BMWLAB OAI Project Team

NB-IoT Layer2 Design based on OAI

Advisor : Cheng, Ray-Guang

Student : Nick Ho

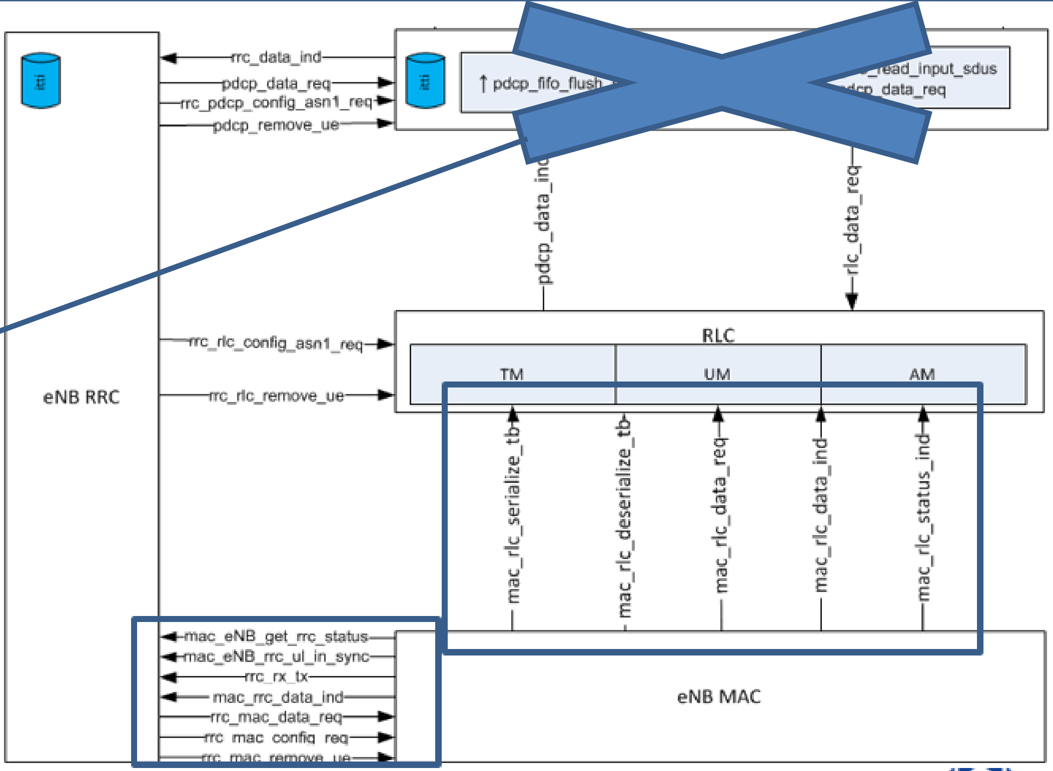
Date : 2017/02

Outline

- eNB Layer 2 Interface for NB-IoT
  - MAC-RLC
  - MAC-RRC
- eNB MAC-PHY Interface
  - Random Access
  - Phy Config
  - Scheduling
  - Helper Function
  - Event
- Protocol for NB-IoT based on OAI
  - Protocol stack (LTE vs NB-IoT)
  - Packet structure
  - Downlink Packet Transfer
  - Uplink Packet Transfer
  - OAI MAC scheduler
- MAC System Overview & Modification Phase
  - System MAC
  - Partition OAI module & Primitives
  - Partition MAC Module in detail
  - Modification Phase from OAI to NB-IoT

