  2500 MCH: (Padding) (1:remainder)
151 14.772230    MAC-LTE  2500 MCH: (Padding) (1:remainder)
```

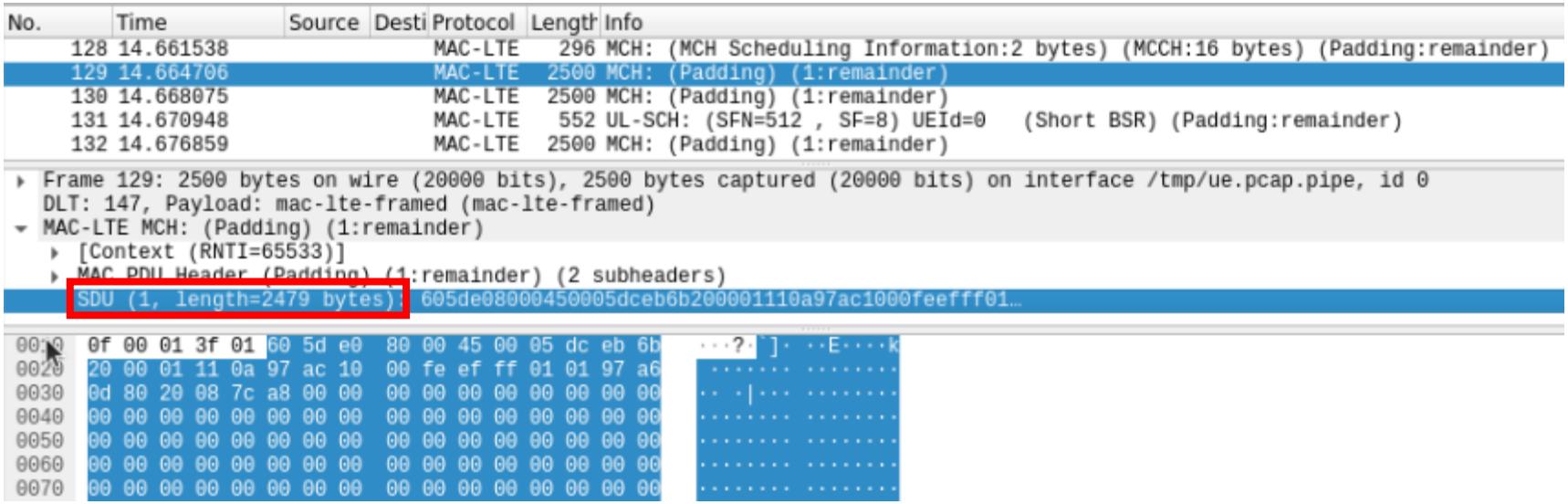
Step5-5 觀測MCH(大流量-MCCH)

透過觀察攜帶MCCH的MCH訊框我們可以觀察到當流量增加時，由於單靠一個MCH訊框無法傳送完整的資料，因此封包被分段並透過多個MCH訊框傳送。以圖中的MCCH內容可以知道在其後面的MCH訊框有連續384個訊框為MTCH所佔用



Step5-6 觀測MCH(大流量-MTCH)

點及其後的MCH訊框可以發現其對應到的Logical Channel為MTCH，而其內容為原始封包的一小部分。由於該訊框不含Padding，所以由此可以知道MCH之MTU為2479 bytes



Step6-1 觀測eMBMS的影響(iperf)

在EPC上執行iperf3 -s以啟動iperf3的Server，並在UE上執行iperf3 -c 172.16.0.1 -t 100 -i 5 --bidir以進行正常Uplink與Downlink流量的模擬與觀測

```
user@Lab01epc:~$ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

```
user@Lab01ue:~$ iperf3 -c 172.16.0.1 -t 100 -i 5 --bidir
Connecting to host 172.16.0.1, port 5201
[  5] local 172.16.0.3 port 41666 connected to 172.16.0.1 port 5201
[  7] local 172.16.0.3 port 41668 connected to 172.16.0.1 port 5201
[ ID][Role] Interval           Transfer     Bitrate         Retr  Cwnd
[  5][TX-C]   0.00-5.00   sec  2.39 MBytes  4.01 Mbits/sec    0    153 KBytes
[  7][RX-C]   0.00-5.00   sec   547 KBytes   897 Kbits/sec
[  5][TX-C]   5.00-10.00  sec  1.49 MBytes  2.50 Mbits/sec   15   97.6 KBytes
[  7][RX-C]   5.00-10.00  sec  1.81 MBytes  3.03 Mbits/sec
[  5][TX-C]  10.00-15.00  sec  1.49 MBytes  2.50 Mbits/sec    0    160 KBytes
[  7][RX-C]  10.00-15.00  sec  1.90 MBytes  3.18 Mbits/sec
[  5][TX-C]  15.00-20.00  sec  2.11 MBytes  3.54 Mbits/sec   66    133 KBytes
[  7][RX-C]  15.00-20.00  sec  1.80 MBytes  3.02 Mbits/sec
```

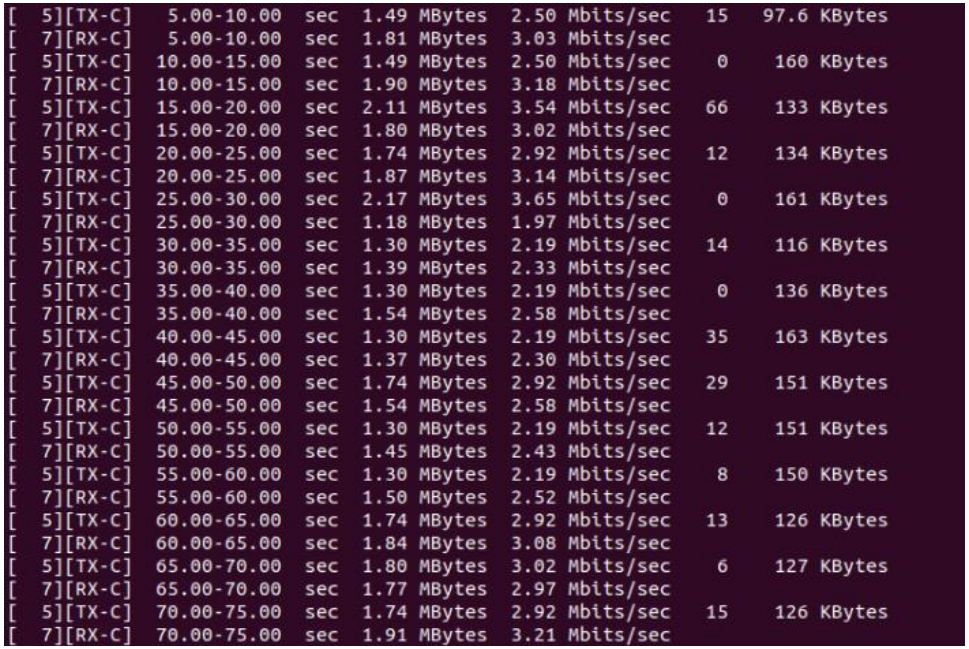
Step6-2 觀測eMBMS的影響(大流量)

在MBMS-GW上執行dd if=/dev/zero bs=1M count=2000 | socat -u - UDP-DATAGRAM:239.255.1.1:3456，此指令會產生2GB的流量並以eMBMS的方式傳送，請記下本指令開始與結束時iperf的時間戳記以方便後續統計結果。在本範例中大約花費30秒傳送所有eMBMS流量

```
user@Lab01epc:~$ dd if=/dev/zero bs=1M count=2000 | socat -u - UDP-DATAGRAM:239.
255.1.1:3456
2000+0 records in
2000+0 records out
2097152000 bytes (2.1 GB, 2.0 GiB) copied, 29.0045 s, 72.3 MB/s
user@Lab01epc:~$
```

Step6-3 觀測eMBMS的影響(結果)

