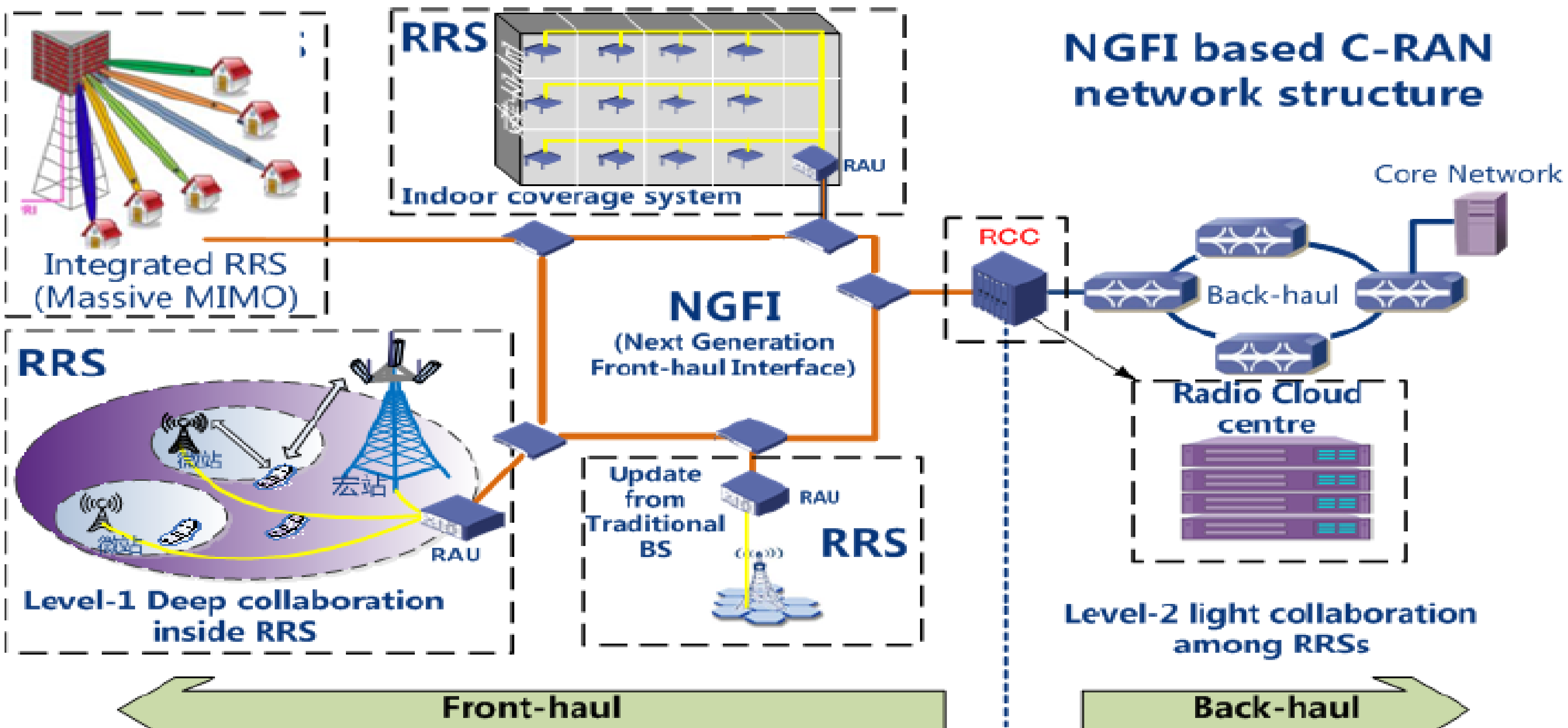
Some Preliminaries

- This description corresponds to the `develop` branch for RAN infrastructure components
- we describe
  1. node functions
  2. current functional splits and packet formats
  3. RAN procedures
  4. process scheduling

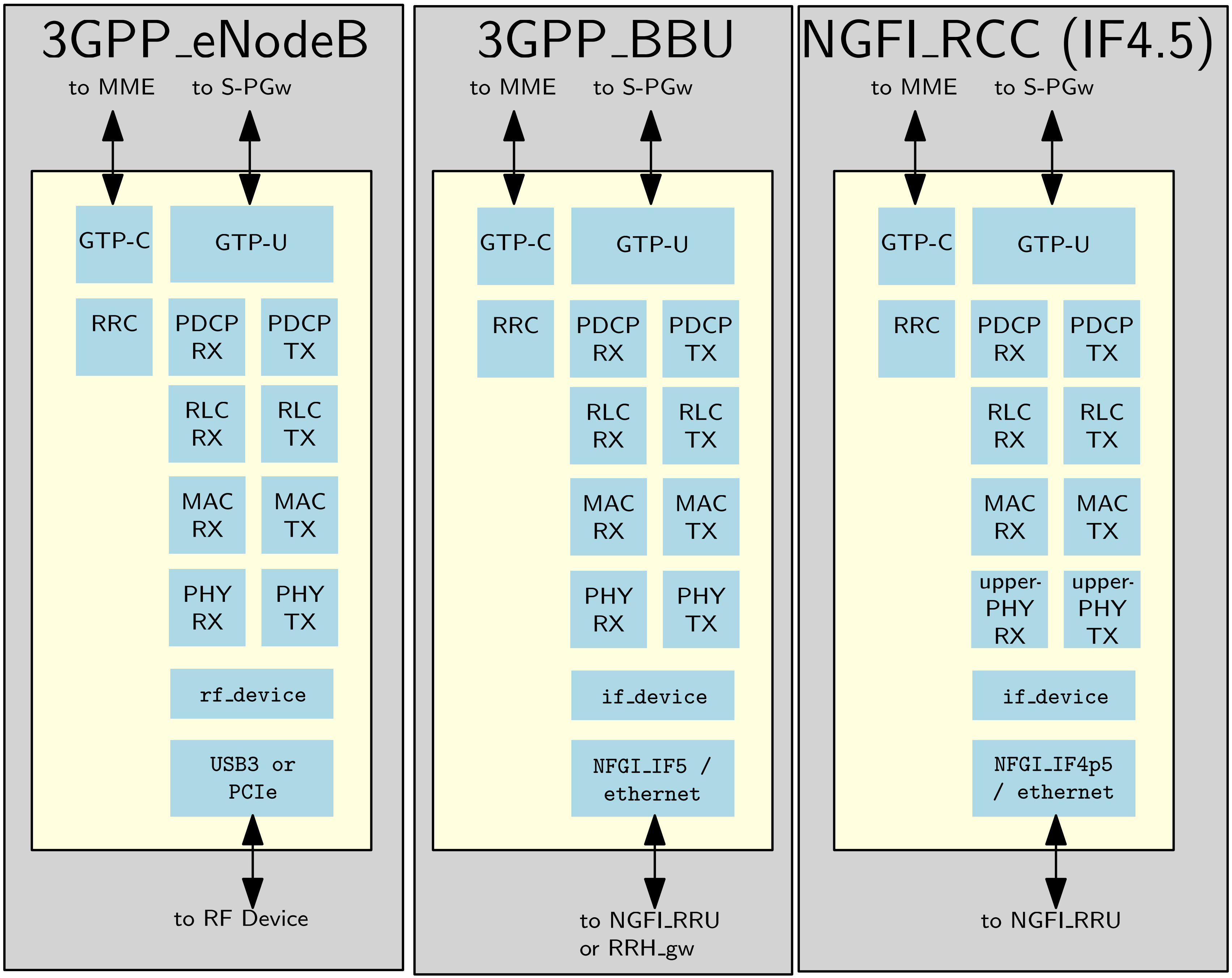
NGFI Harmonization

- New descriptions for OAI RAN infrastructure Node Functions
  - NGFI_RCC : Radio Cloud Center
  - NGFI_RAU : Radio Aggregation Unit
  - NGFI_RRU : Remote Radio Unit
  - 3GPP_BBU : Baseband Unit
  - 3GPP_eNodeB : Complete eNodeB

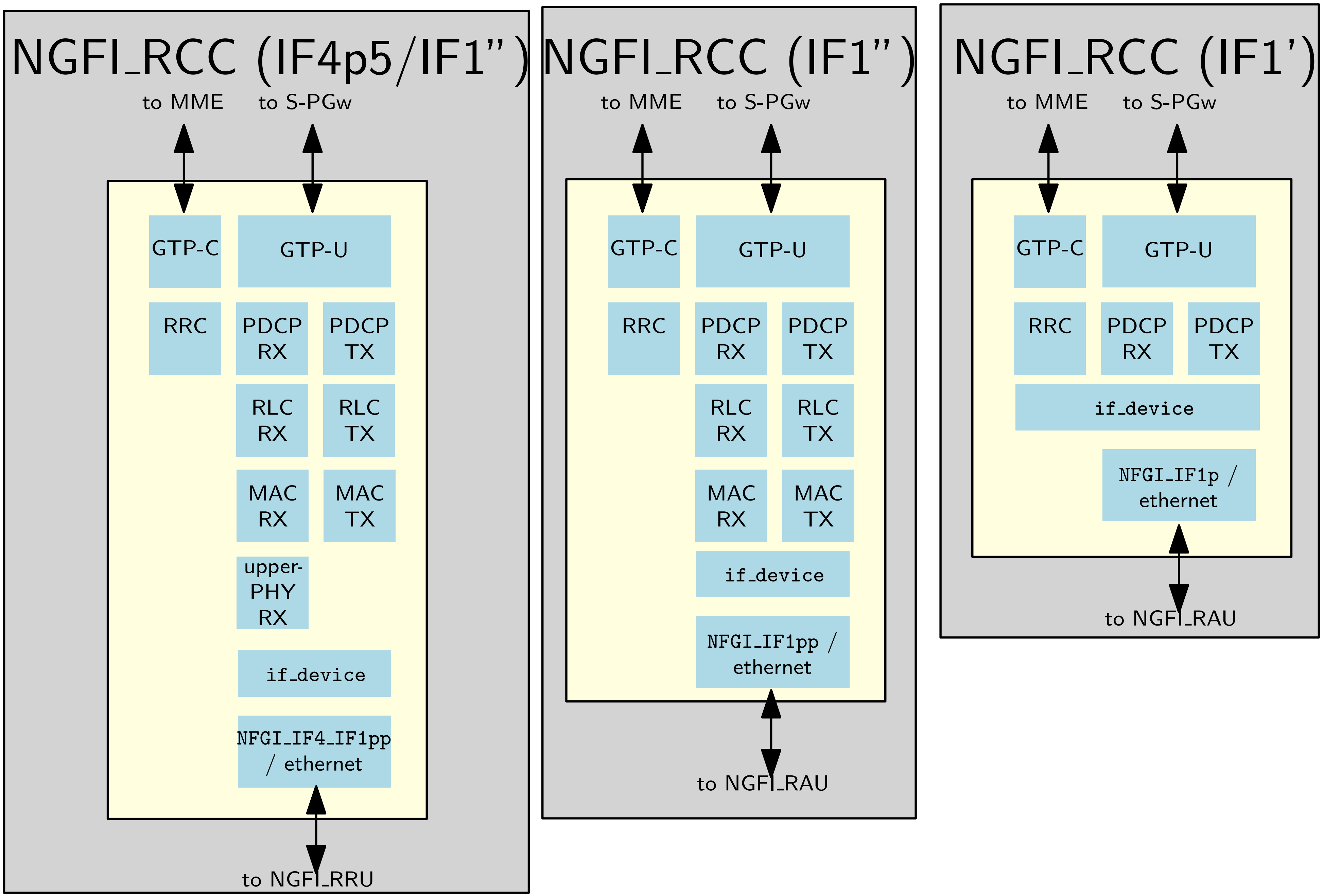


Source: China Mobile - http://www.windriver.com.cn/windforum/download/wf2015_networking03.pdf

