Section 1 **Modification Summary**

After the installation of Openairinterface, doing the following modification to the source code and recompile the base station.

Step ONE: Modify the PRB MCS allocation in ENodeB

1. Function Name: assign_max_mcs_min_rb
   File Name: pre_processor.c
   Operation: Patch - Add patches below the block

>>
>

When ul_total_buffer > 0, allcote 25PRB and MCS=20

UE_template->pre_assigned_mcs_ul=mcs;
UE_template->pre_allocated_rb_table_index_ul=rb_table_index;
UE_template->pre_allocated_nb_rb_ul= rb_table[rb_table_index];

//lrf 25PRB MCS=20

- rb_table_index = 15;
- UE_template->pre_assigned_mcs_ul=20;
- UE_template->pre_allocated_rb_table_index_ul=15;
- UE_template->pre_allocated_nb_rb_ul= rb_table[rb_table_index];

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

Step TWO. Fix the allocated subframe number

1. Function Name: **schedule_ulsch_rnti**
   File Name: eNB_scheduler_ulsch.c
   Operation: Patch
   allocate resource to UE only when (SR > 0 or Inactivity_timer overflow) AND subframe to be scheduled is 0

>>
>

- if(((UE_is_to_be_scheduled(module_idP,CC_id,UE_id,sched_subframe)>0)) || (round>0))
- **if(((**my_UE_is_to_be_scheduled**(**module_idP**,**CC_id**,**UE_id**,**sched_subframe**)>**0**)) || (**round**>**0**))**

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

2) File Name: **Proto.h**
Operation: **Add definition of the new function**

>>
>

//lrf

- uint8_t my_UE_is_to_be_scheduled(module_id_t module_idP,int CC_id,uint8_t UE_id,unsigned char sched_subframe);

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

3) File name: eNB_scheduler_primitives.c

Operation: Add realization of function of defined in Proto.h

Copy UE_is_to_be_scheduled, rename it and patch.

>>>
>

```
uint8_t my_UE_is_to_be_scheduled(module_id_t module_idP,int CC_id,uint8_t UE_id,unsigned char sched_subframe)
if ((UE_template->bsr_info[LCGID0]>0) ||
   (UE_template->bsr_info[LCGID1]>0) ||
   (UE_template->bsr_info[LCGID2]>0) ||
   (UE_template->bsr_info[LCGID3]>0) ||
   ((UE_template->ul_SR>0) &&
    (sched_subframe==0)) || // uplink scheduling request
   ((UE_sched_ctl->ul_inactivity_timer>20)&&
    (UE_sched_ctl->ul_scheduled==0)&&
    (sched_subframe==0))||  // every 2 frames when RRC_CONNECTED
   ((UE_sched_ctl->ul_inactivity_timer>10)&&
    (UE_sched_ctl->ul_scheduled==0)&&
    (mac_eNB_get_rrc_status(module_idP,UE_RNTI(module_idP,UE_id)) < RRC_CONNECTED)))
```

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

Step THREE: Fix the Cellid and C-RNTI (ignore for CTOS smartphone; Otherwise the connection will fail)

1) Cellid = 0
2) Fix C-RNTI = 2048

File Name: eNB_scheduler_RA.c

Function Name: void initiate_ra_proc

>>>
>

- RA_template[i].rnti = taus(); //RNIT of UE is set to Random Number HERE!

\+ RA_template[i].rnti = 2048; *//lrf*

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

Step FOUR: Inform the application about the current RLC SN

1. File Name: Rlc_um_segment.c

Function Name: rlc_um_segment_10()
Operation: Patch - Add patches below the block

>>
>

```
pdu_p->b1 = pdu_p->b1 | ((rlc_pP->vt_us >> 8) & 0x03);
pdu_p->b2 = rlc_pP->vt_us & 0xFF;
rlc_pP->vt_us = rlc_pP->vt_us+1;

//Lrf : update RLC SN
```

- **trace_rlc_sn(rlc_pP->vt_us);**

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

2) File Name: Name Opt.h
Operation: Add definiton of new function (trace_rlc_sn send the current sequence number to the user space)

>>
>

//lrf

- void trace_rlc_sn(uint16_t sn);

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

3) File Name: Probe.c
Operation: Add realization for the new function

>>
>

```
   /* Lrf: Trace  RLC SN*/
void trace_rlc_sn(uint16_t sn){

  ssize_t bytesSent;
  unsigned int frameOffset = 2;

  //printf("trace_rlc_sn\n");
  /* Send out the data over the UDP socket */
  bytesSent = sendto(g_RLC_SN_socksd,  (unsigned char*)&sn, 2, 0,
                   (const struct sockaddr *)&g_RLC_SN_serv_addr,
sizeof(g_RLC_SN_serv_addr));

  if (bytesSent != frameOffset) {
    printf("trace_rlc_sn sendto() failed (errno=%d)\n",errno);
    //exit(1);
```

```
    }
}
```

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

4) File Name Probe.c

Operation: Add the global variables that is used in trace_rlc_sn()