本範例的測試結果如下圖所示，eMBMS流量於時間戳記30sec時開始，大約於60sec時結束，計算平均流量如下表所示。可以觀察到由於MCH訊框多少會占用PHY層的無線電資源，因此正常的Downlink與Uplink通道會被eMBMS所影響。



```
[  5][TX-C]   5.00-10.00  sec  1.49 MBytes  2.50 Mbits/sec   15   97.6 KBytes
[  7][RX-C]   5.00-10.00  sec  1.81 MBytes  3.03 Mbits/sec
[  5][TX-C]  10.00-15.00  sec  1.49 MBytes  2.50 Mbits/sec    0    160 KBytes
[  7][RX-C]  10.00-15.00  sec  1.90 MBytes  3.18 Mbits/sec
[  5][TX-C]  15.00-20.00  sec  2.11 MBytes  3.54 Mbits/sec   66    133 KBytes
[  7][RX-C]  15.00-20.00  sec  1.80 MBytes  3.02 Mbits/sec
[  5][TX-C]  20.00-25.00  sec  1.74 MBytes  2.92 Mbits/sec   12    134 KBytes
[  7][RX-C]  20.00-25.00  sec  1.87 MBytes  3.14 Mbits/sec
[  5][TX-C]  25.00-30.00  sec  2.17 MBytes  3.65 Mbits/sec    0    161 KBytes
[  7][RX-C]  25.00-30.00  sec  1.18 MBytes  1.97 Mbits/sec
[  5][TX-C]  30.00-35.00  sec  1.30 MBytes  2.19 Mbits/sec   14    116 KBytes
[  7][RX-C]  30.00-35.00  sec  1.39 MBytes  2.33 Mbits/sec
[  5][TX-C]  35.00-40.00  sec  1.30 MBytes  2.19 Mbits/sec    0    136 KBytes
[  7][RX-C]  35.00-40.00  sec  1.54 MBytes  2.58 Mbits/sec
[  5][TX-C]  40.00-45.00  sec  1.30 MBytes  2.19 Mbits/sec   35    163 KBytes
[  7][RX-C]  40.00-45.00  sec  1.37 MBytes  2.30 Mbits/sec
[  5][TX-C]  45.00-50.00  sec  1.74 MBytes  2.92 Mbits/sec   29    151 KBytes
[  7][RX-C]  45.00-50.00  sec  1.54 MBytes  2.58 Mbits/sec
[  5][TX-C]  50.00-55.00  sec  1.30 MBytes  2.19 Mbits/sec   12    151 KBytes
[  7][RX-C]  50.00-55.00  sec  1.45 MBytes  2.43 Mbits/sec
[  5][TX-C]  55.00-60.00  sec  1.30 MBytes  2.19 Mbits/sec    8    150 KBytes
[  7][RX-C]  55.00-60.00  sec  1.50 MBytes  2.52 Mbits/sec
[  5][TX-C]  60.00-65.00  sec  1.74 MBytes  2.92 Mbits/sec   13    126 KBytes
[  7][RX-C]  60.00-65.00  sec  1.84 MBytes  3.08 Mbits/sec
[  5][TX-C]  65.00-70.00  sec  1.80 MBytes  3.02 Mbits/sec    6    127 KBytes
[  7][RX-C]  65.00-70.00  sec  1.77 MBytes  2.97 Mbits/sec
[  5][TX-C]  70.00-75.00  sec  1.74 MBytes  2.92 Mbits/sec   15    126 KBytes
[  7][RX-C]  70.00-75.00  sec  1.91 MBytes  3.21 Mbits/sec
```

Time	Avg. Downlink	Avg. Uplink
5-30	2.868Mbps	3.022Mbps
30-60	2.457Mbps	2.312Mbps
60-75	3.087Mbps	2.953Mbps

Stage 3 Check List

	項目	內容
	建立Named Pipe	確認以建立一Named Pipe供UE存放其MAC層的PDU
	將UE MAC PDU引導至Named Pipe	確認設定檔中有將UE MAC層PDU引導至Named Pipe
	Wireshark的封包解析器	確認有新增一DLT=147的mac-lte-framed封包解析器
	MIB	確認可以觀察到MIB且其內容正確
	SIB	確認可以觀察到SIB1、SIB2和SIB13且其內容正確
	MCH訊框	確認可以觀察到定期發送的MCH訊框
	MCCH	確認該定期發送的MCH訊框對應Logical Channel為MCCH
	MTCH	確認進行eMBMS Multicast時會有資料透過MTCH傳送

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 環境設定
- Stage 2. eMBMS 啟動與測試
- Stage 3. eMBMS 觀測
- Stage 4. Physical Layer參數調整
  - Step1 原始Radio Frame觀察
  - Step2 MCH分配周期調整
  - Step3 MCH分配偏移調整
  - Step4 MCH SF分配調整
  - Step5 MCCH重複周期調整
  - Step6 MCCH偏移調整
  - Step7 MCCH SF分配調整
- Stage 5. eMBMS應用
- 總結及問題

Step1-1 原始Radio Frame觀察(設定)

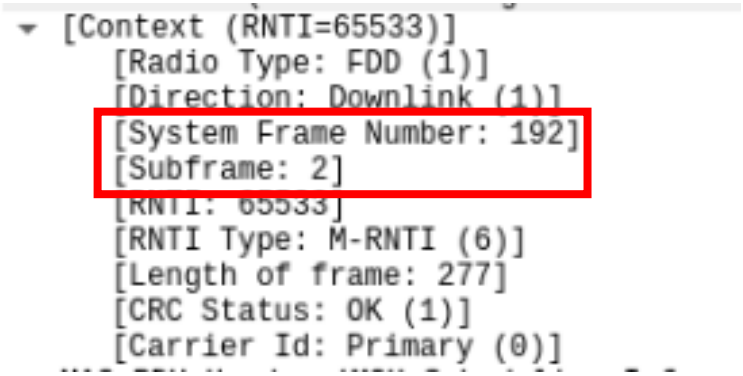
本階段所會動到的設定檔主要為/etc/srslte/sib.conf.mbsfn下的sib2的mbsfnSubframeConfigList以及sib13，其原始設定如下兩張圖所示，若後續沒有特別說明須更動的設定項目請按照此圖將其改為預設值。

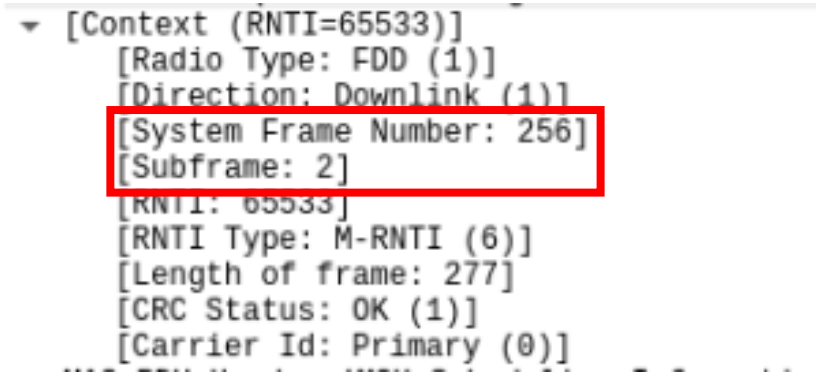
```
mbsfnSubframeConfigList =
{
    radioframeAllocationPeriod = 1;
    subframeAllocationNumFrames = 1;
    radioframeAllocationOffset = 0;
    subframeAllocation = 63;
};
```

```
sib13 =
{
    mbsfn_notification_config =
    {
        mbsfn_notification_repetition_coeff = "n2";
        mbsfn_notification_offset = 0;
        mbsfn_notification_sf_index = 1;
    };
    mbsfn_area_info_list_size = 1;
    mbsfn_area_info_list =
    {
        non_mbsfn_region_length = "s2";
        mcch_repetition_period = "rf64";
        mcch_modification_period = "rf512";
        signalling_mcs = "n2";
        mbsfn_area_id = 1;
        notification_indicator = 0;
        mcch_offset = 0;
        sf_alloc_info = 32;
    };
};
```

Step1-2 原始Radio Frame觀察(MCCH)

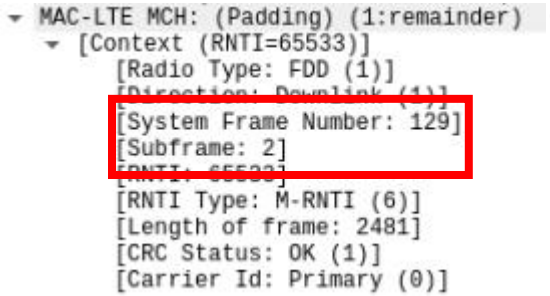
直接以Wireshark觀察UE MAC層的PDU可以發現預設設定值的MCCH每隔64個Radio Frame就會出現(SFN mod 64 =0)且其所出現的Sub-Frame都是Sub-Frame#2(Wireshark的Sub-Frame編號從1開始)