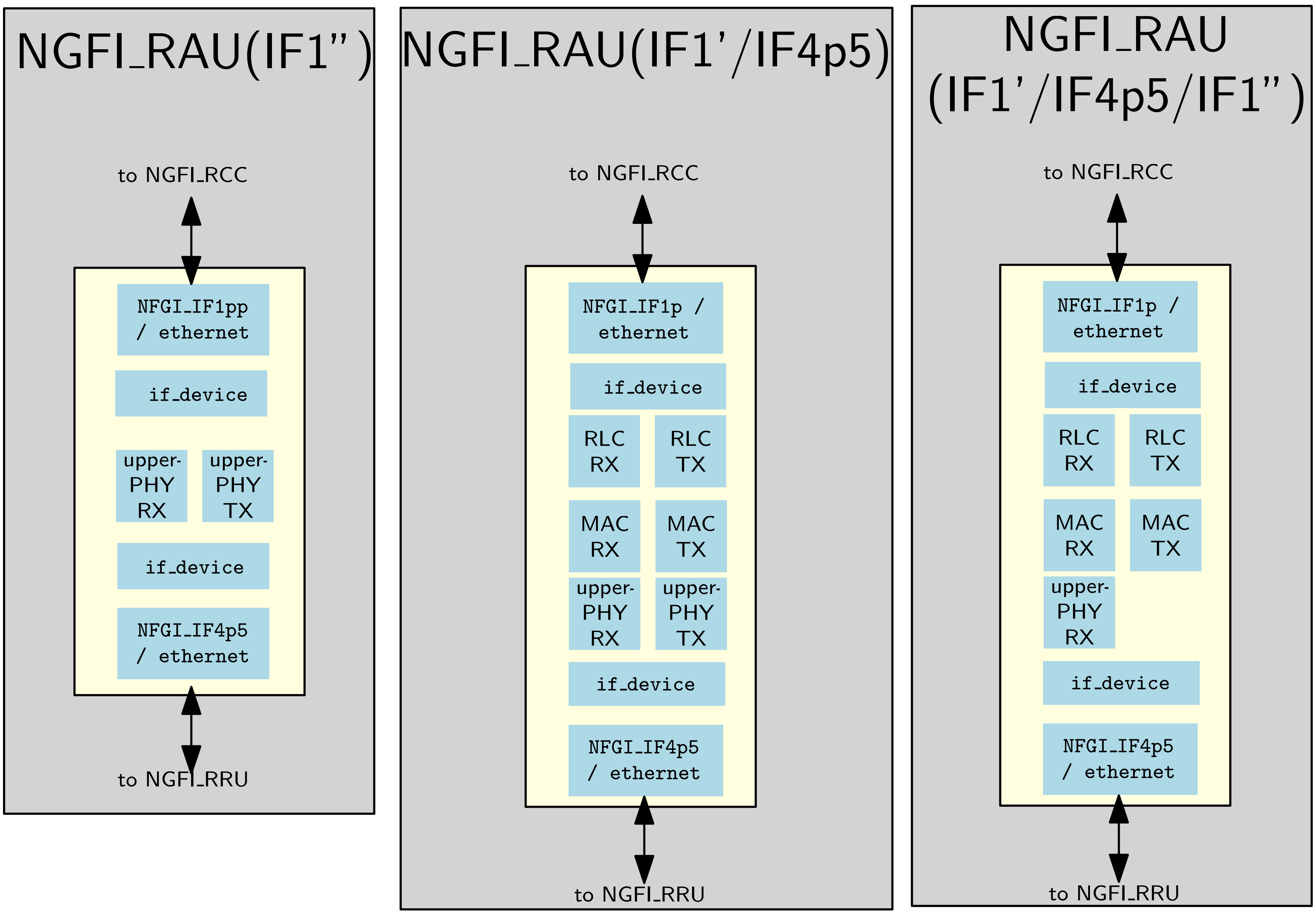
Functional Splits



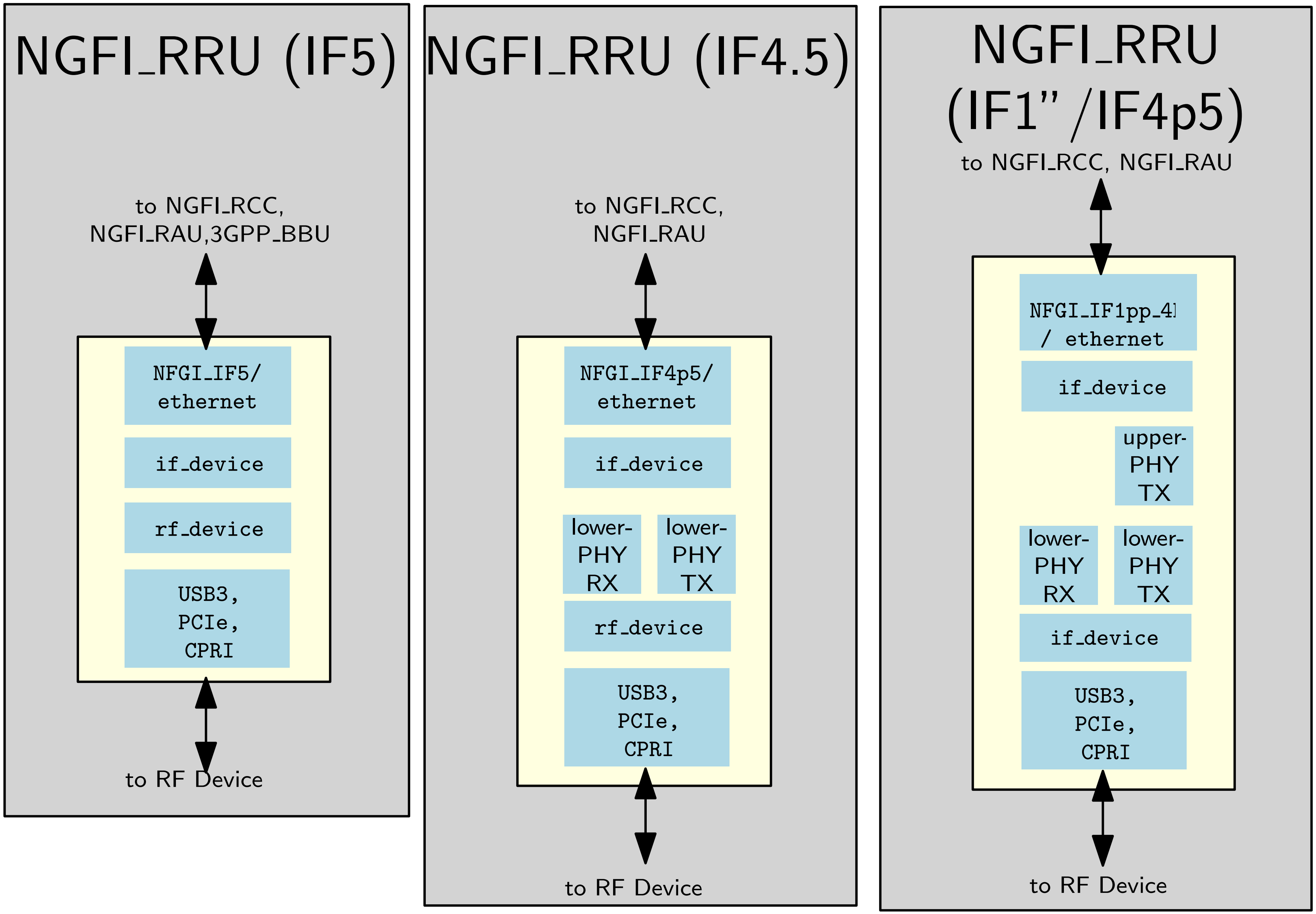
Functional Splits



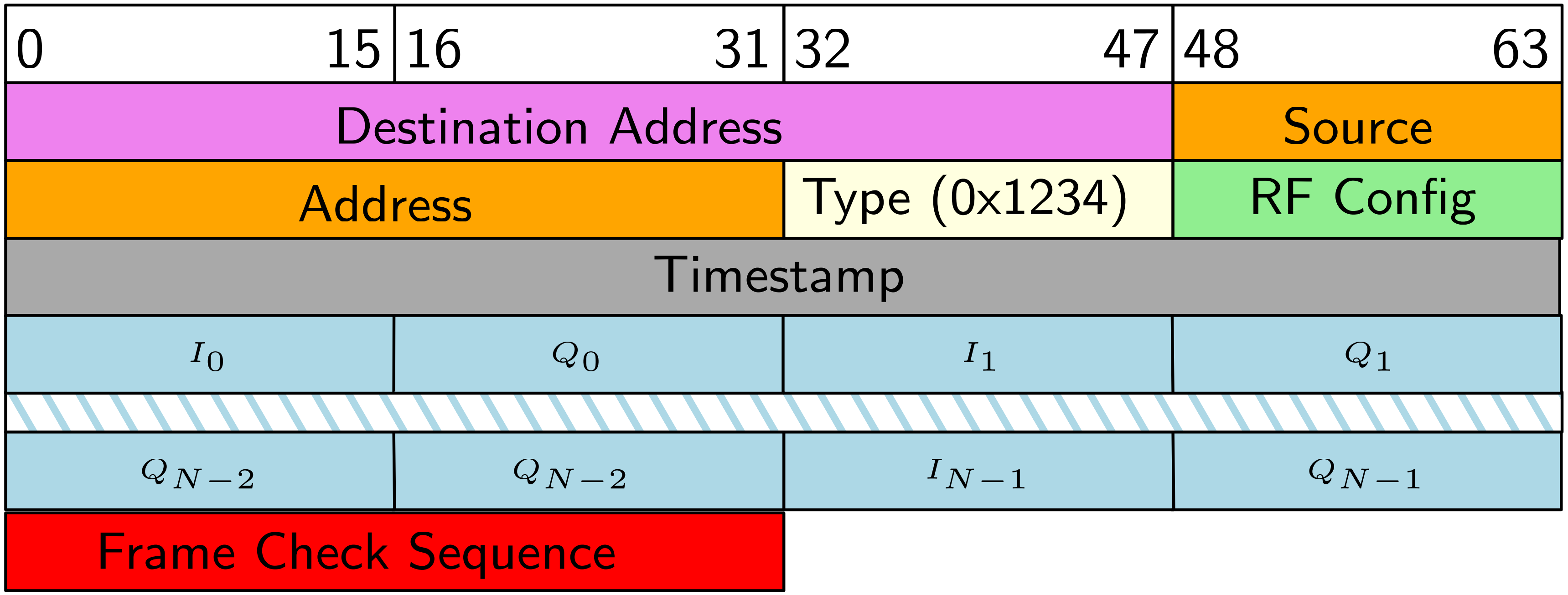
Functional Splits



Functional Splits

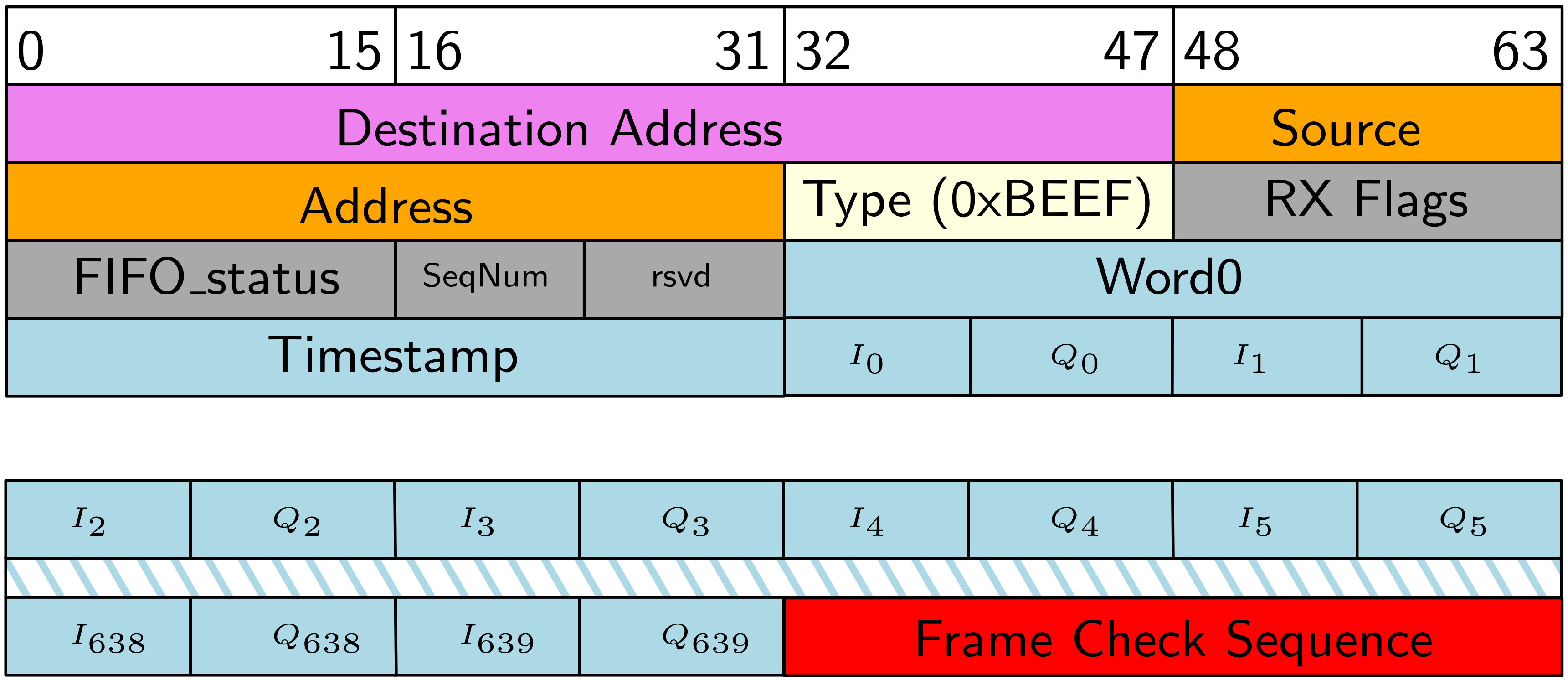


IF5 Packet Format (16-bit)



- **Type:** 2 byte (16 bit) field that specifies the RoE protocol
- **RX Config:** 16-bit. Currently just antenna index (0-7). Can later be used for gain/timing adjustments.
- **Timstamp:** Timestamp in samples of the first sample of the received packet.
- **data block :** Uncompressed IQ samples, 16-bit resolution for each real and imaginary component. N complex samples per packet. N can be configured at initialization.

IF5 Packet Format (8-bit)



- **Type:** 2 byte (16 bit) field that specifies the RoE protocol
- **RX Flags:** overrun indicator. should be '0'.
- **FIFO status:** 2 bytes. should be '0'.
- **SeqNum:** 1 byte. Sequence number of the ethernet packet.
- **rsvd:** 1 byte. shoult be '0'.
- **Word0:** 4 byte (32-bit). should be '0'.
- **Timstamp:** Timestamp in samples of the first sample of the received packet.
- **data block :** Uncompressed IQ samples, 8-bit resolution for each real and imaginary component. 640 complex samples per packet.

IF4p5 Packet Formats (RAW)

IF4p5 PRACH Packet (RRU→ RAU,RCC)

0 15	16 31	32 47	48 63
Destination Address			Source
Address		Type (0x080A)	Subtype (0x0021)
Reserved		LTE PRACH Configuration	
PRACH data block (one antenna)			
Frame Check Sequence			

- **Type:** 2 byte (16 bit) field that specifies the RoE protocol
- **Subtype:** 2 byte (16 bit) field that specifies the packet subtype
- **Reserved:** 4 byte (32 bit) field reserved
- **LTE PRACH conf:** 4 byte (32-bit) field that details the configuration of the LTE PRACH packet

field (0 is LSB, 31 is MSB)	description
rsvd (0:2)	Reserved.
ant (3:5)	3-bit Antenna index of received PRACH sequence
RF Num (6:21)	16-bit field indicating the Radio Frame number of this received PRACH sequence
SF Num (22:25)	4-bit field indicating the sub-frame number in the radio frame for the received PRACH sequence Valid range of 0 to 9.
Exponent (26:31)	FFT exponent output (0 if unscaled)

- **PRACH data block:** Uncompressed IQ samples

RE 0 (Real)	RE 0 (Imag)	RE 1 (Real)	RE 1 (Imag)
