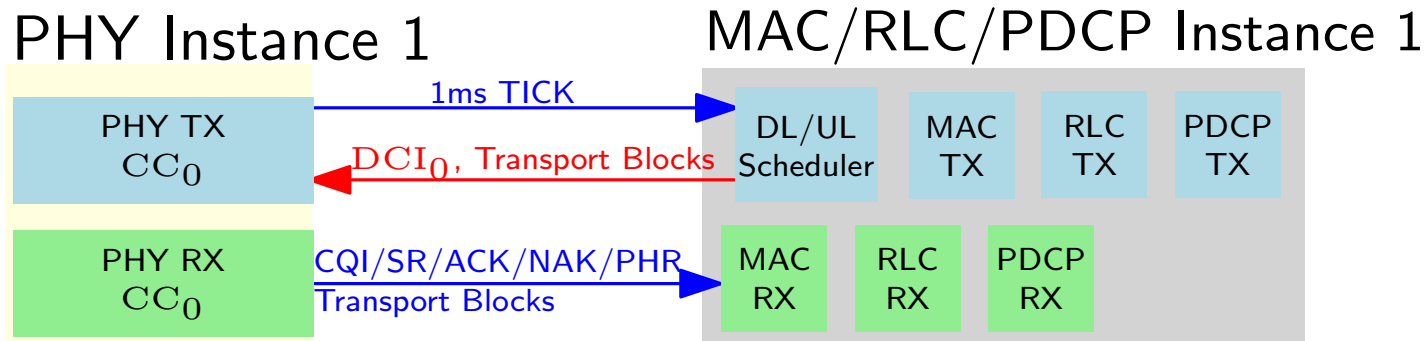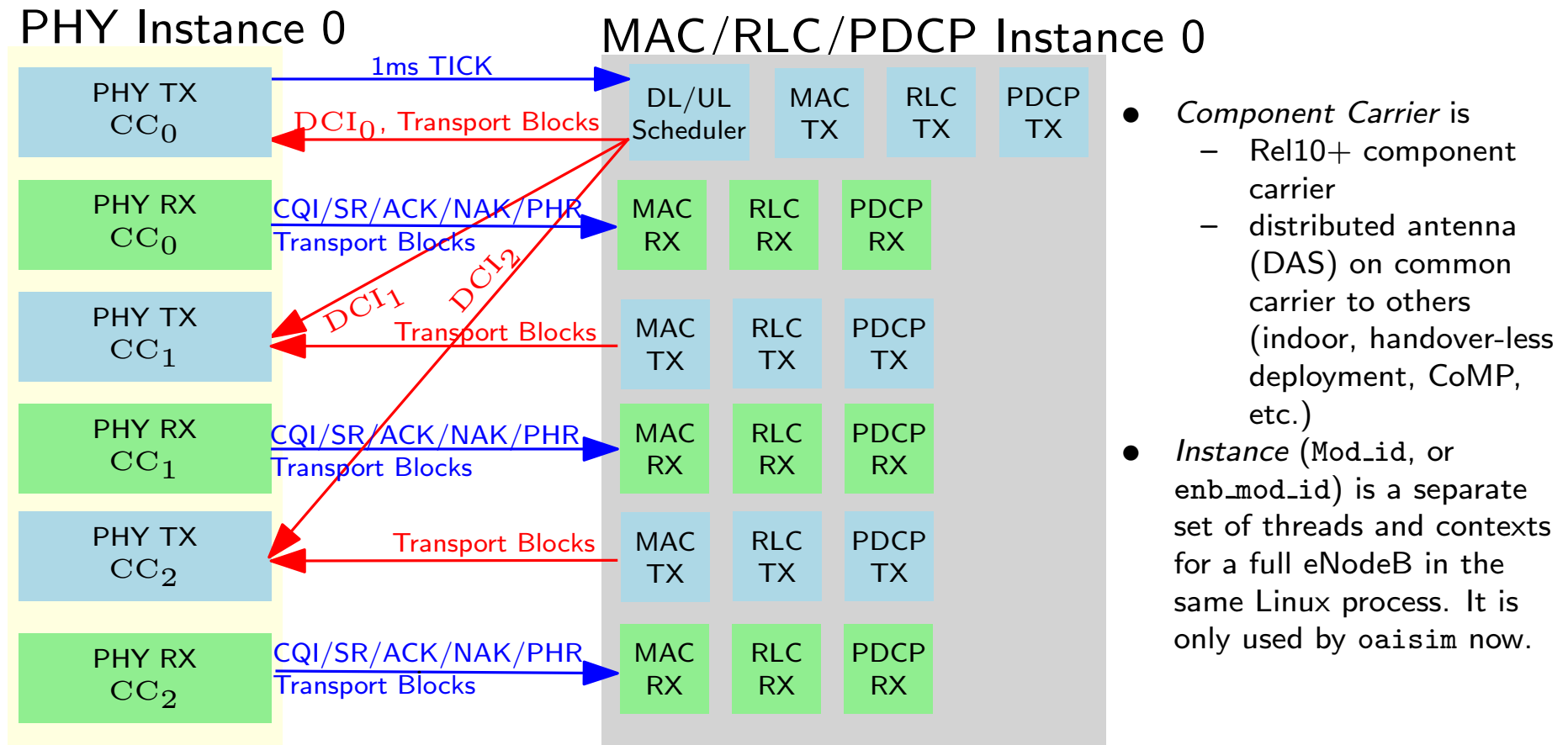