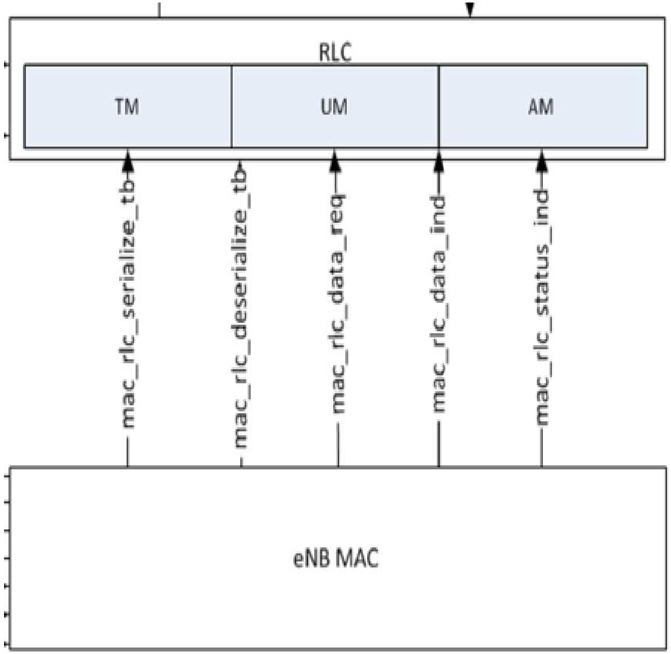
eNB layer 2 interface for NB-IoT

- Openair2/RRC/LITE
- Openair2/LAYER2/
  - MAC
  - RLC
  - PDCP



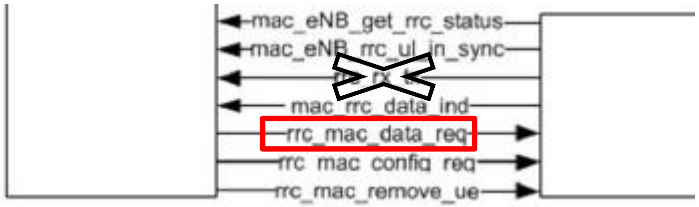
PDCP have no use for NB-IoT Control plane solution.

MAC-RLC



- mac_rlc_serialize_tb:
  - Serialize a list of Transport blocks
- mac_rlc_deserialize_tb:
  - deserialize a list of Transport blocks
- mac_rlc_data_req:
  - Delivery RLC data to MAC
- mac_rlc_data_ind:
  - Map data indication to the RLC corresponding to the radio bearer
- mac_rlc_status_ind:
  - Request and set the number of bytes scheduled for transmission by RLC

MAC-RRC



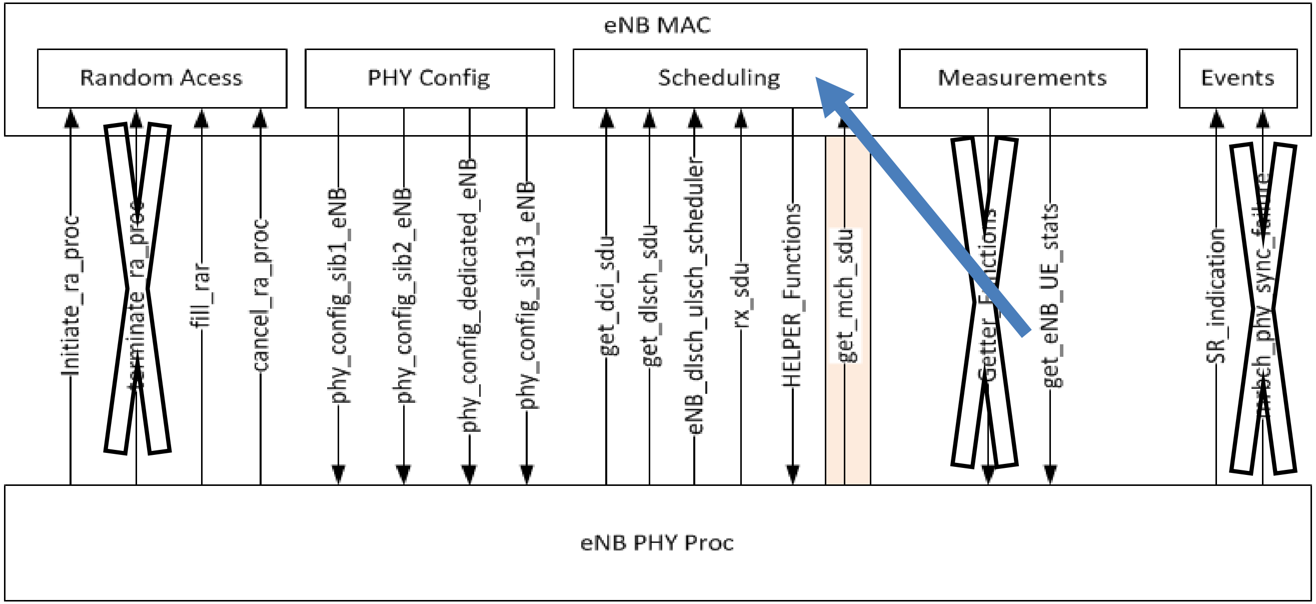
- mac_eNB_get_rrc_status
  - MAC get status of UE from RRC
- mac_eNB_rrc_ul_in_sync
  - Brief Function to remove UE when radio link failure.
- mac_rrc_data_ind
  - RRC receive data from different logical channel (MCCH BCCH CCCH).
- mac_rrc_data_req
  - For MCCH, CCCH, BCCH data delivery
- rrc_mac_config_req
  - brief RRC Configuration primitive for PHY/MAC. Allows configuration of PHY/MAC resources based on System Information (SI), RRCConnectionSetup and RRCConnectionReconfiguration messages
- rrc_mac_remove_ue
  - Check if uplink failure in scheduler procedure
  - In NB-IoT, still need check if UL fail during scheduling procedure.