RE 837 (Real)	RE 837 (Imag)	RE 838 (Real)	RE 838 (Imag)

IF4p5 Packets : ULRE (RRU$\rightarrow$ RAU,RCC)

0 — 15		16 — 31		32 — 47		48 — 63	
Destination Address						Source	
Address				Type (0x080A)		Subtype (0x0019)	
Reserved				Frame status			
Gain 0		Gain 1		Gain 2		Gain 3	
Gain 4		Gain 5		Gain 6		Gain 7	
RE 0 Ant 1 (Re)	RE 0 Ant 1 (Im)	RE 1 Ant 1 (Re)	RE 1 Ant 1 (Im)	RE 2 Ant 1 (Re)	RE 2 Ant 1 (Im)	RE 3 Ant 1 (Re)	RE 3 Ant 1 (Im)
RE $N-4$ Ant R (Re)	RE $N-4$ Ant R (Im)	RE $N-3$ Ant R (Re)	RE $N-3$ Ant R (Im)	RE $N-2$ Ant R (Re)	RE $N-2$ Ant R (Im)	RE $N-1$ Ant R (Re)	RE $N-1$ Ant R (Im)
Frame Check Sequence							

- **Type:** 2 byte (16 bit) field that specifies the RoE protocol
- **Subtype:** 2 byte (16 bit) field that specifies the packet subtype
- **Reserved:** 4 byte (32 bit) field reserved
- **Frame Status:** 4 byte (32 bit) field

field (0 is LSB, 31 is MSB)	description
ant (0:2)	The number of Antenna Carriers represented in the packet. Antenna numbers range from 0 to 7 with valid inputs being 0,1, 3 and 7 (1,2,4,8 antennas)
ant start (3:5)	starting antenna number
RF Num (6:21)	16-bit field indicating the Radio Frame number of this received PRACH sequence
SF Num (22:25)	4-bit field indicating the sub-frame number in the radio frame for the received PRACH sequence Valid range of 0 to 9.
Sym Num: (26:29) **rsvd:** (30:31)	Symbol number. Valid range of 0 to 13. reserved

- **ULRE data block:** compressed IQ samples (8-bit A-law). N is the number of resource elements $N_{\mathrm{RB}}^{\mathrm{UL}}$.

IF4p5 Packets : DLRE (RAU,RCC →RRU)

0 15		16 31		32 47		48 63	
Destination Address						Source	
Address				Type (0x080A)		Subtype (0x0020)	
Reserved				Frame status			
RE 0 Ant 1 (Re)	RE 0 Ant 1 (Im)	RE 1 Ant 1 (Re)	RE 1 Ant 1 (Im)	RE 2 Ant 1 (Re)	RE 2 Ant 1 (Im)	RE 3 Ant 1 (Re)	RE 3 Ant 1 (Im)
RE $N-4$ Ant R (Re)	RE $N-4$ Ant R (Im)	RE $N-3$ Ant R (Re)	RE $N-3$ Ant R (Im)	RE $N-2$ Ant R (Re)	RE $N-2$ Ant R (Im)	RE $N-1$ Ant R (Re)	RE $N-1$ Ant R (Im)
Frame Check Sequence							

- **Type:** 2 byte (16 bit) field that specifies the RoE protocol
- **Subtype:** 2 byte (16 bit) field that specifies the packet subtype
- **Reserved:** 4 byte (32 bit) field reserved
- **Frame Status:** 4 byte (32 bit) field