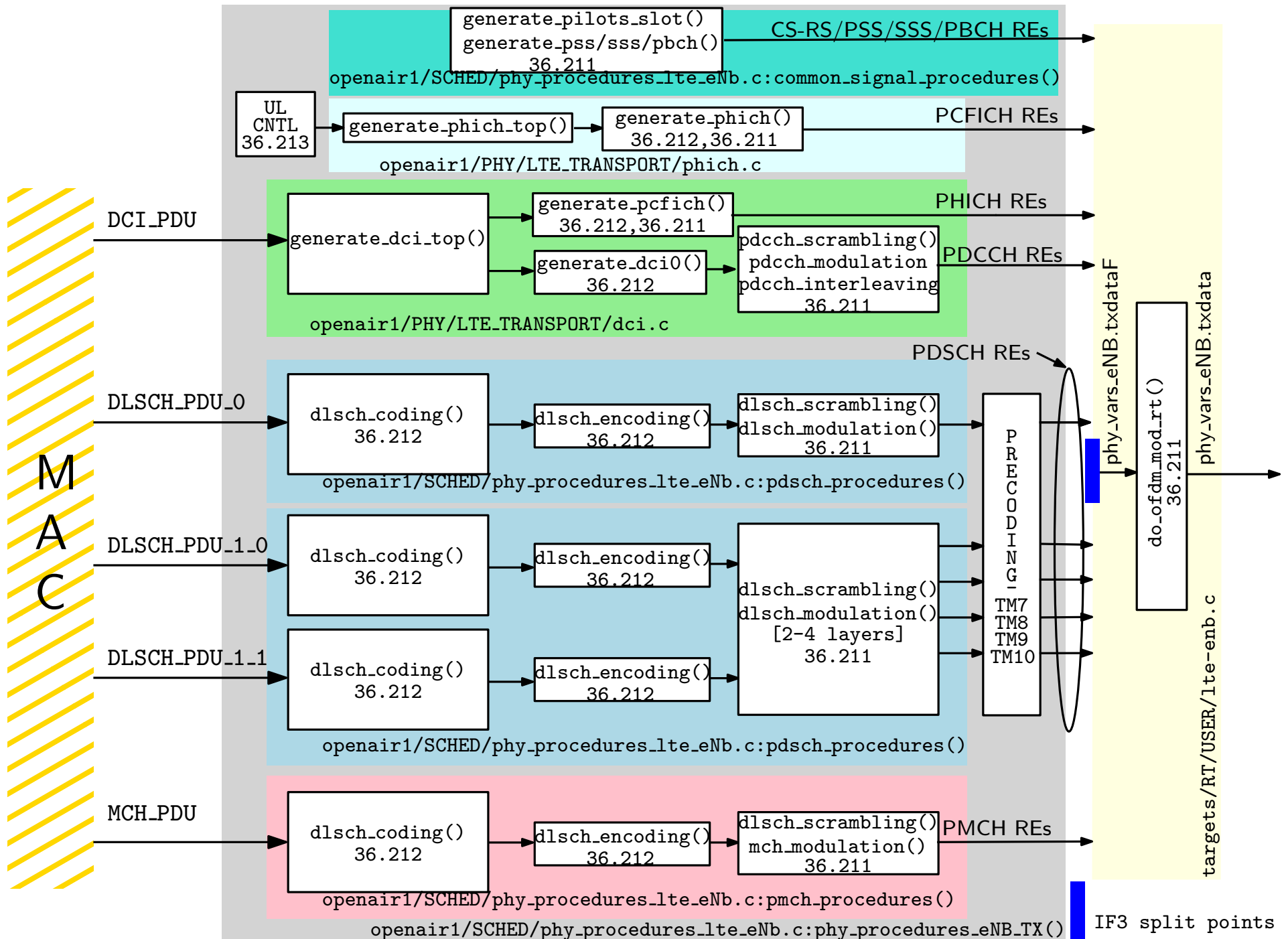# Some Preliminaries