>>
>

//lrf

- **static int g_RLC_SN_socksd = -1;/* *UDP socket used for RLC SN update* */**
- **static struct sockaddr_in g_RLC_SN_serv_addr;**

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<<

5) File Name: Probe.c

Function Name: init_opt()

Operation: Patch - Add patches below the block

>>
>

```
g_serv_addr.sin_family = AF_INET;
g_serv_addr.sin_port = htons(in_port);
g_serv_addr.sin_addr.s_addr = inet_addr(in_ip);

/*lrf: init the socket for RLC SN update*/
```

- g_RLC_SN_socksd = socket(AF_INET, SOCK_DGRAM, 0);

- if(g_RLC_SN_socksd == -1) {
- printf("Create rlc sn socket failed (errno=%d)\n", errno);
- }
- g_RLC_SN_serv_addr.sin_family = AF_INET;
- g_RLC_SN_serv_addr.sin_port = htons(6666);
- g_RLC_SN_serv_addr.sin_addr.s_addr = inet_addr(in_ip);

```
printf("CREATING SN UPDATE SOCKET\n");
```

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

Step FIVE: Modify the PHR period

1. File Name: openair2/RRC/LITE/rrc_eNB.c

   Function Name:
rrc_eNB_generate_defaultRRCConnectionReconfiguration()

   Operation: Comment and Patch

>>
>

```
/*
mac_MainConfig->phr_Config->choice.setup.periodicPHR_Timer =
MAC_MainConfig__phr_Config__setup__periodicPHR_Timer_sf20; // sf20 = 20 subframes
```

mac_MainConfig->phr_Config->choice.setup.prohibitPHR_Timer =
MAC_MainConfig__phr_Config__setup__prohibitPHR_Timer_sf20; // sf20 = 20 subframes
***/**

- mac_MainConfig->phr_Config->choice.setup.periodicPHR_Timer =
  MAC_MainConfig__phr_Config__setup__periodicPHR_Timer_sf1000; // sf1000 = 1000 subframes //lrf

- mac_MainConfig->phr_Config->choice.setup.prohibitPHR_Timer = MAC_MainConfig__phr_Config__setup__prohibitPHR_Timer_sf1000; // sf1000 = 1000 subframes //lrf

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

Step SIX: Modify the central frequency

1. File Name: The eNB configuration File
   $OPENAIR_TARGETS/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tm1.100PRB.usrpb210.conf

- downlink_frequency = 2660000000L;
- uplink_frequency_offset = -120000000;
- downlink_frequency = 2625000000L;
- uplink_frequency_offset = -120000000;

Step SEVEN: Modify the cqi report schedule

1. File Name: eNB_scheduler_ulsch.c
   Function Name: schedule_ulsch_rnti
   Operation: Comment and Patch

>>
>

```
 if (status < RRC_CONNECTED)
       cqi_req = 0;
   //else if (UE_sched_ctrl->cqi_req_timer>30 )
```

- else if (UE_sched_ctrl->cqi_req_timer>30 && **sched_subframe !=8**) {

```
       cqi_req = 1;
       UE_sched_ctrl->cqi_req_timer=0;
   }
```

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<

STEP EIGHT: Add **Debug** Output

1. File Name: Ulsch_modualtion.c
   Operation: Add a new function

(Ignore for CTOS smartphone)

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>
//lrf: printout b_tilde
//-----------------------------------------------------------------------------
void b_tilde_print_hex_octets(  uint8_t* dataP, const signed long sizeP)
//-----------------------------------------------------------------------------
{
  unsigned long octet_index = 0;

  if (dataP == NULL) {
    return;
  }

  printf("+-----+-------------------------------------------------+\n");
  printf("|     |  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f |\n");
  printf("+-----+-------------------------------------------------+\n");

  for (octet_index = 0; octet_index < sizeP/8; octet_index++) {

    uint8_t temp = 0;
    int i = 0;
    for(i=0; i<8;i++){
        temp = (temp << 1) | dataP[octet_index*8+i];
    }
    if ((octet_index % 16) == 0) {
      if (octet_index != 0) {
        printf(" |\n");
      }

      printf(" %04lu |", octet_index);
    }
    /*
     * Print every single octet in hexadecimal form
     */
    printf(" %02x", temp);
    /*
     * Align newline and pipes according to the octets in groups of 2
     */
  }

  /*
   * Append enough spaces and put final pipe
   */
  unsigned char index;

  for (index = octet_index; index < 16; ++index) {
    printf("   ");
  }

   printf(" |\n");
}

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<
```

2. File Name: Ulsch_modualtion.c
Operation: patch

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>
ulsch_modulation()
  s = lte_gold_generic(&x1, &x2, 0);
}

//lrf output b_tilde
 • if(subframe == 8){
 • b_tilde_print_hex_octets(ulsch->b_tilde,G);
 • }
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

Section 2 Run the modified OAI with Smartphone

1. Compile and Run trace_rlc_udp.c in the machine where OAI base Station is running.
2. Install Android App