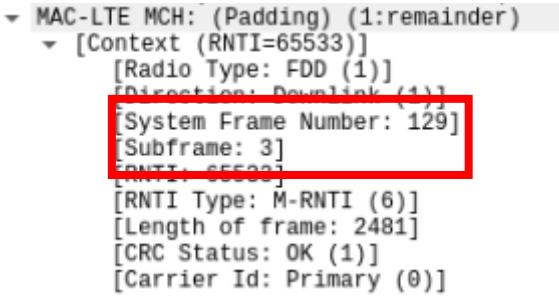


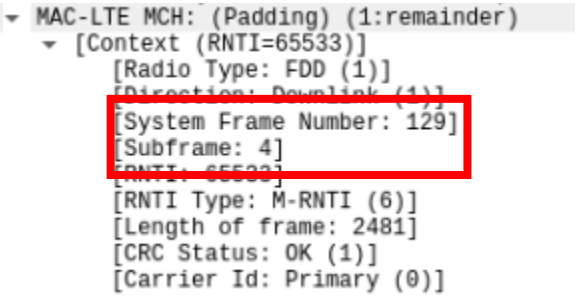


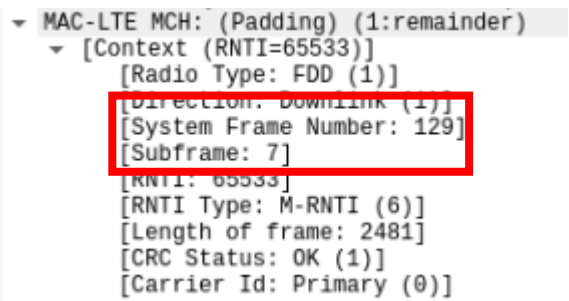
Step1-3 原始Radio Frame觀察(MTCH)

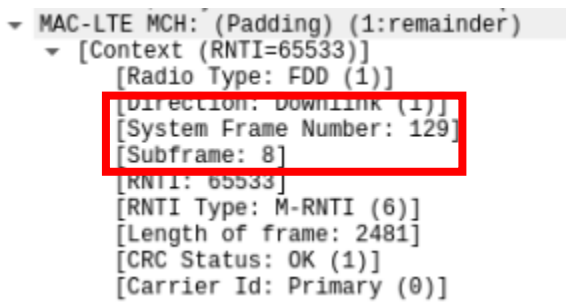
在MBMS-GW上執行dd if=/dev/zero bs=1M count=10 | socat -u - UDP-DATAGRAM:239.255.1.1:3456產生MCTH並以Wireshark觀察UE MAC層的PDU可以發現預設設定值的MTCH每個Radio Frame都出現(SFN mod 1 =0)且其所出現的Sub-Frame為#2、#3、#4、#7、#8及#9

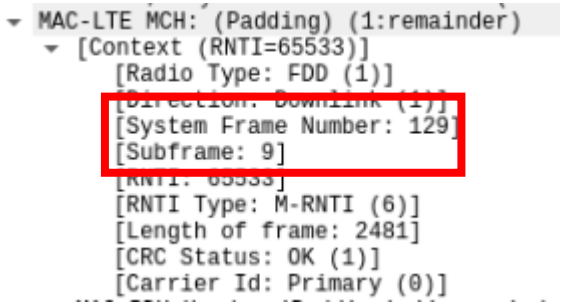












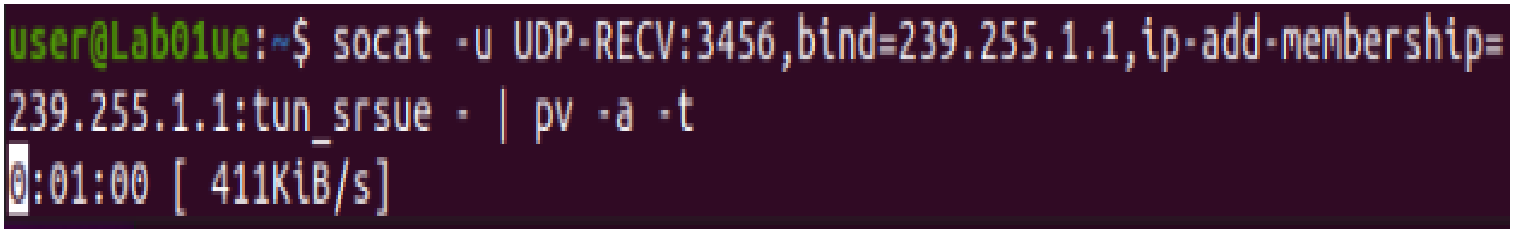
Step1-4 原始Radio Frame觀察(測試)

在MBMS-GW上執行pv -L 1M -r /dev/zero | socat -u - UDP-DATAGRAM:239.255.1.1:3456，此指令會從/dev/zero穩定的產生1MB/sec(8Mbps)的流量並使用eMBMS進行傳播，此指令在本階段需全程開啟以利後續的測試

```
user@Lab01epc:~$ pv -L 1M -r /dev/zero | socat -u - UDP-DATAGRAM:239.255.1.1:345
6
[1.00MiB/s]
```

Step1-5 原始Radio Frame觀察(流通量)

在UE上執行socat -u UDP-RECV:3456,bind=239.255.1.1,ip-add-membership=239.255.1.1:tun_srsue - | pv -a -t，此指令會接收來自MBMS-GW的eMBMS測試流量並顯示出平均流通量，執行此指令約一分鐘後即可按Ctrl-C結束本命令，所得數值即為從開始到結束為止的平均流通量。原始設定檔的流通量測試結果為441KB/sec (3528Kbps)



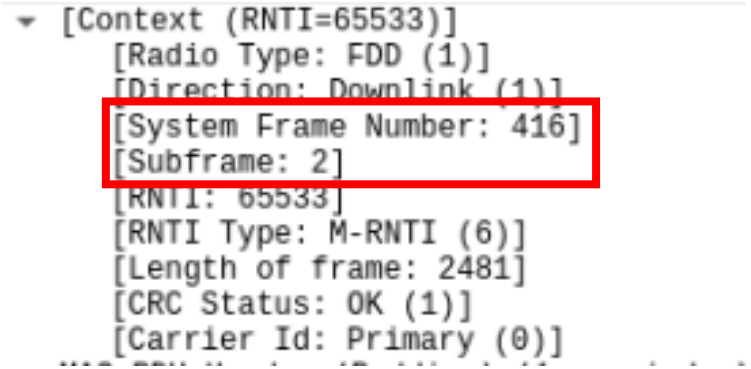
Step2-1 MCH分配週期調整(設定)

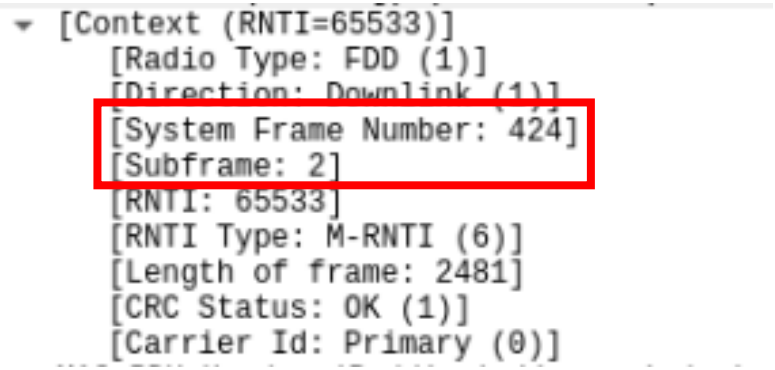
在eNB上輸入sudo vim /etc/srslte/sib.conf.mbsfn以編輯SIB的設定檔，將sib2的mbsfnSubframeConfigList內的radioframeAllocationPeriod 由1改為8，改完的設定檔如下圖所示，改完後請重啟eNB及UE

```
mbsfnSubframeConfigList =
{
    radioframeAllocationPeriod = 8;
    subframeAllocationNumFrames = 1;
    radioframeAllocationOffset = 0;
    subframeAllocation = 63;
};
```

Step2-2 MCH分配週期調整(結果)

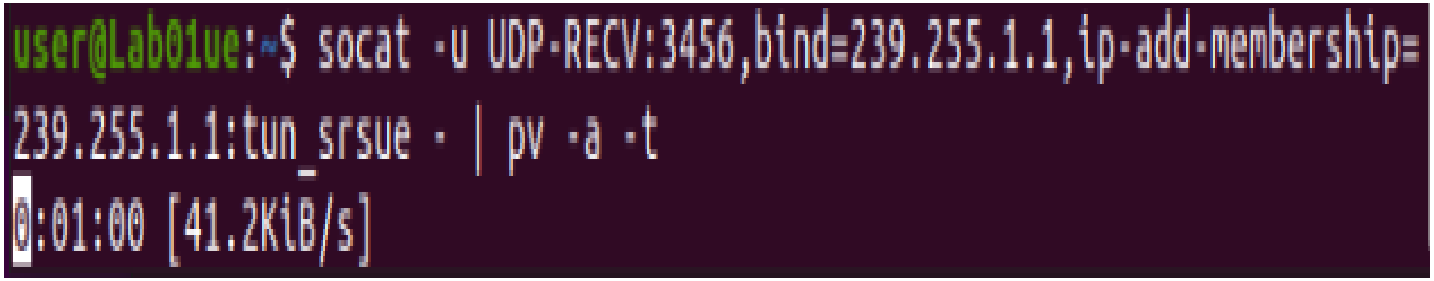
以Wireshark觀察UE MAC層的PDU可以發現MCH的出現頻率從原始設定檔的一個Radio Frame出現一次降為八個Radio Frame才出現一次