- This description corresponds to the enhancement-10-harmony branch
- **purpose 1**: clean up interfaces within L1. L1/L2 are being done by others in the community (e.g. Bell Labs)
- **purpose 2**: harmonize HW plaforms (hence the name)
- **purpose 3**: make the interfaces flexible for functional splits between RU/SU and local radio gateways
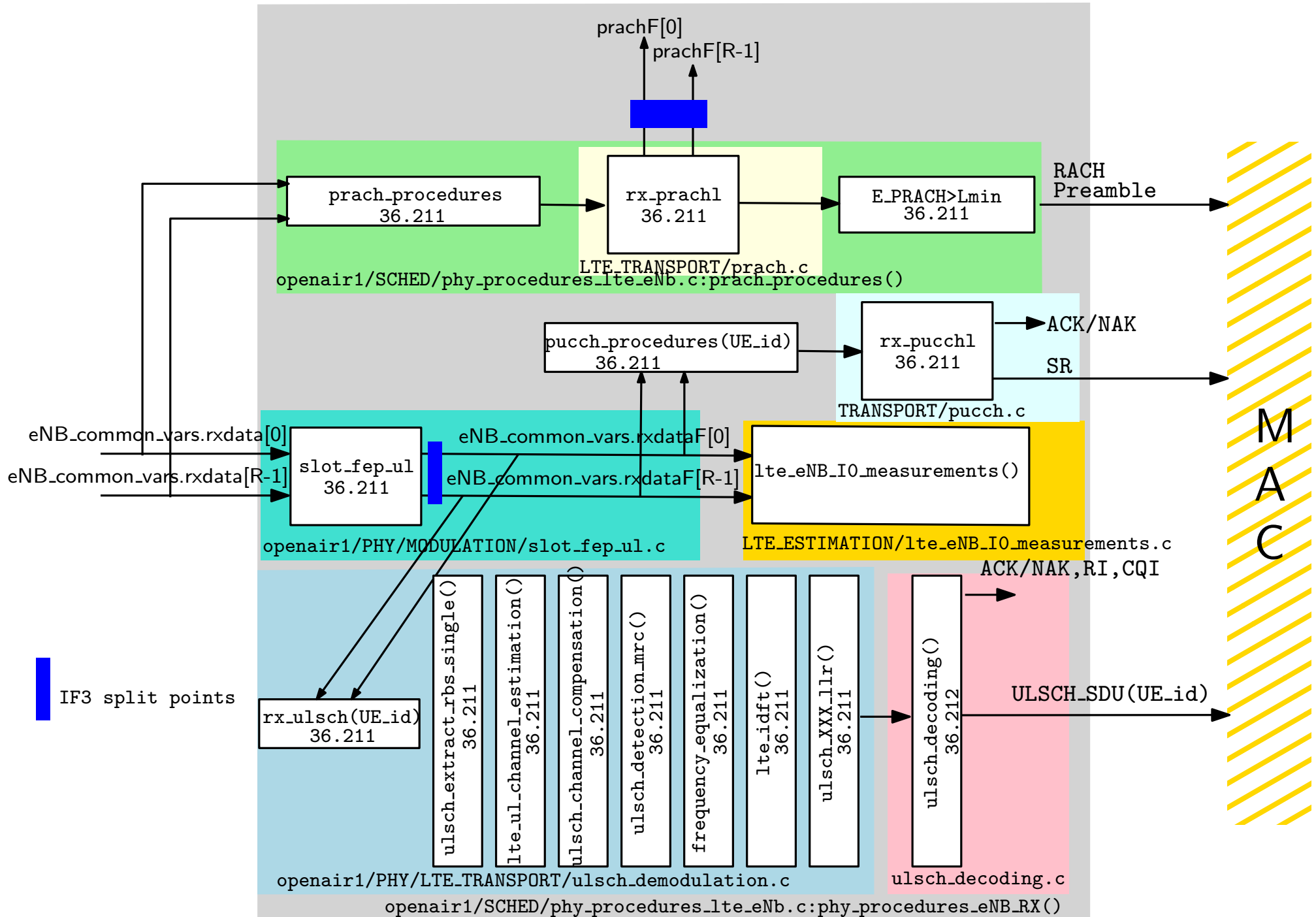- **purpose 4**: harmonize lte-softmodem (lte-ue/lte-enb) and oaisim
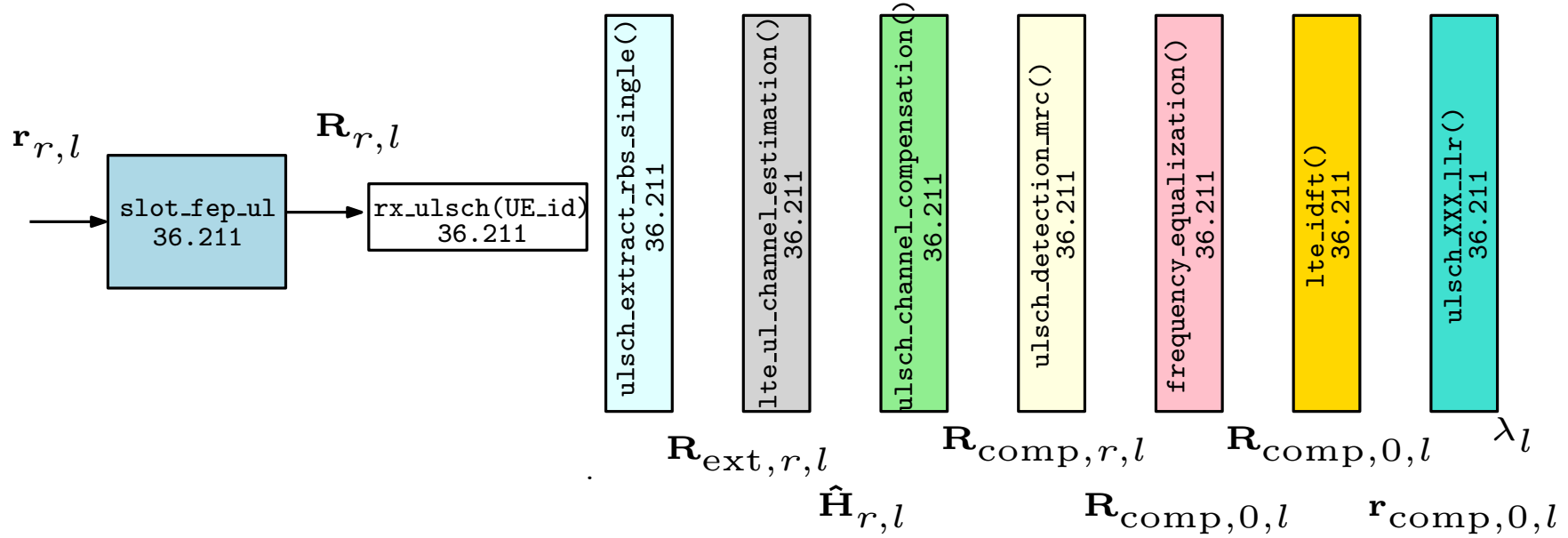
eNodeB Threads, Instances and Component Carriers