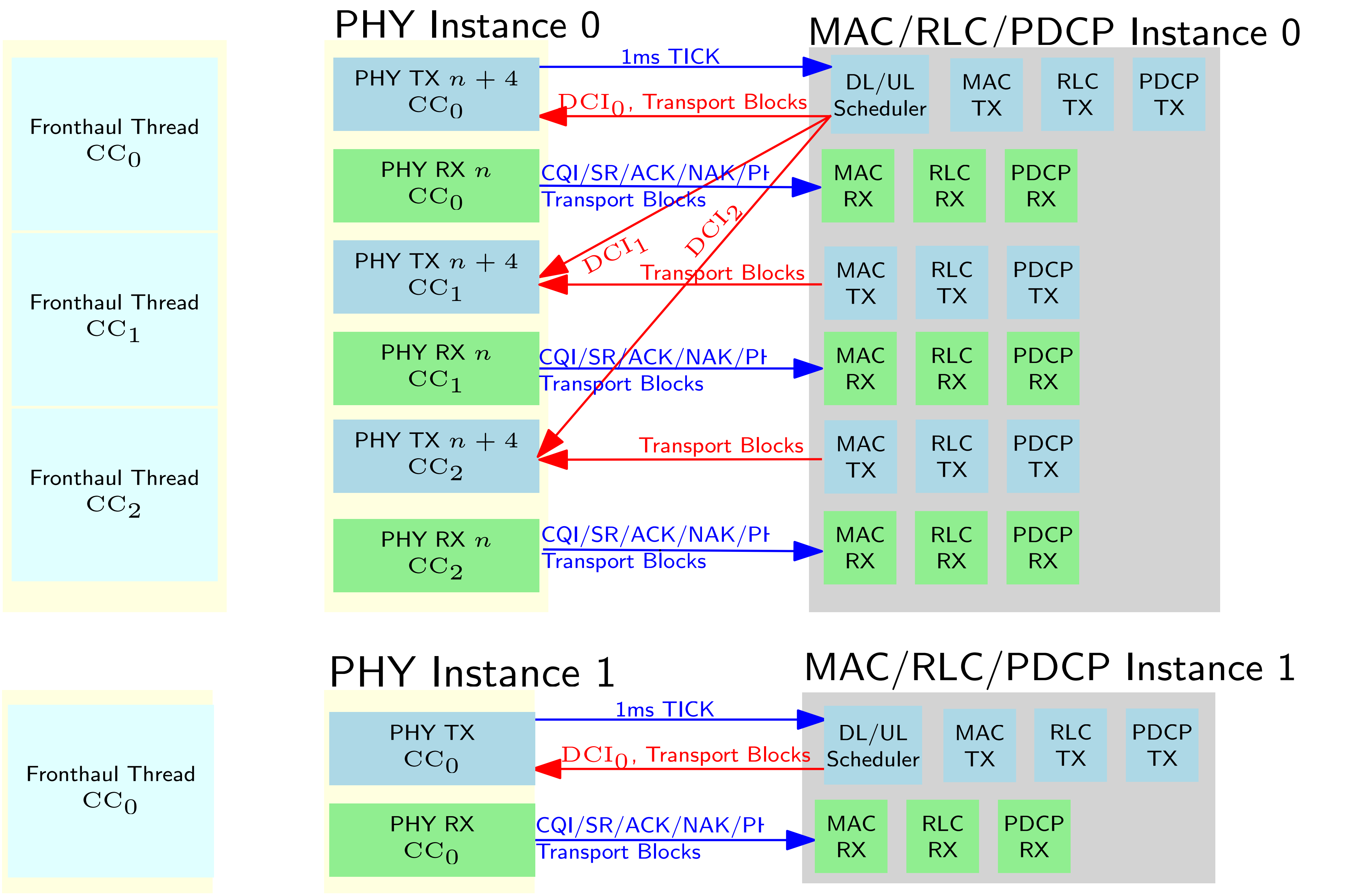
field (0 is LSB, 31 is MSB)	description
ant (0:2)	The number of Antenna Carriers represented in the packet. Antenna numbers range from 0 to 7 with valid inputs being 0,1, 3 and 7 (1,2,4,8 antennas)
ant start (3:5)	starting antenna number
RF Num (6:21)	16-bit field indicating the Radio Frame number of this received PRACH sequence
SF Num (22:25)	4-bit field indicating the sub-frame number in the radio frame for the received PRACH sequence Valid range of 0 to 9.
Sym Num: (26:29)	Symbol number. Valid range of 0 to 13.
rsvd: (30:31)	reserved

- DLRE data block : compressed IQ samples (8-bit A-law). N is the number of resource elements $N_{\mathrm{RB}}^{\mathrm{DL}}$.

Some Notes on usage of splits

- IF4p5 corresponds to the split-point at the input (TX) and output (RX) of the OFDM symbol generator (i.e. frequency-domain signals). According to NGFI, IF4 is "Resource mapping and IFFT" and "FFT and Resource demapping". We currently do not try to exploit multiplexing gains for unused spectral components. So, IF4p5 is simply compressed transmitted or received resource elements in the usable channel band.
- The simplest deployment for DAS (indoor) is one NGFI_RCC (IF4p5) and many NGFI_RRU (IF4p5). Spatio-temportal filtering (Precoding, later) is done in RCC and RRU perform IFFT/FFT and signal generation/acquisition. Fronthaul rates in this case are feasible with 1GbE copper links. This allows for PoE in addition to fronthaul data.
- More complex indoor, for instance with RCC in a common data center with outdoor RRS, could be
    1. RCC-RAU with IF1", RAU-RRU with IF4p5. Spatio-temporal filtering is done in frequency-domain in RAU along with full TX and RX processing (L1/L2) for the indoor RRS. Note that IF1' fronthaul on TX to RRU would be difficult because spatio-temporal filtering should be used. RRU does only IFFT/FFT and signal generation/acquisition
    2. RCC-RAU with IF1', RAU-RRU with IF4p5. Here RCC does L2, RAU does L1 and precoding for RRS.
- A massive-MIMO solution would consist either of
    1. an embedded RAU with processing (Spatio-temporal in frequency-domain, lower/upper PHY TX/RX) like the IF1' DAS solution above
    2. or more simply a high-speed fronthaul (IF4p5) with an RAU for multiple sites
    3. directly connected to RCC via high-speed IF4p5 (several virtual cells, precoder and IFFT/FFT in array).
- RCC solution with IF1" would cater to evolved-PDCP for heterogenity (4G,5G,WIFI,IoT)
- Currently supported node functionalities
    1. `3GPP_eNodeB`
    2. `3GPP_eNodeB_BBU` [NGFI_IF5]
    3. `NGFI_RCC` [NGFI_IF4p5]
    4. `NGFI_RRU` [NGFI_IF5]
    5. `NGFI_RRU` [NGFI_IF4p5]

FH/eNodeB Threads, Instances and Component Carriers

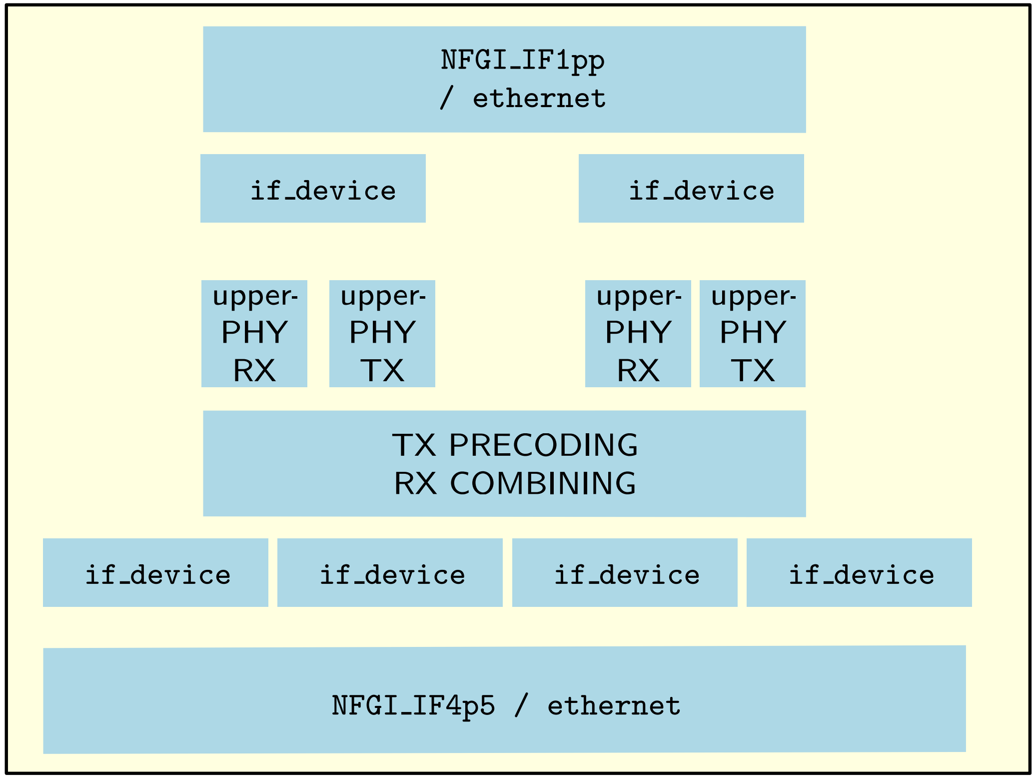


FH/RAU/RCC Threads, Instances and Component Carriers

- *Component Carrier* is
  - sectored antenna component
  - Rel10+ component carrier
  - virtual cell for DAS or Massive-MIMO array
- *Instance* (`Mod_id`, or `enb_mod_id`) is a separate set of threads and contexts for a full eNodeB, or at least the MAC/RLC entities, in the same Linux process. It is only used by `oaisim` now.
- Example configurations
  - *itsolated eNB:* one instance and one or several component carriers (multiple-frequencies or antenna sectors). Here there is a common layer-2 instnance driving multiple layer1 procedures
  - *indoor DAS system (full centralization in RCC or RAU)* Single layer 2 instance diving multiple component carriers (virtual cells). Here RAU only implements precoding (mapping from logical vCells to antenna ports)
  - *indoor DAS system (RCC split with L1/L2 RAU) Multiple layer 2 instances each driving one or more component carriers* here the RAU implements multiple L1/L2 instances and precoding function
  - *massive-MIMO array* same as 2nd indorr DAS system (i.e. integrated L1/L2 RAU with array)

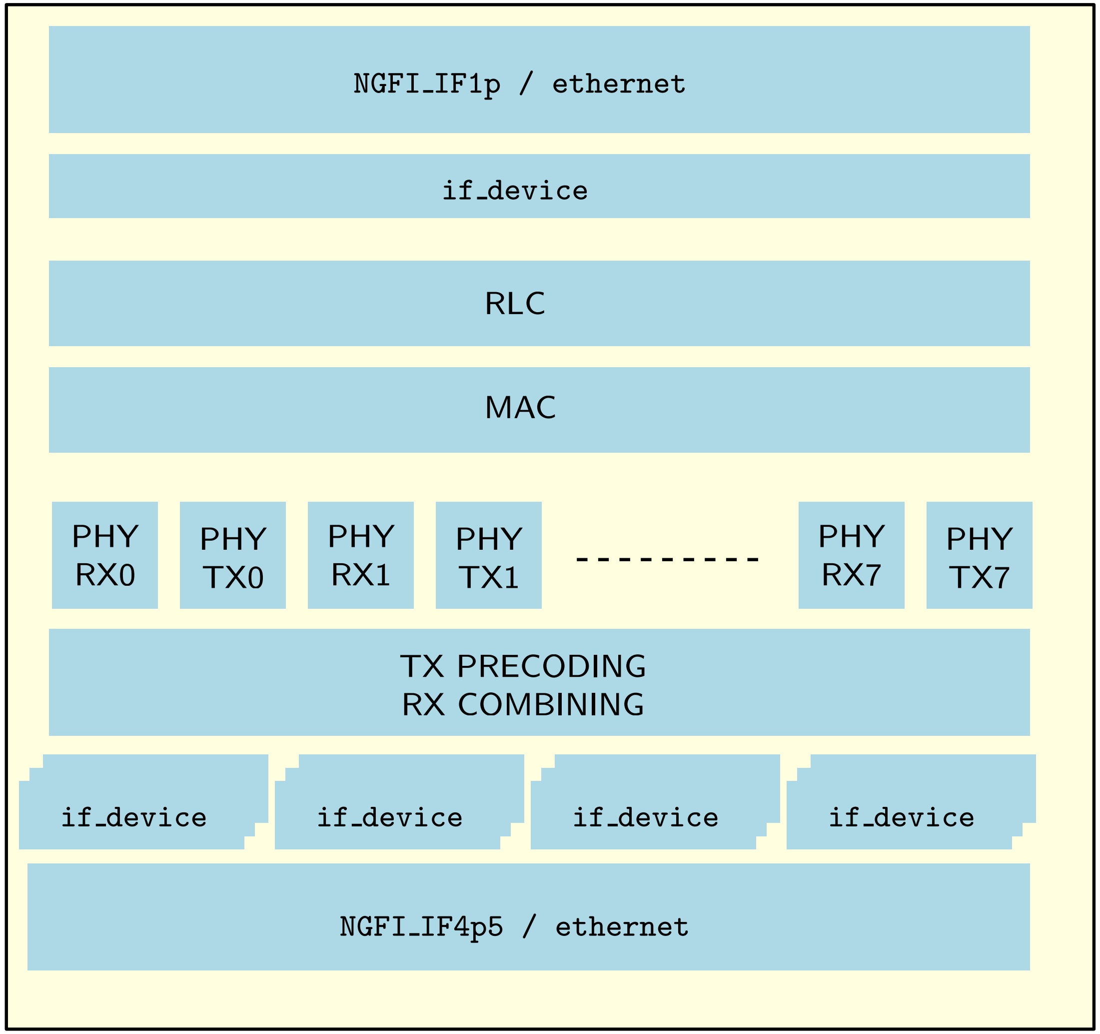
FH/RAU/RCC Threads, Instances and Component Carriers