Step2-3 MCH分配週期調整(流通量)

在UE上執行socat -u UDP-RECV:3456,bind=239.255.1.1,ip-add-membership=239.255.1.1:tun_srsue - | pv -a -t，此指令會接收來自MBMS-GW的eMBMS測試流量並顯示出平均流通量，執行此指令約一分鐘後即可按Ctrl-C結束本命令，所得數值即為從開始到結束為止的平均流通量。將radioframeAllocationPeriod由1改為8的流通量測試結果為41.2KB/sec (329.6Kbps)，為原本的0.093倍



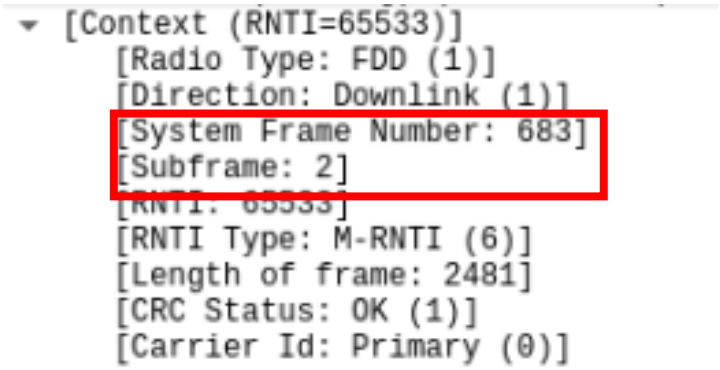
Step3-1 MCH分配偏移調整(設定)

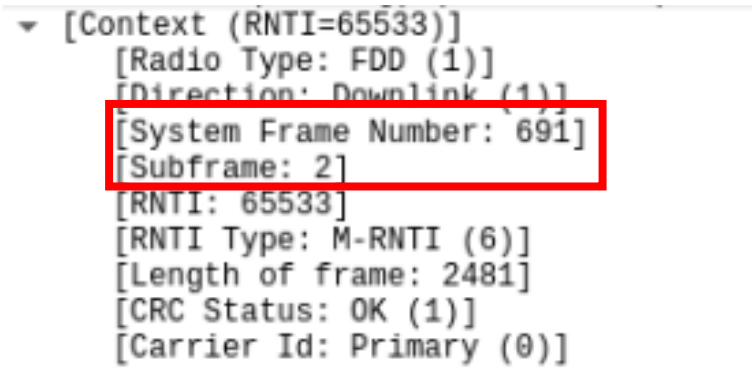
在eNB上輸入sudo vim /etc/srslte/sib.conf.mbsfn以編輯SIB的設定檔，將sib2的mbsfnSubframeConfigList內的radioframeAllocationPeriod 由1改為8並同時將radioframeAllocationOffset由0改為3，改完的設定檔如下圖所示，改完後請重啟eNB及UE

```
mbsfnSubframeConfigList =
{
    radioframeAllocationPeriod = 8;
    subframeAllocationNumFrames = 1;
    radioframeAllocationOffset = 3;
    subframeAllocation = 63;
};
```

Step3-2 MCH分配偏移調整(結果)

以Wireshark觀察UE MAC層的PDU可以發現MCH的出現頻率從原始設定檔的一個Radio Frame出現一次降為八個Radio Frame才出現一次外，原本要SFN mod 8 = 0 才會被分配的MCH改為SFN mod 8 = 3 才會被分配

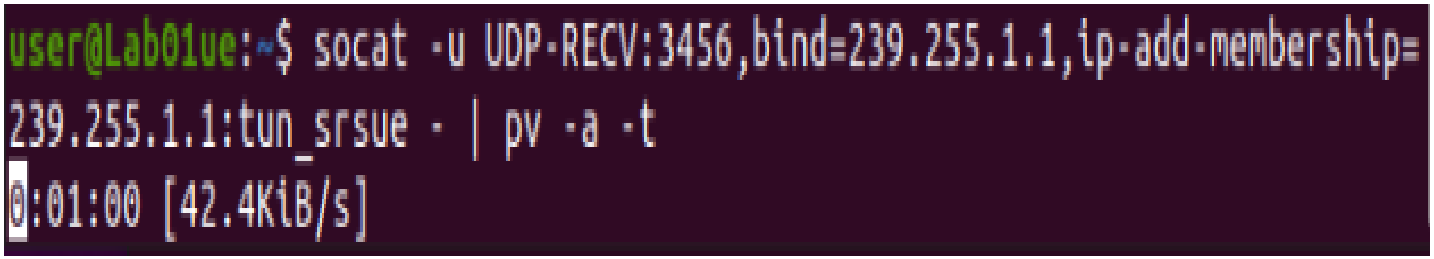




Step3-3 MCH分配偏移調整(流通量)

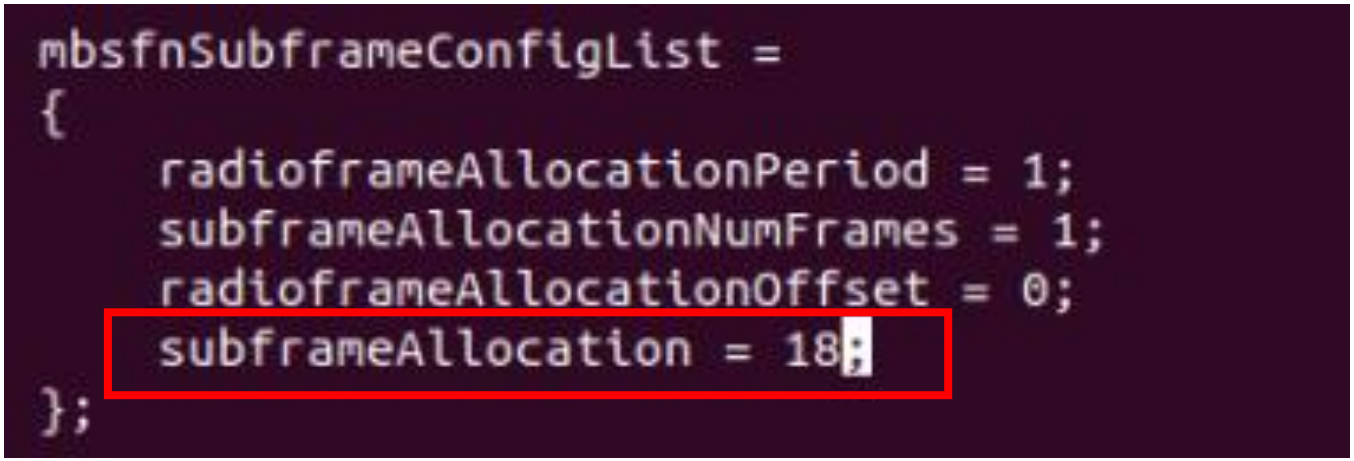
在UE上執行socat -u UDP-RECV:3456,bind=239.255.1.1,ip-add-membership=239.255.1.1:tun_srsue - | pv -a -t，執行此指令約一分鐘後即可按Ctrl-C結束本命令，所得數值即為從開始到結束為止的平均流通量

將radioframeAllocationOffset由0改為3的流通量測試結果為42.4KB/sec (339.2Kbps)，為原本的0.096倍，並與單純改動radioframeAllocationPeriod的結果相差無幾



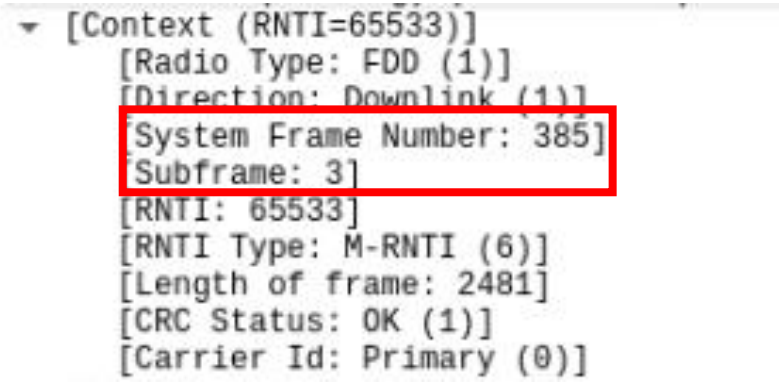
Step4-1 MCH SF分配調整(設定)