PHY Instance 0

| PHY TX CC_0 |
| PHY RX CC_0 |
| PHY TX CC_1 |
| PHY RX CC_1 |
| PHY TX CC_2 |
| PHY RX CC_2 |

MAC/RLC/PDCP Instance 0

1ms TICK

DCI_0, Transport Blocks

CQI/SR/ACK/NAK/PHR
Transport Blocks

DCI_1

DCI_2

Transport Blocks

CQI/SR/ACK/NAK/PHR
Transport Blocks

Transport Blocks

CQI/SR/ACK/NAK/PHR
Transport Blocks

DL/UL Scheduler	MAC TX	RLC TX	PDCP TX
MAC RX	RLC RX	PDCP RX	
MAC TX	RLC TX	PDCP TX	
MAC RX	RLC RX	PDCP RX	
MAC TX	RLC TX	PDCP TX	
MAC RX	RLC RX	PDCP RX	

- *Component Carrier* is
  - Rel10+ component carrier
  - distributed antenna (DAS) on common carrier to others (indoor, handover-less deployment, CoMP, etc.)
- *Instance* (`Mod_id`, or `enb_mod_id`) is a separate set of threads and contexts for a full eNodeB in the same Linux process. It is only used by `oaisim` now.

PHY Instance 1

| PHY TX CC_0 |
| PHY RX CC_0 |

MAC/RLC/PDCP Instance 1

1ms TICK

DCI_0, Transport Blocks

CQI/SR/ACK/NAK/PHR
Transport Blocks

| DL/UL Scheduler | MAC TX | RLC TX | PDCP TX |
| MAC RX | RLC RX | PDCP RX | |

eNodeB PHY TX Procedures

eNodeB PHY RX Procedures

prachF[0]

prachF[R-1]

prach_procedures
36.211

rx_prachl
36.211

E_PRACH>Lmin
36.211

RACH
Preamble

LTE_TRANSPORT/prach.c

openair1/SCHED/phy_procedures_lte_eNb.c:prach_procedures()

pucch_procedures(UE_id)
36.211

rx_pucchl
36.211

ACK/NAK

SR

TRANSPORT/pucch.c

eNB_common_vars.rxdata[0]

eNB_common_vars.rxdata[R-1]

slot_fep_ul
36.211

eNB_common_vars.rxdataF[0]

eNB_common_vars.rxdataF[R-1]

lte_eNB_I0_measurements()

openair1/PHY/MODULATION/slot_fep_ul.c

LTE_ESTIMATION/lte_eNB_I0_measurements.c

ACK/NAK,RI,CQI

M
A
C

IF3 split points

rx_ulsch(UE_id)
36.211

ulsch_extract_rbs_single()
36.211

lte_ul_channel_estimation()
36.211

ulsch_channel_compensation()
36.211

ulsch_detection_mrc()
36.211

frequency_equalization()
36.211

lte_idft()
36.211

ulsch_XXX_llr()
36.211

ulsch_decoding()
36.212

ULSCH_SDU(UE_id)