- Example: RAU with `NGFI_IF1pp` fronthaul (MAC/PHY split), 2 vCell logical interfaces (2 L1/L2 instances, or 1 L2 instance and 2 CCs), 4 RRUs with `NGFI_IF4p5`



FH/RAU/RCC Threads, Instances and Component Carriers

- Example: massive-MIMO RAU with `NGFI_IF1p` fronthaul (PDCP split), 8 L1 component carriers,1 L2 instances, many local RRUs with `NGFI_IF4p5`



FH/RAU/RCC Threads, Instances and Component Carriers

- Threads (all in `targets/RT/USER/lte-enb.c`)
  - multi RX/TX thread mode (optional)
    - `eNB_thread_FH`: Thread per CC/Instance which manages fronthaul interface
    - `eNB_thread_rxtx`: 2 threads per CC/Instance which do both RX procedures for subframe n and TX procedures for subframe $n+4$. One operates on even subframes, one on odd. This allows 1ms subframe processing to use multiple-cores.
  - single RX/TX thread mode (default)
    - `eNB_thread_single`: Thread per CC/Instance which sequentially performs fronthaul read, RX/TX processing for subframe n and $n+4$ followed by fronthaul write
  - `eNB_prach`: Thread per CC_id/Instance for PRACH processing

- Synchronization on fronthaul interface
  - *synch_to_ext_device* : synchronizes to incoming samples from RF or Fronthaul interface using blocking read
  - *synch_to_other* : synchronizes via POSIX mechanism to other source (other CC, timer) which maintains real-time.

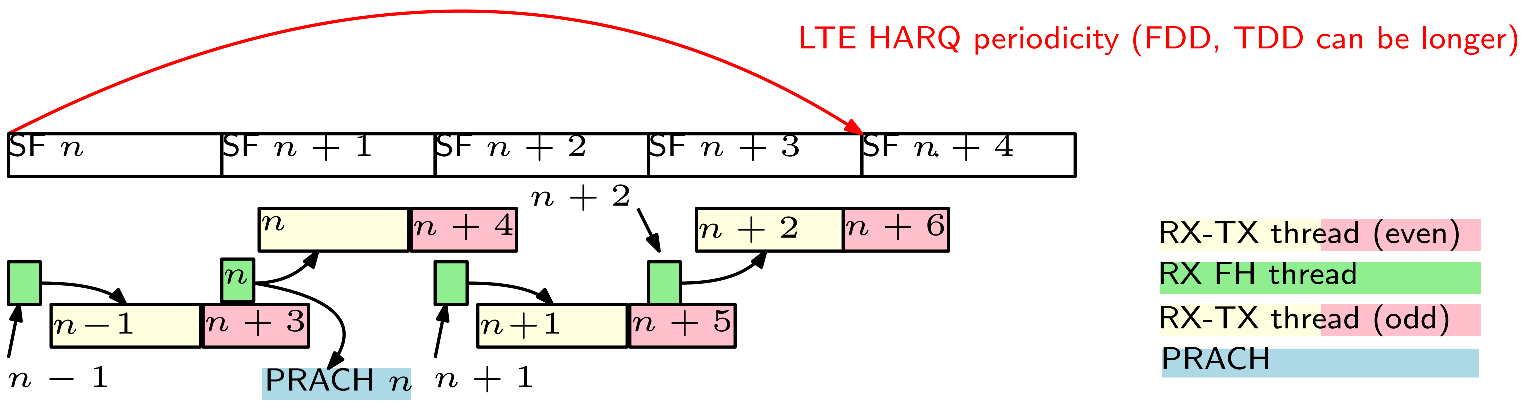
FH/RAU/RCC Threads, Instances and Component Carriers

- Threads (all in `targets/RT/USER/lte-enb.c`)
  - multi RX/TX thread mode (optional)
    - `eNB_thread_FH`: Thread per CC/Instance which manages fronthaul interface
    - `eNB_thread_rxtx`: 2 threads per CC/Instance which do both RX procedures for subframe n and TX procedures for subframe $n+4$. One operates on even subframes, one on odd. This allows 1ms subframe processing to use multiple-cores.
  - single RX/TX thread mode (default)
    - `eNB_thread_single`: Thread per CC/Instance which sequentially performs fronthaul read, RX/TX processing for subframe n and $n+4$ followed by fronthaul write
  - `eNB_prach`: Thread per CC_id/Instance for PRACH processing
  - `fh_**_asynch_**`: Threads for FH reception which do not block the main thread (`eNB_thread_FH` or `eNB_thread_single`). Signal is synchronized via timestamp in the received packets.

FH/RAU/RCC Threads, Instances and Component Carriers

- Synchronization on fronthaul interface
  - *synch_to_ext_device* : synchronizes to incoming samples from RF or Fronthaul interface using blocking read
  - *synch_to_other* : synchronizes via POSIX mechanism to other source (other CC, timer) which maintains real-time.

RAN Timing (multi-thread mode)



- The current processing requires approximately 1ms peak in each direction (basically 1 core RX, 1core TX). The current architecture will work on a single core if the sum of RX and TX procedures is limited to 1ms. It can fit on 2 cores if the sum of RX,TX and PRACH is less than 2ms.
- four threads, RX FH, RX-TX even, RX-TX odd and PRACH. RX FH blocks on reception of an entire subframe of signal. It then wakes up one of the 2 RX-TX threads and the PRACH thread if necessary. The RX-TX thread first performs RX common and ue-specific processing for subframe n and then TX common and ue-specific processing for subframe $n+4$. This insures the data dependency between TX $n+4$ and RX n is respected. The duration of this thread should be less than 2ms which can compensate some jitter on the RX processing.
- An alternative with only one RX-TX thread is also being investigated. Here some of the lower-level routines (FFT, turbo-encode, turbo-decode) will be run on multiple-cores. In this case, the entire RX-TX processing should fit within 1ms.

RAN Timing

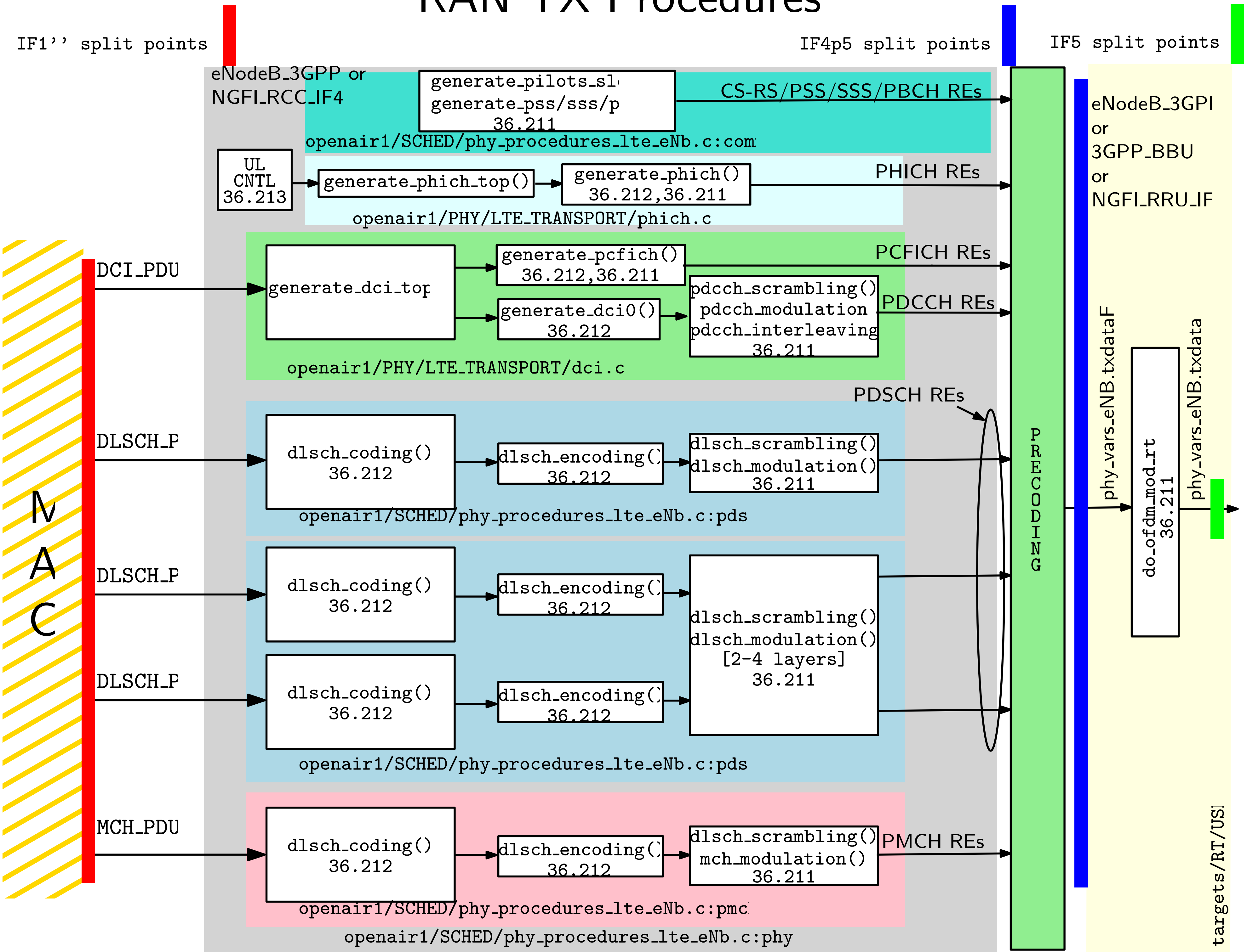
RX processing subframe n (eNB_thread_rx)

- trx_read(1 ms)
- slot_fep for slots $2n$ and $2n+1$ [phy_procedures_eNB_common_rx()]
- launch PRACH RX thread (subframe n)
  - prach_procedures
- phy_procedures_eNB_rx() (subframe n)