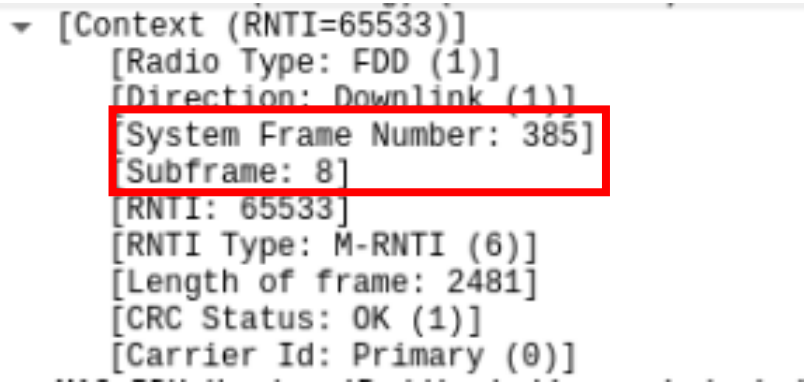
在eNB上輸入sudo vim /etc/srslte/sib.conf.mbsfn以編輯SIB的設定檔，將sib2的mbsfnSubframeConfigList內的subframeAllocation 由63改為18(0b010010)，改完的設定檔如下圖所示，改完後請重啟eNB及UE



Step4-2 MCH SF分配調整(結果)

以Wireshark觀察UE MAC層的PDU可以發現原本會出現在Sub-Frame #2、#3、#4、#7、#8及#9的MCH訊框變成只會出現在Sub-Frame #3和#8，這符合剛剛的設定值0b010010

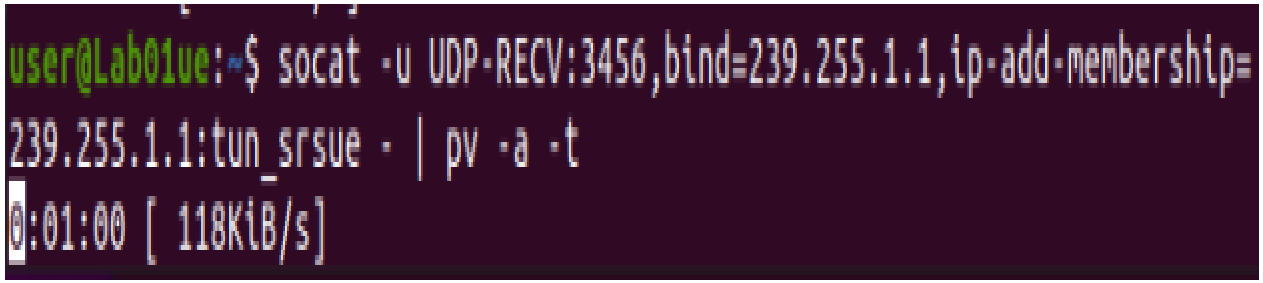




Step4-3 MCH SF分配調整(流通量)

在UE上執行socat -u UDP-RECV:3456,bind=239.255.1.1,ip-add-membership=239.255.1.1:tun_srsue - | pv -a -t，執行此指令約一分鐘後即可按Ctrl-C結束本命令，所得數值即為從開始到結束為止的平均流通量

將subframeAllocation由63改為18的流通量測試結果為118KB/sec (944Kbps)，為原本的0.2676倍，推測為Sub-Frame的數量只有原始設定1/3的緣故



Step5-1 MCCH 重複週期調整(設定)

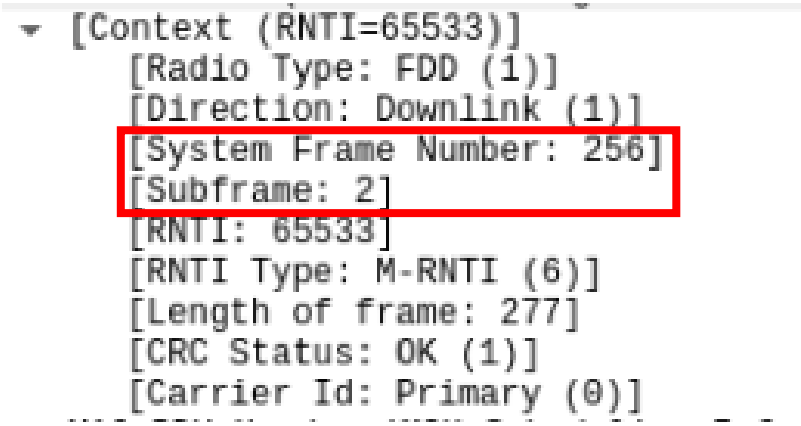
在eNB上輸入sudo vim /etc/srslte/sib.conf.mbsfn以編輯SIB的設定檔，將sib13內的mcch_repetition_period 由rf64改為rf128，而sib2的設定維持原始設定檔的模樣，改完的設定檔如下圖所示，改完後請重啟eNB及UE

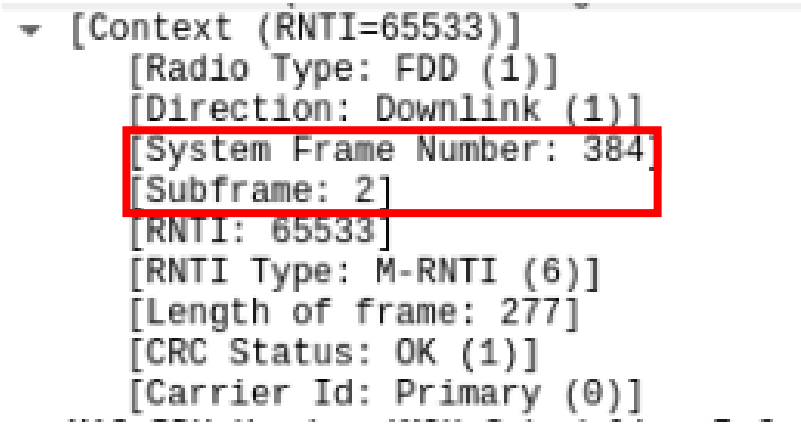
```
mbsfnSubframeConfigList =
{
    radioframeAllocationPeriod = 1;
    subframeAllocationNumFrames = 1;
    radioframeAllocationOffset = 0;
    subframeAllocation = 63;
};
```

```
sib13 =
{
    mbsfn_notification_config =
    {
        mbsfn_notification_repetition_coeff = "n2";
        mbsfn_notification_offset = 0;
        mbsfn_notification_sf_index = 1;
    };
    mbsfn_area_info_list_size = 1;
    mbsfn_area_info_list =
    {
        non_mbsfn_region_length = "s2";
        mcch_repetition_period = "rf128";
        mcch_modification_period = "rf512";
        signalling_mcs = "n2";
        mbsfn_area_id = 1;
        notification_indicator = 0;
        mcch_offset = 0;
        sf_alloc_info = 32;
    };
};
```

Step5-2 MCCH 重複週期調整(結果)

以Wireshark觀察UE MAC層的PDU可以發現原本每64個Radio Frame才會出現一次的MCCH，其出現頻率降低到每128個Radio Frame才會出現一次，符合剛剛的設定

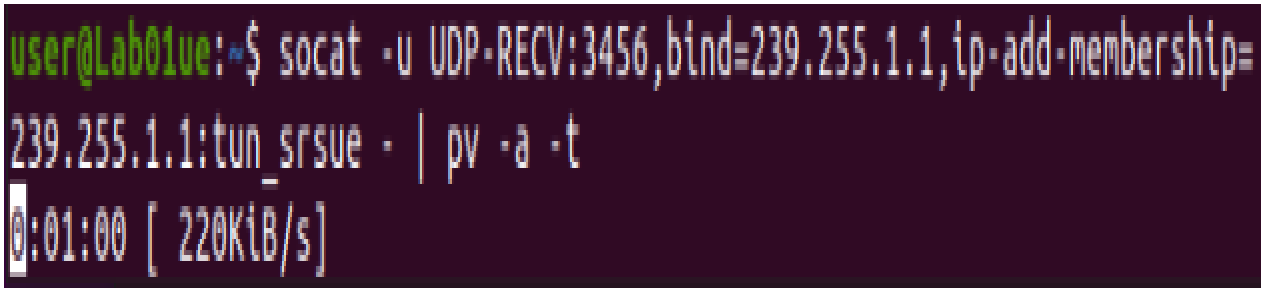




Step5-3 MCCH 重複週期調整(流通量)

在UE上執行socat -u UDP-RECV:3456,bind=239.255.1.1,ip-add-membership=239.255.1.1:tun_srsue - | pv -a -t，執行此指令約一分鐘後即可按Ctrl-C結束本命令，所得數值即為從開始到結束為止的平均流通量

將mcch_repetition_period由rf64改為rf128的流通量測試結果為220KB/sec (1760Kbps)，為原本的0.499倍，推測為MCCH的重複週期為只有原始設定兩倍的緣故



Step6-1 MCCH 偏移調整(設定)