openair1/PHY/LTE_TRANSPORT/ulsch_demodulation.c

ulsch_decoding.c

openair1/SCHED/phy_procedures_lte_eNb.c:phy_procedures_eNB_RX()

eNodeB ULSCH Demodulation

$\mathbf{R}_{r,l} = \mathrm{DFT}_{N_{\mathrm{fft}}}(\mathbf{r}_{r,l} \odot \mathbf{F}_{7.5}), r = 0, 1, \cdots, R - 1, l = 0, 1, \cdots, N_{\mathrm{symb}} - 1$ **(eNB_common_vars→rxdataF[][])**

$R_{\mathrm{ext},r,l}(n) = R_{r,l}(12\mathrm{firstPRB} + n), n = 0, 1, \cdots, 12N_{\mathrm{PRB}} - 1$ **(eNB_pusch_vars→ulsch_rxdataF_ext[][])**

$\hat{\mathbf{H}}_{r,l} = \mathbf{R}_{\mathrm{ext},r,l} \odot \mathbf{DRS}_l^*(\mathrm{cyclicShift}, n_{\mathrm{DMRS}}(2), n_{\mathrm{PRS}})$, **(eNB_pusch_vars→drs_ch_estimates[])**

$\mathbf{R}_{\mathrm{comp},r,l} = \hat{\mathbf{H}}_r \odot \mathbf{R}_{\mathrm{ext},r,l} 2^{-\log_2|H_{\mathrm{max}}|}, \hat{\mathbf{H}}_r = \frac{1}{2}(\hat{\mathbf{H}}_{r,3} + \hat{\mathbf{H}}_{r,10})$
(eNB_pusch_vars→ulsch_rxdataF_comp)

$\mathbf{R}_{\mathrm{comp},0,l} = \frac{1}{R} \sum_{r=0}^{R-1} \mathbf{R}_{\mathrm{comp},r,l}$

$R_{\mathrm{comp},0,l}(n) = R_{\mathrm{comp},0,l}(n)\dot{Q}_8\left(\frac{1}{|\hat{H}(n)|^2 + I_0}\right), \hat{H}(n) = \sum_{r=0}^{R-1} \hat{H}_r(n)$
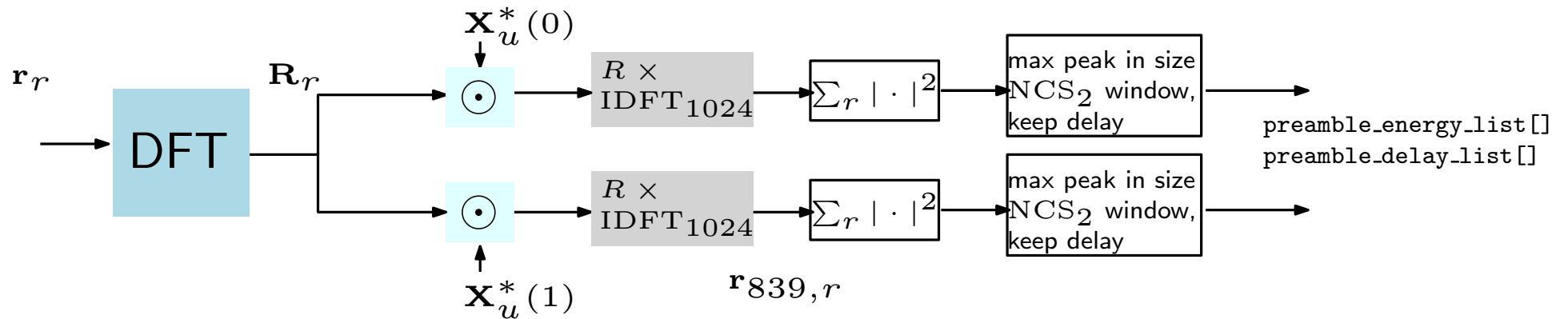
$\mathbf{r}_{\mathrm{comp},0,l} = \mathrm{DFT}_{12N_{\mathrm{PRB}}}(\mathbf{R}_{\mathrm{comp},0,l})$

$\mathrm{QPSK} : \lambda_l(2n) = \mathrm{Re}(r_{\mathrm{comp},0,l}(n)), \lambda_l(2n + 1) = \mathrm{Im}(r_{\mathrm{comp},0,l}(n))$ **(eNB_pusch_vars→ulsch_llr)**

$16\mathrm{QAM} : \lambda_l(4n) = \mathrm{Re}(r_{\mathrm{comp},0,l}(n)), \lambda_l(4n + 2) = \mathrm{Im}(r_{\mathrm{comp},0,l}(n))$