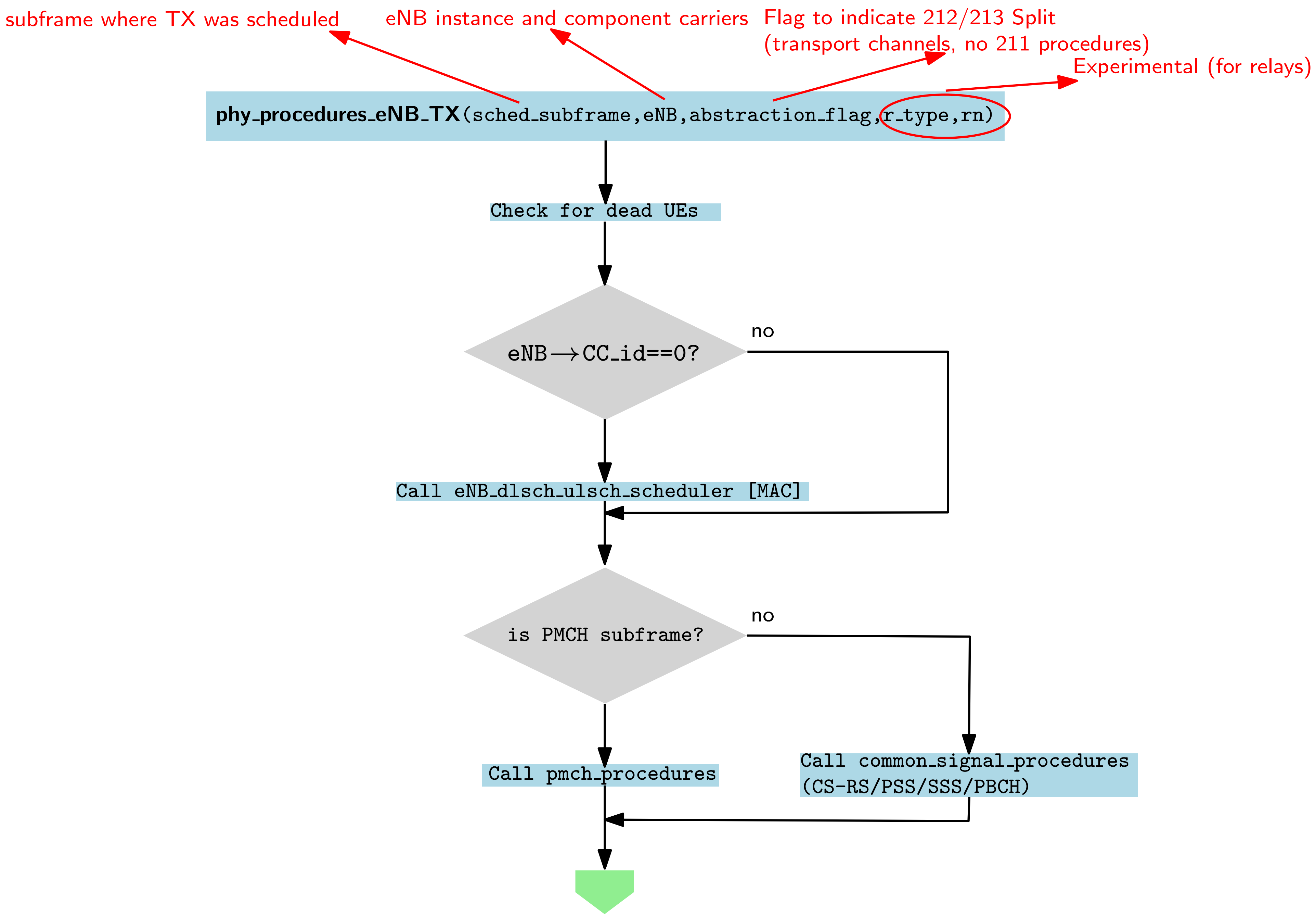
TX processing subframe $n+4$ (eNB_thread_tx)

- phy_procedures_eNB_tx
- ofdm_mod
- trx_write(1 ms)