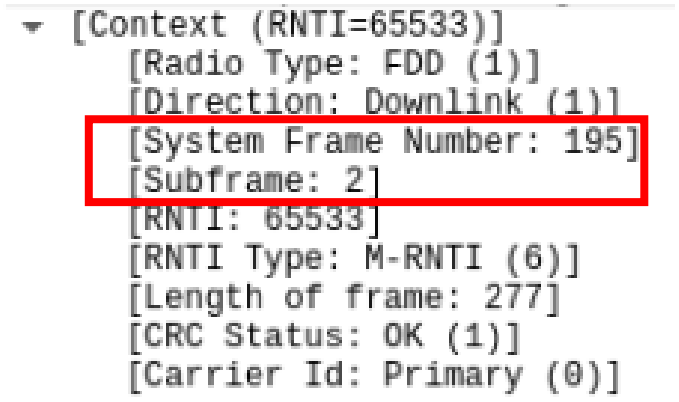
在eNB上輸入sudo vim /etc/srslte/sib.conf.mbsfn以編輯SIB的設定檔，將sib2的mbsfnSubframeConfigList內的radioframeAllocationOffset由0改為3並同時將sib13內的mcch_offset由0改為3，改完的設定檔如下圖所示，改完後請重啟eNB及UE

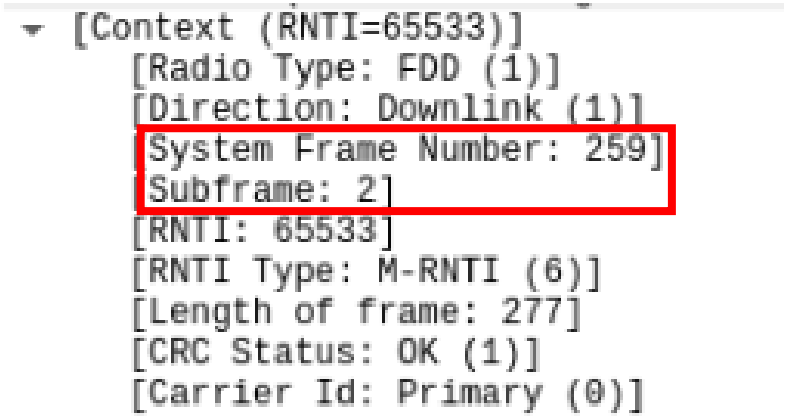
```
mbsfnSubframeConfigList =
{
    radioframeAllocationPeriod = 1;
    subframeAllocationNumFrames = 1;
    radioframeAllocationOffset = 3;
    subframeAllocation = 63;
};
```

```
sib13 =
{
    mbsfn_notification_config =
    {
        mbsfn_notification_repetition_coeff = "n2";
        mbsfn_notification_offset = 0;
        mbsfn_notification_sf_index = 1;
    };
    mbsfn_area_info_list_size = 1;
    mbsfn_area_info_list =
    {
        non_mbsfn_region_length = "s2";
        mcch_repetition_period = "rf64";
        mcch_modification_period = "rf512";
        signalling_mcs = "n2";
        mbsfn_area_id = 1;
        notification_indicator = 0;
        mcch_offset = 3;
        sf_alloc_info = 32;
    };
};
```

Step6-2 MCCH 偏移調整(結果)

以 Wireshark 觀察 UE MAC 層的 PDU 可以發現原本 SFN mod 64 = 0 才會進行排程的MCCH變成SFN mod 64=3 才會被排程，符合剛才的設定

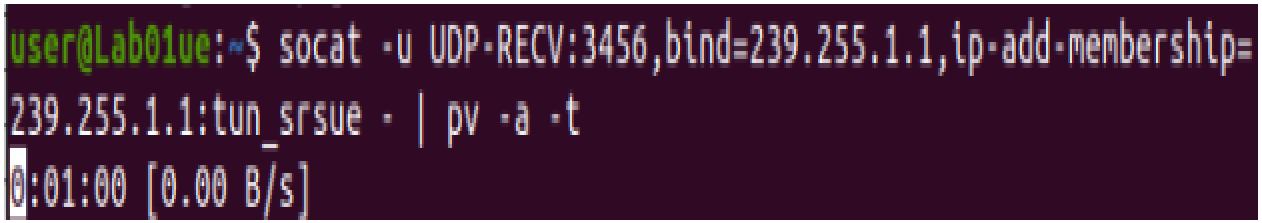




Step6-3 MCCH 偏移調整(流通量)

在 UE 上 執 行 socat -u UDP-RECV:3456,bind=239.255.1.1,ip-add-membership=239.255.1.1:tun_srsue - | pv -a -t，執行此指令約一分鐘後即可按Ctrl-C結束本命令，所得數值即為從開始到結束為止的平均流通量

將MCCH偏移量改為3的流通量測試結果為0B/sec，推測是MCCH雖然可以順利進行排程但是MCH無法找到符合排程條件SFN mod 1 = 3 的Radio Frame，因此MTCH便無法進行傳輸，當然就不會有量通量



Step7-1 MCCH SF分配調整(設定)

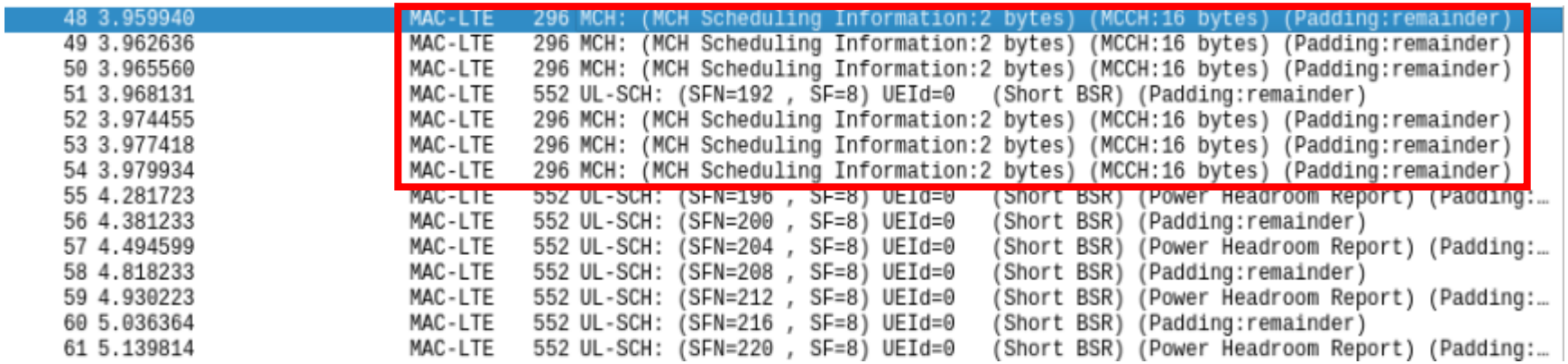
在eNB上輸入sudo vim /etc/srslte/sib.conf.mbsfn以編輯SIB的設定檔，將sib13內的sf_alloc_info由32改為63，改完的設定檔如下圖所示，改完後請重啟eNB及UE

```
mbsfnSubframeConfigList =
{
    radioframeAllocationPeriod = 1;
    subframeAllocationNumFrames = 1;
    radioframeAllocationOffset = 0;
    subframeAllocation = 63;
};
```

```
sib13 =
{
    mbsfn_notification_config =
    {
        mbsfn_notification_repetition_coeff = "n2";
        mbsfn_notification_offset = 0;
        mbsfn_notification_sf_index = 1;
    };
    mbsfn_area_info_list_size = 1;
    mbsfn_area_info_list =
    {
        non_mbsfn_region_length = "s2";
        mcch_repetition_period = "rf64";
        mcch_modification_period = "rf512";
        signalling_mcs = "n2";
        mbsfn_area_id = 1;
        notification_indicator = 0;
        mcch_offset = 0;
        sf_alloc_info = 63;
    };
};
```

Step7-2 MCCH SF分配調整(結果)

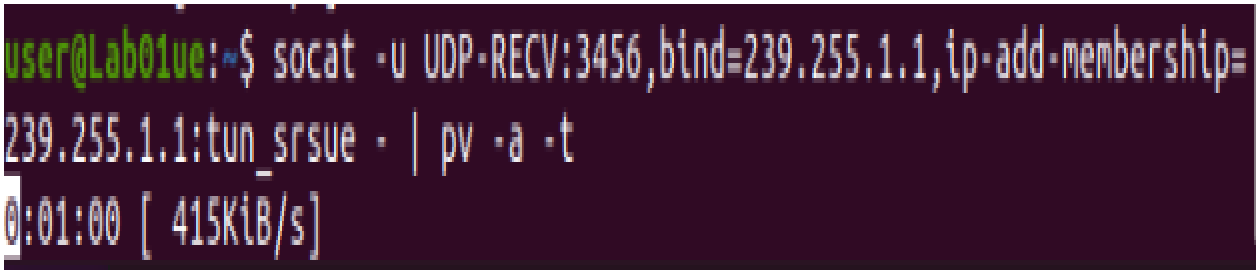
以 Wireshark 觀 察 UE MAC 層 的 PDU 可 以 發 現 原 本 只會分配在Sub-Frame #2的MCCH變成分配在Sub-Frame #2、#3、#4、#7、#8及#9上，符合剛才的設定