$\lambda_l(4n + 1) = |\mathrm{Re}(r_{\mathrm{comp},0,l}(n))| - 2\overline{|h(n)|}, \lambda_l(4n + 3) = |\mathrm{Im}(r_{\mathrm{comp},0,l}(n))| - 2\overline{|h(n)|}$
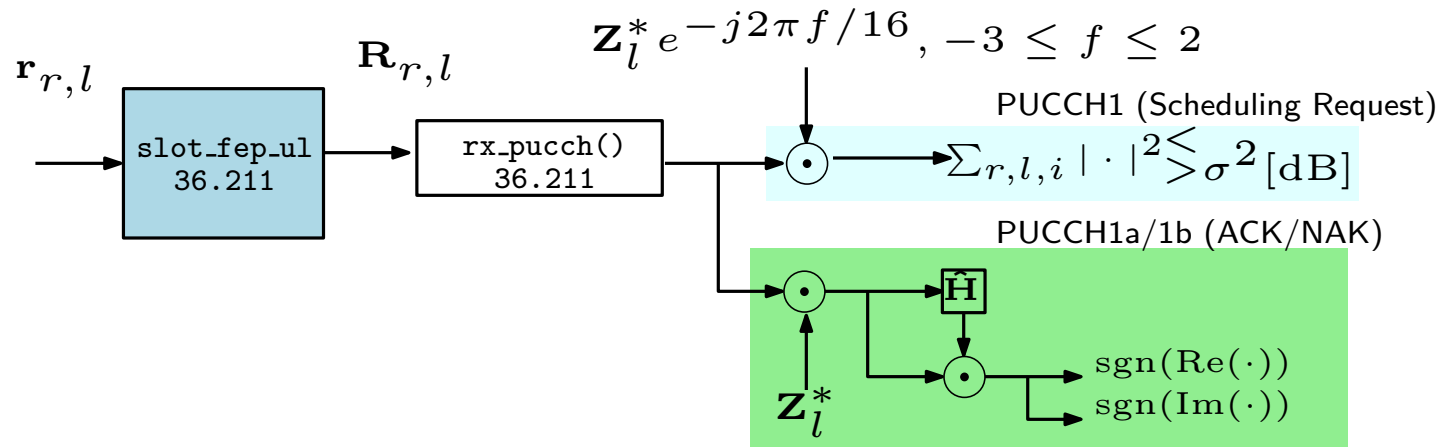
eNodeB PRACH Detection

$$\mathbf{R}_r = \mathrm{DFT}_{N_{\mathrm{PRACH}}}(\mathbf{r}_r),\ r = 0, 1, \cdots, R - 1\ \text{(lte\_eNB\_prach\_vars}\rightarrow\text{rxsigF[])}$$

$$\mathbf{R}_{\mathrm{comp},r} = \mathbf{R}_r \odot \mathbf{X}_u^*[i],\ r = 0, 1, \cdots, R - 1\ \text{(lte\_eNB\_prach\_vars}\rightarrow\text{prachF[])}$$