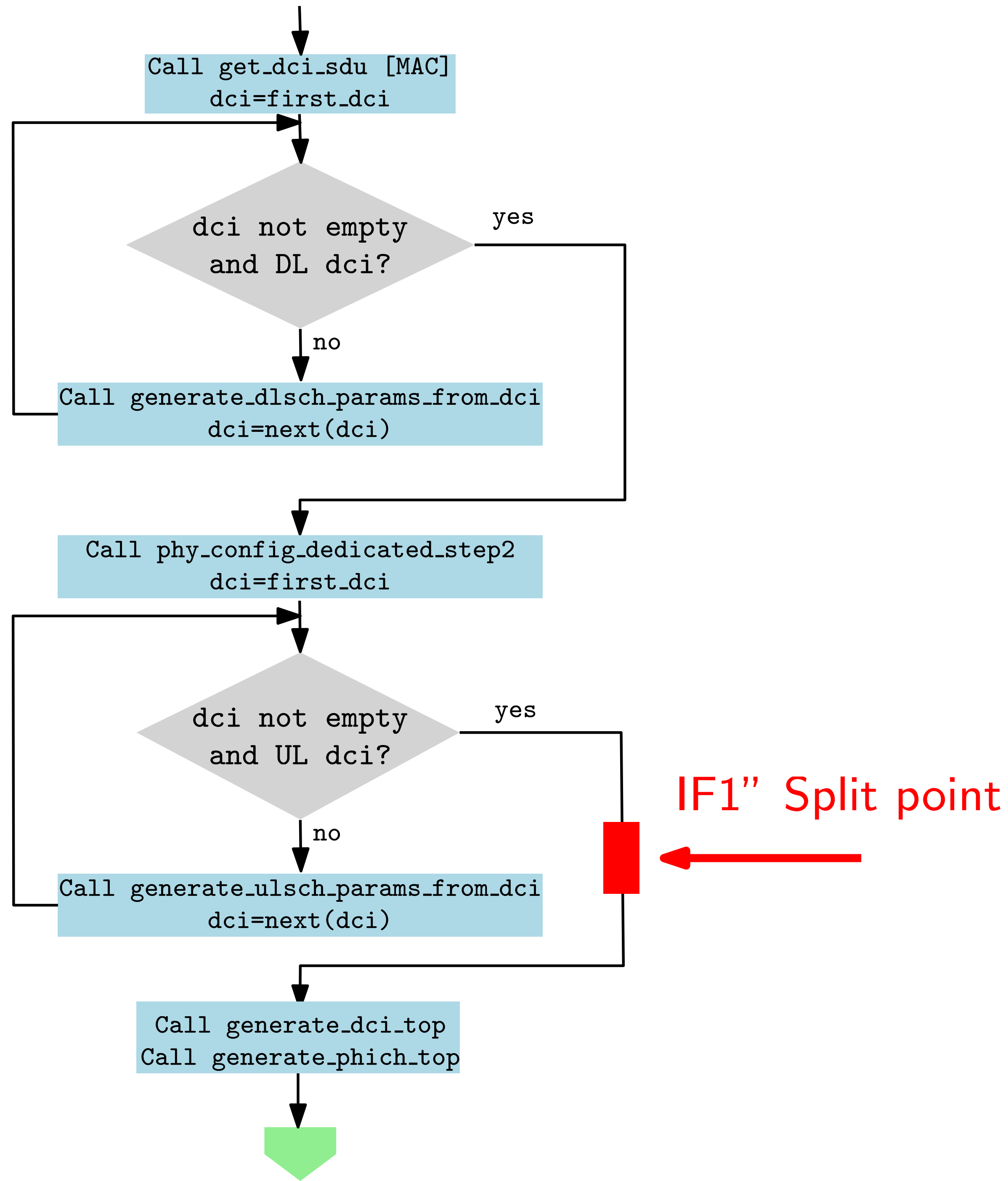
RAN TX Procedures



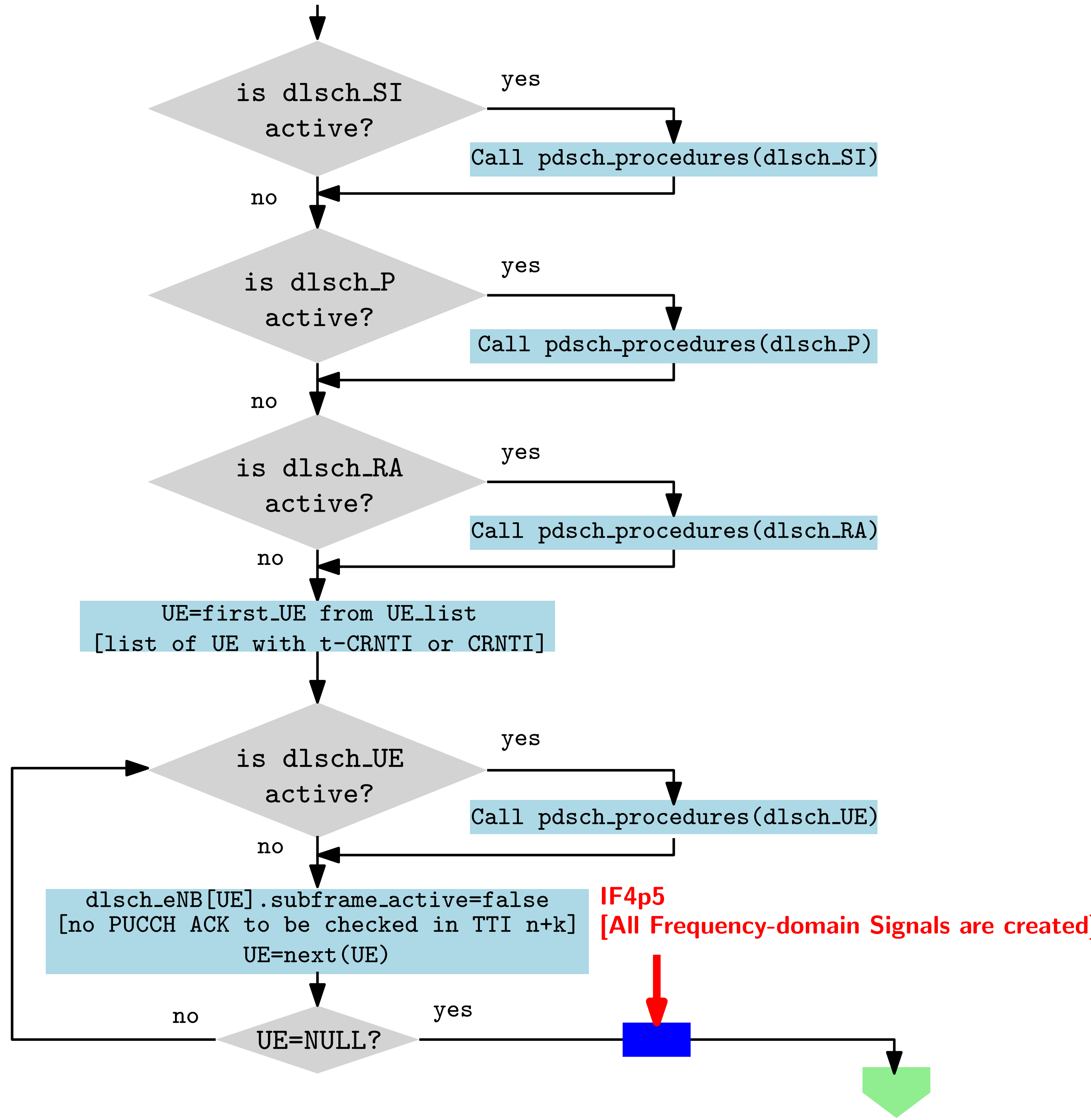
RAN TX Flowchart



RAN TX Flowchart

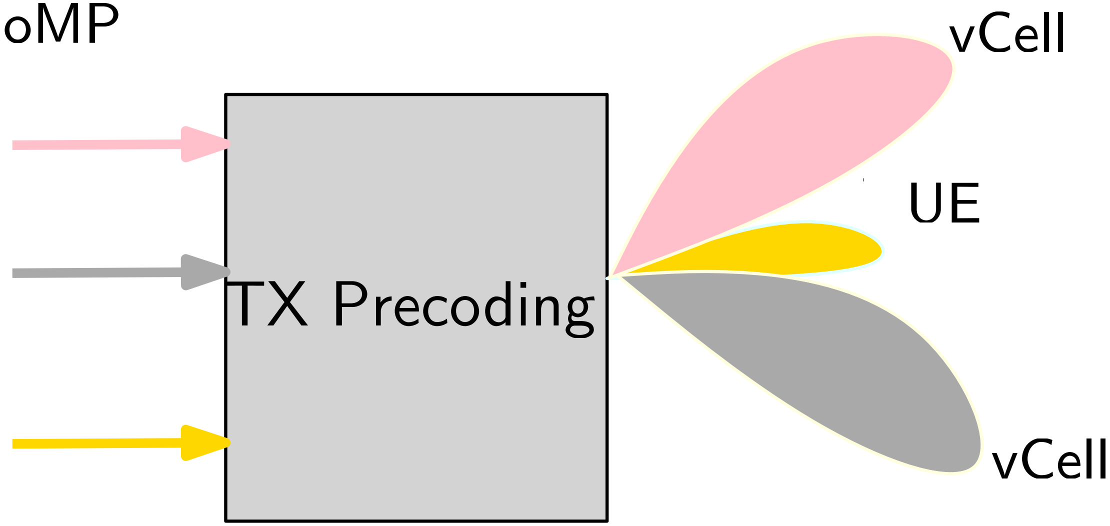


RAN TX Flowchart

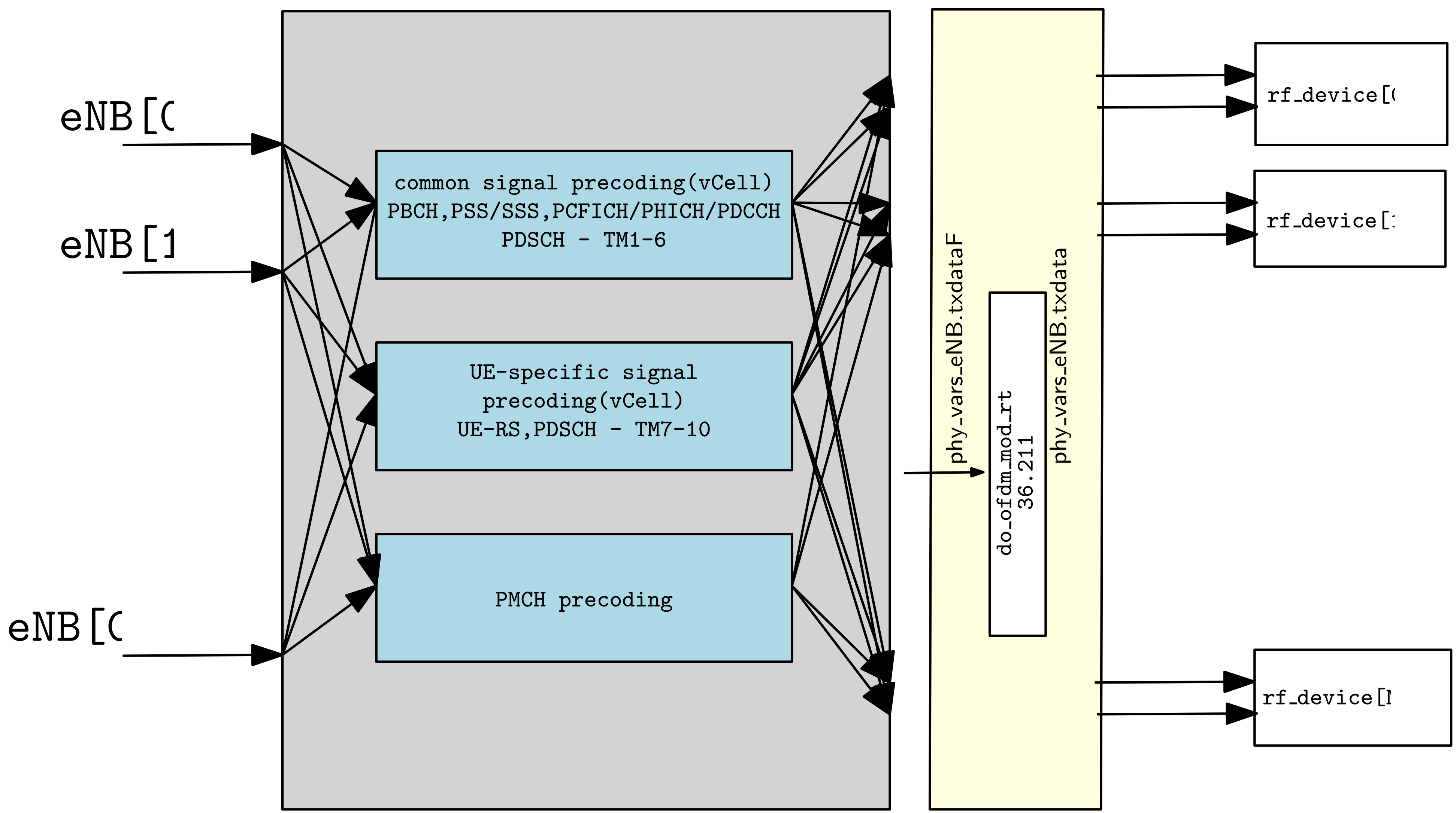


TX Precoding

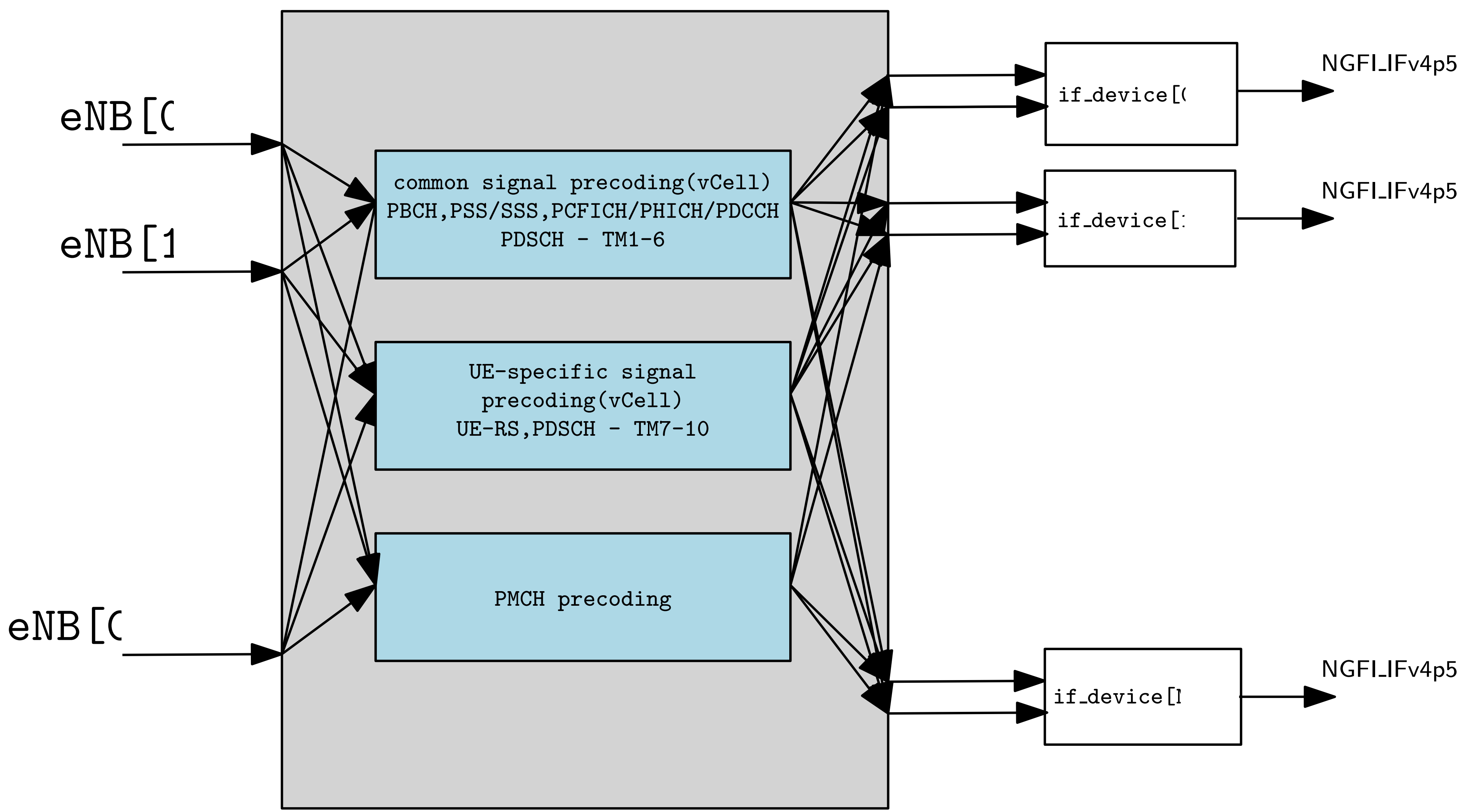
- Spatio-temporal filtering for muli-cell (vCell) and multi-user transmission. Input and output are frequency-domain signals.
- can be applied to Rel-10/11/12/13 physical channels and Rel-8 common channels
  - UE-specific precoding (TM7-10)
  - vCell-specific precoding (PDCCH + TM1-6) for groups of UEs
  - PMCH vCells
- Precoding applicable to
  1. indoor DAS
  2. outdoor co-localized arrays (e,g, Massive-MIMO)
  3. outdoor CoMP