Step7-3 MCCH SF分配調整(流通量)

在UE上執行socat -u UDP-RECV:3456,bind=239.255.1.1,ip-add-membership=239.255.1.1:tun_srsue - | pv -a -t，執行此指令約一分鐘後即可按Ctrl-C結束本命令，所得數值即為從開始到結束為止的平均流通量

將sf_alloc_info由32改為63的流通量測試結果為415KB/sec(3320Kbps)，為原本的0.941倍，可以得知MCCH若占用太多Sub-Frame的話不但不會提升流通量，甚至會降低流通量



Stage 4 Check List

	項目	內容
	原始Radio Frame	確認是否觀察到原始設定內MCH及MCCH、MTCH的排程情形
	MCH分配周期	確認是否完成調整MCH分配週期並觀察UE MAC PDU及Throughput變化
	MCH分配偏移	確認是否完成調整MCH分配偏移並觀察UE MAC PDU及Throughput變化
	MCH SF分配	確認是否完成調整MCH SF分配並觀察UE MAC PDU及Throughput變化
	MCCH重複周期	確認是否完成調整MCCH重複周期並觀察UE MAC PDU及Throughput變化
	MCCH偏移	確認是否完成調整MCCH偏移並觀察UE MAC PDU及Throughput變化
	MCCH SF分配	確認是否完成調整MCCH SF分配並觀察UE MAC PDU及Throughput變化

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 環境設定
- Stage 2. eMBMS 啟動與測試
- Stage 3. eMBMS 觀測
- Stage 4. Physical Layer參數調整
- Stage 5. eMBMS應用
  - Step1 調整SIB設定
  - Step2 FFmpeg 串流影片檔
  - Step3 FFplay 接收
- 總結及問題

Step1 調整SIB設定

在開始本階段的步驟前，請先將前一階段的SIB設定檔還原為原始設定，sib2及sib13的原始設定如下圖所示

```
mbsfnSubframeConfigList =
{
    radioframeAllocationPeriod = 1;
    subframeAllocationNumFrames = 1;
    radioframeAllocationOffset = 0;
    subframeAllocation = 63;
};
```

```
sib13 =
{
    mbsfn_notification_config =
    {
        mbsfn_notification_repetition_coeff = "n2";
        mbsfn_notification_offset = 0;
        mbsfn_notification_sf_index = 1;
    };
    mbsfn_area_info_list_size = 1;
    mbsfn_area_info_list =
    {
        non_mbsfn_region_length = "s2";
        mcch_repetition_period = "rf64";
        mcch_modification_period = "rf512";
        signalling_mcs = "n2";
        mbsfn_area_id = 1;
        notification_indicator = 0;
        mcch_offset = 0;
        sf_alloc_info = 32;
    };
};
```

Step2 FFmpeg串流影片檔

請先確認前一階段位於MBMS-GW上的流量產生器已關閉，若已關閉則請在MBMS-GW的終端機輸入 ffmpeg -re -i ‘your_video_file’ -vcodec libx264 -preset ultrafast -tune zerolatency -f mpegts -b:v 1M udp://239.255.1.1:1234 其中‘your_video_file’為預先準備好之影片檔路徑。

本指令參數說明如下:

- -re : 讓ffmpeg以原始FPS進行串流，否則會一下就播完
- -i : 指定輸入影片檔
- -vcodec libx264：因不確定來源影片原始編碼格式，因此一律使用 libx264，並同時設定libx264的preset與tune
- -f mpegts : 指定以MPEG Transport Stream格式進行串流
- -b:v 1M : 限制輸出平均 bitrate 在 1Mbit/s
- udp://239.255.1.1:1234 : 欲串流的Multicast Group

FFmpeg串流影片檔結果

輸入上述指令後FFmpeg會立即進行Multicast串流

```
[libx264 @ 0x5633e877a400] using SAR=133/100
[libx264 @ 0x5633e877a400] using cpu capabilities: MMX2 SSE2 SSE3 Cache64
[libx264 @ 0x5633e877a400] profile Constrained Baseline, level 4.0
Output #0, mpegts, to 'udp://239.255.1.1:1234':
  Metadata:
    WMFSDKNeeded    : 0.0.0.0000
    DeviceConformanceTemplate: MP@HL
    WMFSDKVersion   : 12.0.17763.592
    IsVBR           : 1
    VBR Peak        : 22125023
    Buffer Average  : 81344
    encoder         : Lavf58.29.100
    Stream #0:0(eng): Video: h264 (libx264), yuv420p(progressive), 1440x1080 [SA
R 133:100 DAR 133:75], q=-1--1, 1000 kb/s, 29.97 fps, 90k tbn, 29.97 tbc
    Metadata:
      encoder         : Lavc58.54.100 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/1000000 buffer size: 0 vbv_delay: -1
    Stream #0:1(eng): Audio: mp2, 44100 Hz, stereo, s16, 384 kb/s
    Metadata:
      encoder         : Lavc58.54.100 mp2
frame=    1 fps=0.0 q=41.0 size=      19kB time=00:00:00.61 bitrate= 256.3kbits/
frame=   14 fps= 14 q=26.0 size=      73kB time=00:00:00.61 bitrate= 966.8kbits/
s speed= 0.6x
```