$$\mathbf{r}_{839,r} = \mathrm{IDFT}_{1024}(\mathbf{R}_{\mathrm{comp},r}),\ r = 0, 1, \cdots, R - 1\ \text{(lte\_eNB\_prach\_vars}\rightarrow\text{prach\_ifft[])}$$
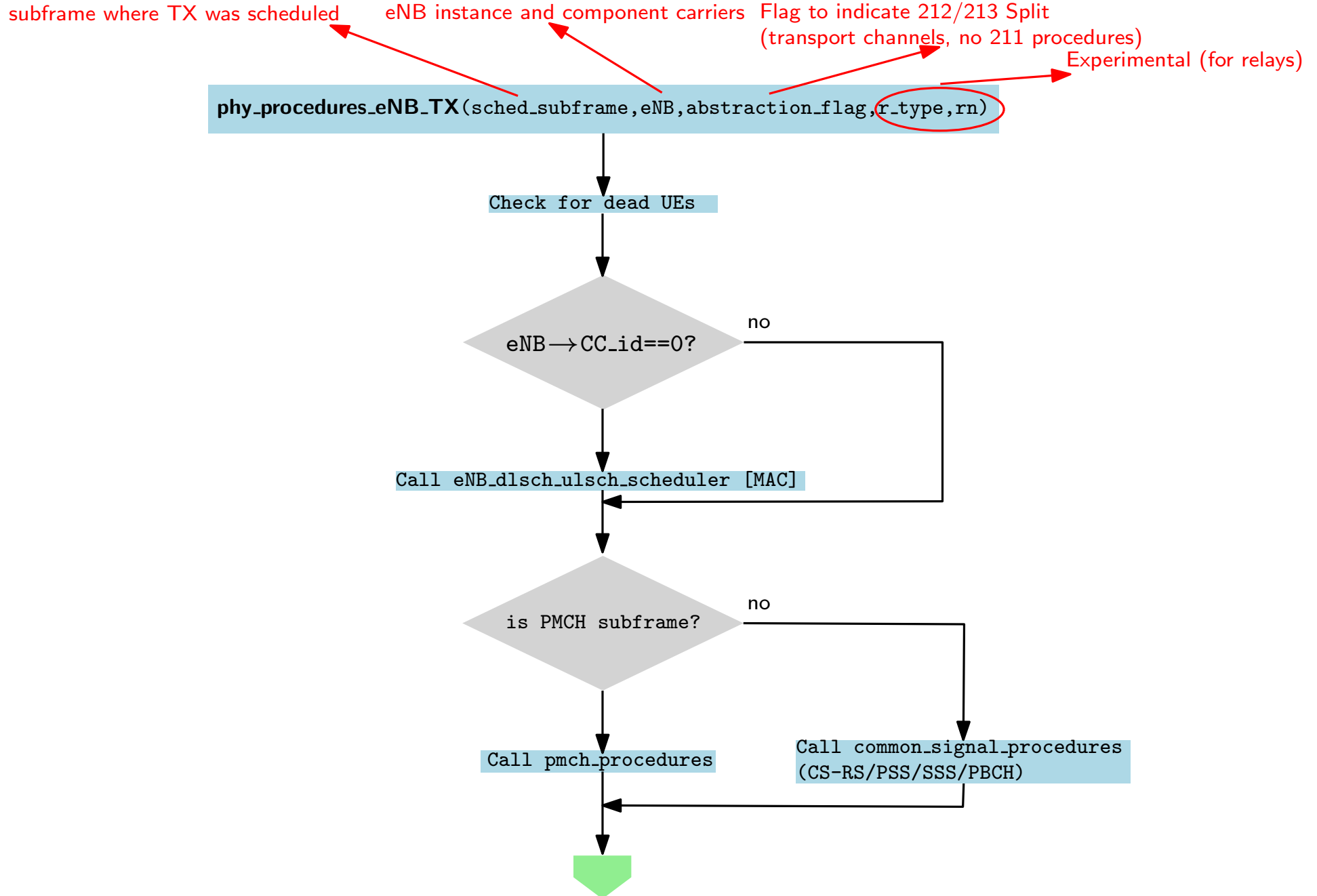
- PRACH detection is a quasi-optimal non-coherent receiver for vector observations (multiple antennas)

- correlation is done in the frequency-domain, number of correlations (in the example above 2) depends on *zeroCorrelationConfig* configuration parameter