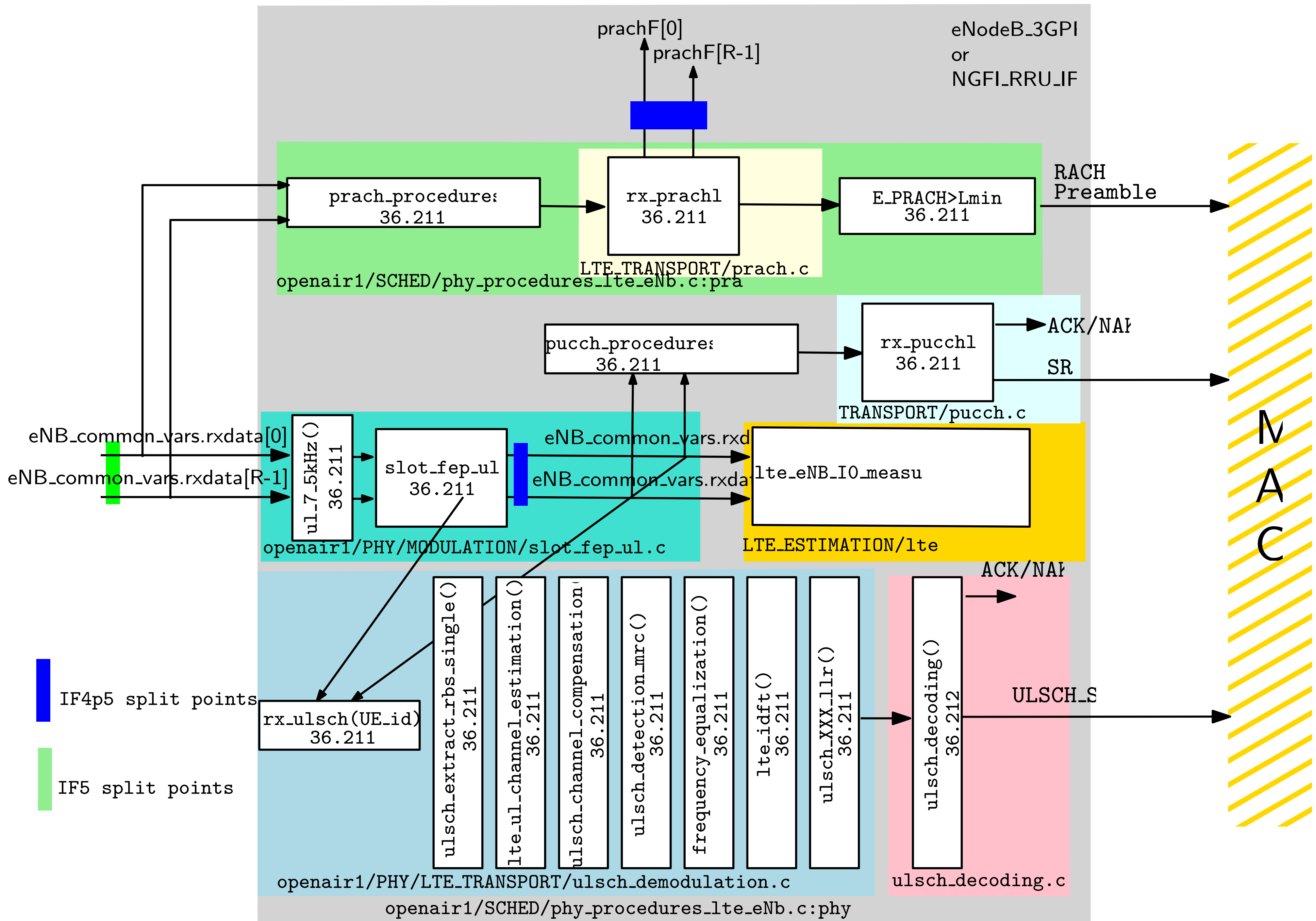
TX Precoding (to RF device)



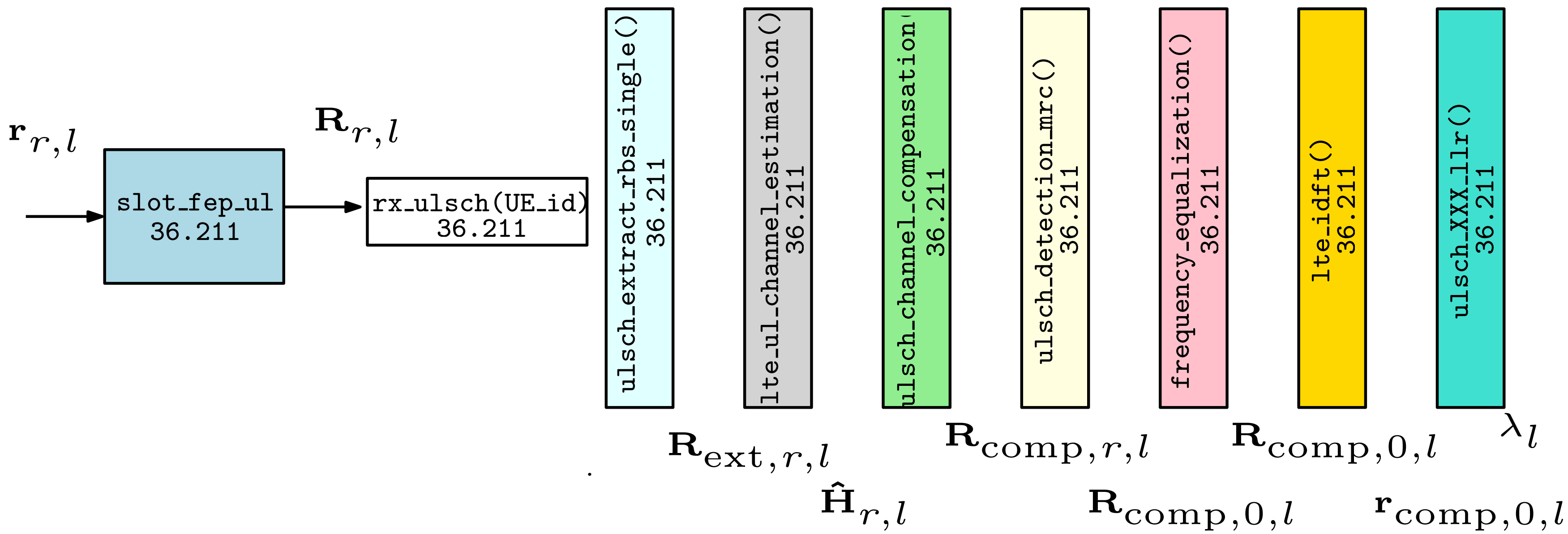
TX Precoding (to IF device, NGFI_IFv4p5)



RAN PHY RX Procedures



RAN 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[])**

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

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

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

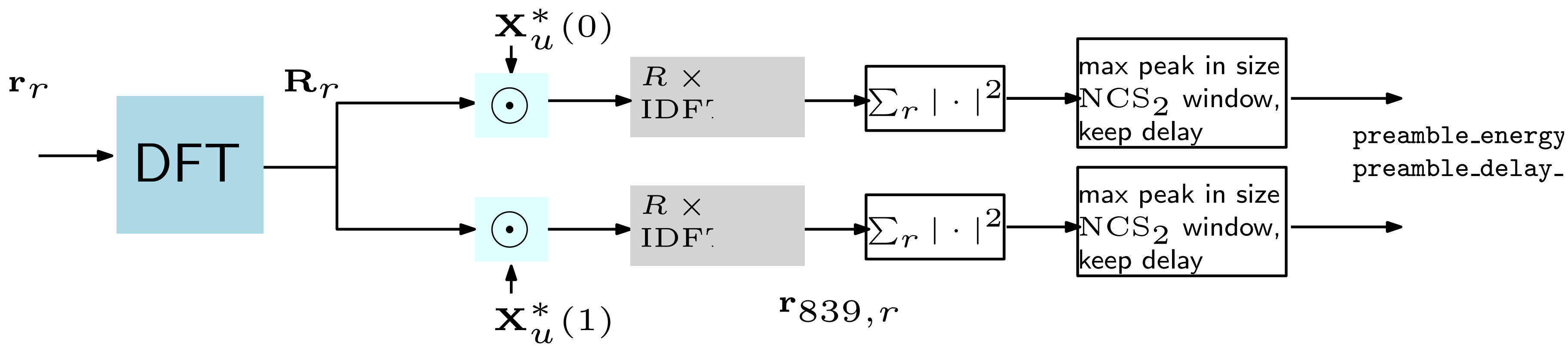
$\mathbf{r}_{\mathrm{comp},0,l} = \mathrm{DFT}_{12N_{\mathrm{PRB}}}(\mathbf{R}_{\mathrm{comp},0,l})$

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)**

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

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

RAN PRACH Detection



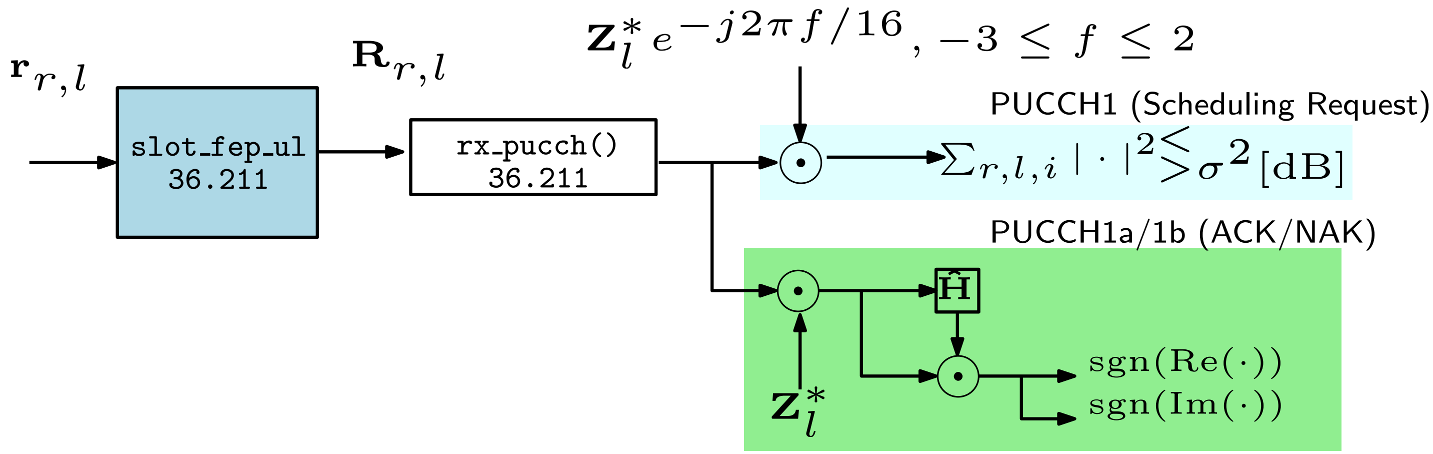
$\mathbf{R}_r = \mathrm{DFT}_{N_{\mathrm{PRACH}}}(\mathbf{r}_r), r = 0, 1, \cdots, R-1$ **(lte_eNB_prach_vars→rxsigF[])**

$\mathbf{R}_{\mathrm{comp},r} = \mathbf{R}_r \odot \mathbf{X}_u^*[i], r = 0, 1, \cdots, R-1$ **(lte_eNB_prach_vars→prachF[])**

$\mathbf{r}_{839,r} = \mathrm{IDFT}_{1024}\left(\mathbf{R}_{\mathrm{comp},r}\right), r = 0, 1, \cdots, R-1$ **(lte_eNB_prach_vars→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

RAN 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