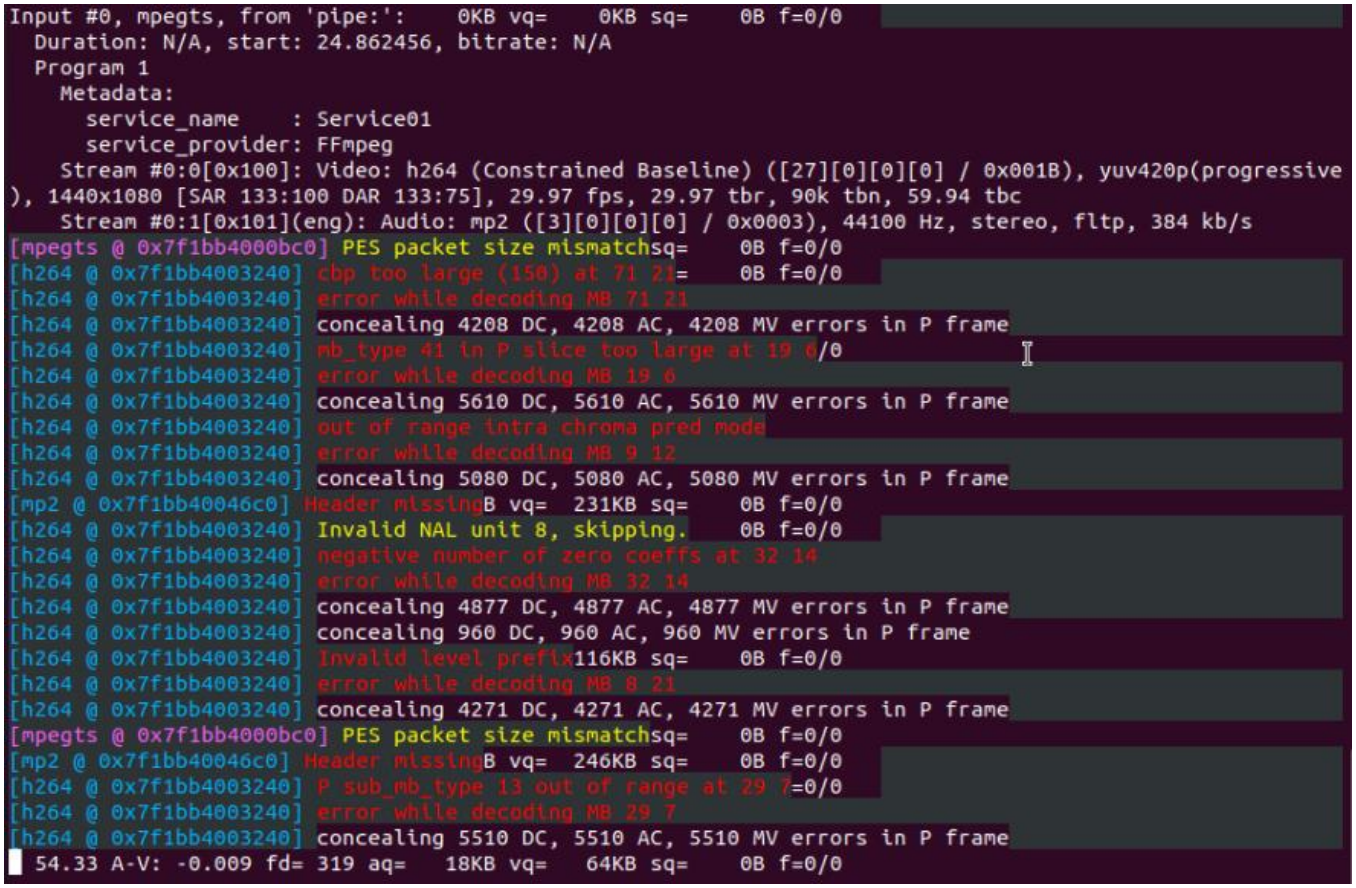
Step3 FFplay接收

在UE上輸入socat UDP-RECV:1234,bind=239.255.1.1,ip-add-membership=239.255.1.1:tun_srsue - | ffplay - 接收從MBMS-GW串流的影片檔

```
user@Lab01ue:~$ socat -u UDP-RECV:1234,bind=239.255.1.1,ip-add-membership=239.255.1.1:tun_srsue "exec:ff
play -"
ffplay version 4.2.2-1ubuntu1 Copyright (c) 2003-2019 the FFmpeg developers
  built with gcc 9 (Ubuntu 9.3.0-3ubuntu1)
  configuration: --prefix=/usr --extra-version=1ubuntu1 --toolchain=hardened --libdir=/usr/lib/x86_64-li
nux-gnu --incdir=/usr/include/x86_64-linux-gnu --arch=amd64 --enable-gpl --disable-stripping --enable-av
resample --disable-filter=resample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libaom --e
nable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libcodec2 --
enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgme --enable
-libgsm --enable-libjack --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt
 --enable-libopus --enable-libpulse --enable-librsvg --enable-librubberband --enable-libshine --enable-l
ibsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --ena
ble-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx265
--enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --enable-omx --enable-op
enal --enable-opencl --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libiec618
83 --enable-nvenc --enable-chromaprint --enable-frei0r --enable-libx264 --enable-shared
  libavutil      56. 31.100 / 56. 31.100
  libavcodec     58. 54.100 / 58. 54.100
  libavformat    58. 29.100 / 58. 29.100
  libavdevice    58.  8.100 / 58.  8.100
  libavfilter     7. 57.100 /  7. 57.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  5.100 /  5.  5.100
  libswresample   3.  5.100 /  3.  5.100
  libpostproc    55.  5.100 / 55.  5.100
[h264 @ 0x7f1bb400e040] non-existing PPS 0 referenced    0B f=0/0
    Last message repeated 1 times
[h264 @ 0x7f1bb400e040] decode_slice_header error
[h264 @ 0x7f1bb400e040] no frame!
[h264 @ 0x7f1bb400e040] non-existing PPS 0 referenced    0B f=0/0
    Last message repeated 1 times
[h264 @ 0x7f1bb400e040] decode_slice_header error
[h264 @ 0x7f1bb400e040] no frame!
[h264 @ 0x7f1bb400e040] non-existing PPS 0 referenced
```

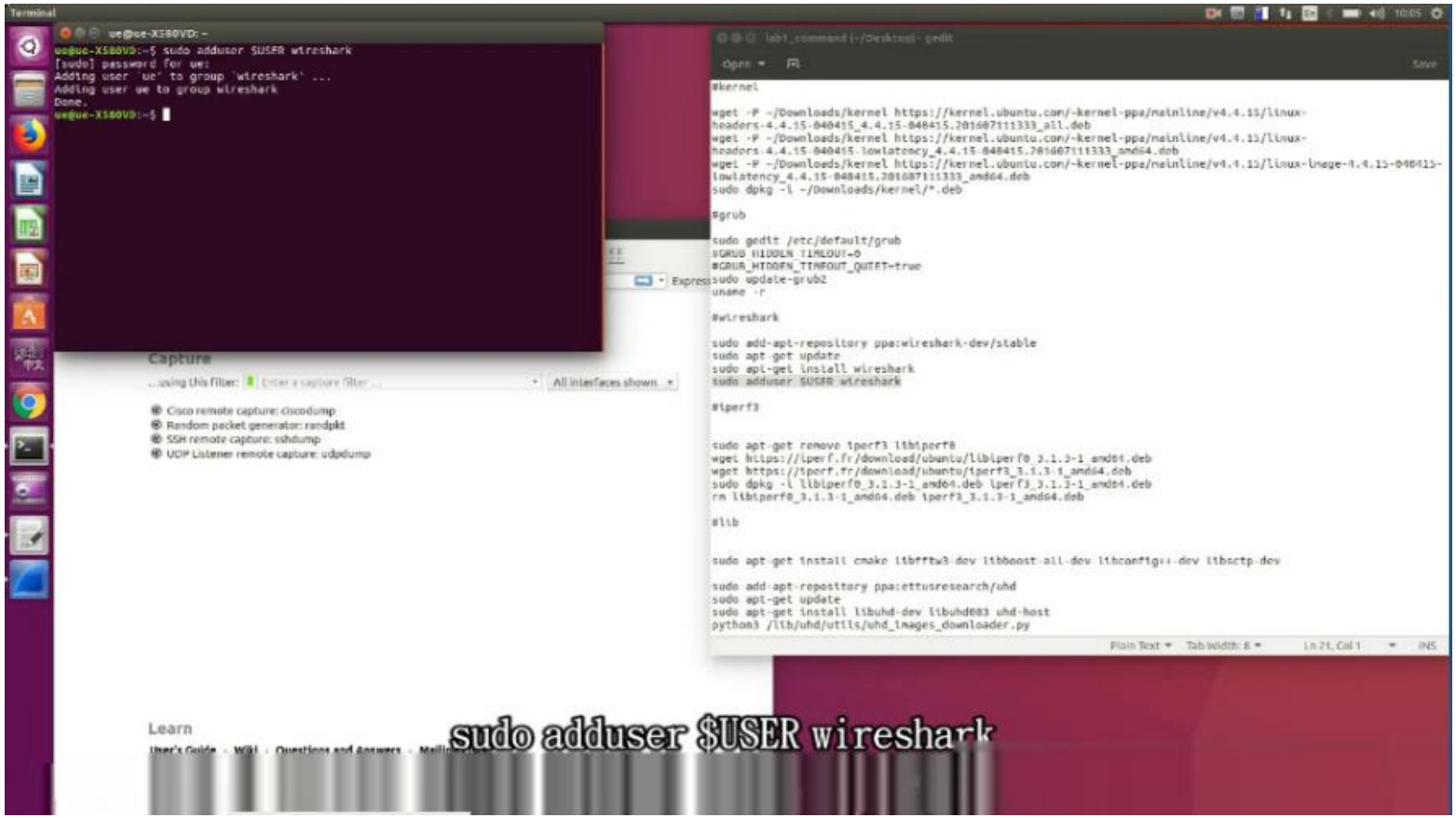
FFplay接收結果-1

若出現大量錯誤訊息為正常現象，因須等MPEG-TS順利同步後才能正確解碼影片，請稍待一會



FFplay接收結果-2

等待一小段時間後FFplay應可正確撥放影片，若影片撥放過程中有任何模糊現象均屬以H.264串流的正常結果



Stage 5 Check List

	項目	內容
	SIB設定	確認有將SIB還原至原始設定
	FFmpeg串流	確認FFmpeg能順利進行影片的Multicast串流
	FFplay接收	確認UE端的FFplay能順利接收到來自MBMS-GW的串流

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 環境設定
- Stage 2. eMBMS 啟動與測試
- Stage 3. eMBMS 觀測
- Stage 4. Physical Layer參數調整
- Stage 5. eMBMS應用
- 總結及問題

總結

- 了解srsLTE在如何進行eMBMS的設定
- 了解如何對eMBMS的封包進行觀測
  - 實際操作SGI_MB上的封包觀測
  - 實際操作M1上的封包觀測
- 實際觀測UE MAC PDU進一步了解eMBMS的資源分配
  - 了解Transport Channel MCH的排程機制
  - 了解Logical Channel MCCH及MTCH的排程機制
  - 了解MIB及SIB1、SIB2及SIB13的意義
- 實際調整SIB參數了解其對上層通道的影響
- 實際操作影片Multicast串流以學習eMBMS的應用

問題

請找出一SIB設定使得透過該SIB所設定的eMBMS Throughput較srsLTE範例設定設定高，並記錄:

- 所變更之SIB設定
- 變更此SIB設定會對Radio Frame及Sub-Frame造成什麼改變
- 變更後的Throughput為原本的幾倍
- 在沒有eMBMS流量時對正常Downlink及Uplink通道有何影響
- 在有eMBMS流量時對正常Downlink及Uplink通道有何影響