- peak-detection (for delay estimation) is performed in each NCS time-window

eNodeB PUCCH Detection
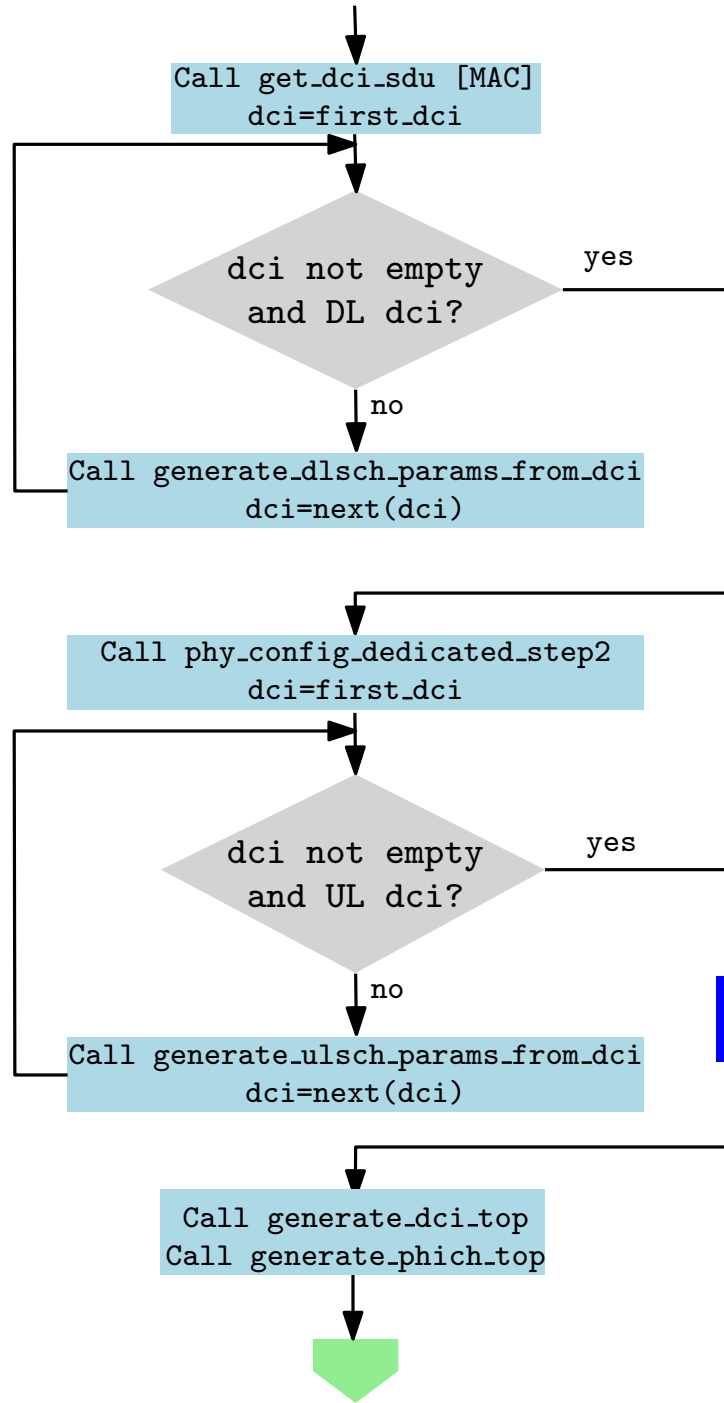

- PUCCH1 detection is a quasi-optimal non-coherent receiver (energy detector) for vector observations (multiple antennas) for scheduling request. Care is taken to handle residual frequency-offset.

- PUCCH1A/1B detection is quasi-coherent based on a rough channel estimate obtained on the 3 symbols without data modulation.

- In both cases, correlation is done in the frequency-domain

eNodeB TX Flowchart

subframe where TX was scheduled

eNB instance and component carriers

Flag to indicate 212/213 Split
(transport channels, no 211 procedures)

Experimental (for relays)

phy_procedures_eNB_TX(sched_subframe,eNB,abstraction_flag,r_type,rn)

Check for dead UEs

eNB→CC_id==0?   no

Call eNB_dlsch_ulsch_scheduler [MAC]

is PMCH subframe?   no

Call pmch_procedures

Call common_signal_procedures
(CS-RS/PSS/SSS/PBCH)

eNodeB TX Flowchart

Call get_dci_sdu [MAC]
dci=first_dci

dci not empty
and DL dci?

yes

no

Call generate_dlsch_params_from_dci
dci=next(dci)

Call phy_config_dedicated_step2
dci=first_dci

dci not empty
and UL dci?

yes

no

Call generate_ulsch_params_from_dci
dci=next(dci)

Potential Fronthaul
Split
(a bit like NodeB-RNC
interface)

Call generate_dci_top
Call generate_phich_top

eNodeB TX Flowchart

is dlsch_SI active?

yes

Call pdsch_procedures(dlsch_SI)

no

is dlsch_P active?

yes

Call pdsch_procedures(dlsch_P)

no

is dlsch_RA active?

yes

Call pdsch_procedures(dlsch_RA)

no

UE=first_UE from UE_list
[list of UE with t-CRNTI or CRNTI]

is dlsch_UE active?

yes

Call pdsch_procedures(dlsch_UE)

no

dlsch_eNB[UE].subframe_active=false
[no PUCCH ACK to be checked in TTI n+k]
UE=next(UE)

Potential Fronthaul Split (IF3)
[All Frequency-domain Signals are created]

no

UE=NULL?

yes

eNodeB Timing

LTE HARQ periodicity (FDD, TDD can be longer)

| SF $n-1$ | SF n | SF $n+1$ | SF $n+2$ | SF $n+3$ |

| RX SF $n-1$ | RX SF n | RX SF $n+1$ | RX SF $n+2$ |

| TX SF $n+1$ | TX SF $n+2$ | TX SF $n+3$ | TX SF $n+4$ | TX SF $n+5$ |

PRACH RX
SF n

- The current scheduling / timing requires approximately 1ms processing limit (basically 1 core RX, 1core TX)
- TX is delay by one slot to allow for some jitter
- a decision for UL reception in SF $n-1$ is needed in SF $n+3$ (for format0 DCI or PHICH)
- TX for $n+3$ will exit the PC somehwere in second-half of SF n+2
- a separate thread for PRACH RX is spawned in parallel to RX thread

eNodeB Timing

Even-slot processing subframe n
(`phy_procedures_eNB_common`)
- `trx_read(500`μs`)`
- `slot_fep` $2n$
- `launch TX thread subframe`
  $n + 3$
  - `phy_procedures_eNB_tx`
  - `ofdm_mod`
  - `trx_write(1 ms)`

Odd-slot processing subframe n
(`phy_procedures_eNB_common`)
- `trx_read (500` μs`)`
- `slot_fep` $2n + 1$
- `launch PRACH RX thread`
  (`subframe` n)
  - `prach_procedures`
- `phy_procedures_eNB_rx (subframe`
  n)