Outline

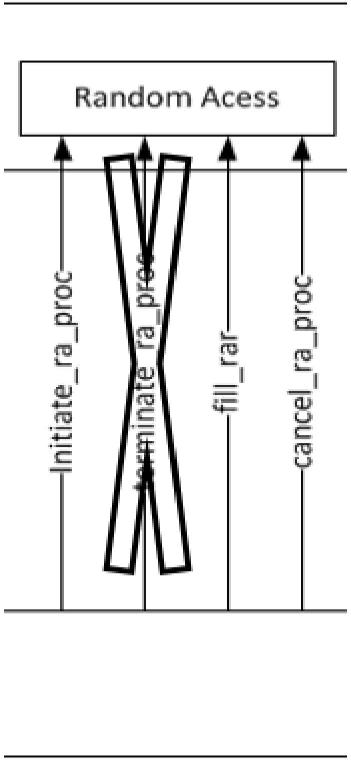
- eNB Layer 2 Interface for NB-IoT
  - MAC-RLC
  - MAC-RRC
- eNB MAC-PHY Interface
  - Random Access
  - Phy Config
  - Scheduling
  - Helper Function
  - Event
- Protocol for NB-IoT based on OAI
  - Protocol stack (LTE vs NB-IoT)
  - Packet structure
  - Downlink Packet Transfer
  - Uplink Packet Transfer
  - OAI MAC scheduler
- MAC System Overview & Modification Phase
  - System MAC
  - Partition OAI module & Primitives
  - Partition MAC Module in detail
  - Modification Phase from OAI to NB-IoT

eNB MAC-PHY interface

- Top level function
  - Openair2/PHY_INTERFACE/defs.h



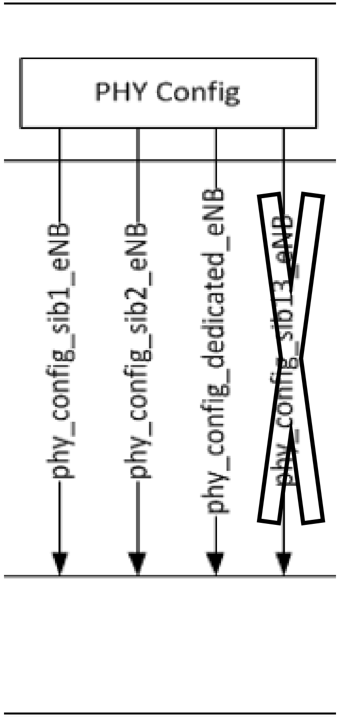
Random Access



- **[Modify]**Initiate_ra_proc:
  - Function to indicate a received preamble on PRACH. It initiates the RA procedure.
  - The preamble format has changed, and use the time and frequency domain to indicate the preamble.
- **[Modify]**fill_rar:
  - Function in eNB to fill RAR pdu when requested by PHY. This provides a single RAR SDU for the moment and returns the t-CRNTI.
  - The field of DCI for RAR and the RAR grant for MSG3 has changed.
- **[Re-use]**cancel_ra_proc:
  - Function to indicate a failed RA response. It removes all temporary variables related to the initial connection of a UE.

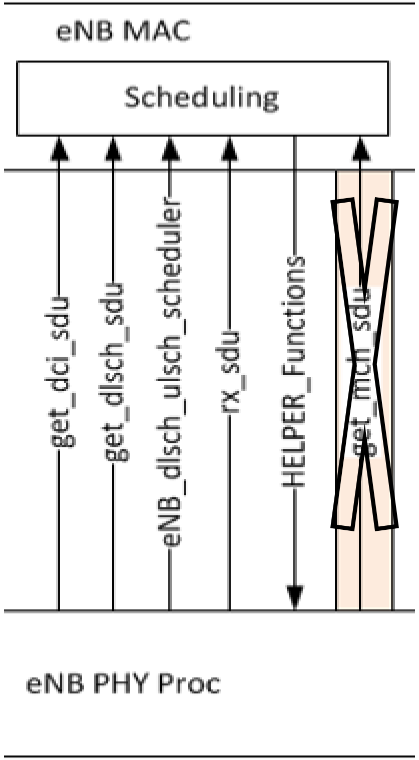
BMWLAB

PHY Config



- [Modify]Phy_config_sib1_eNB:
  - SI window size & SI period
- [Modify]Phy_config_sib2_eNB:
  - Configuration of most frame parameters & channel
- [Modify]Phy_config_dedicated_eNB:
  - Configure PHY_VARS_eNB with components of physicalConfigDedicated

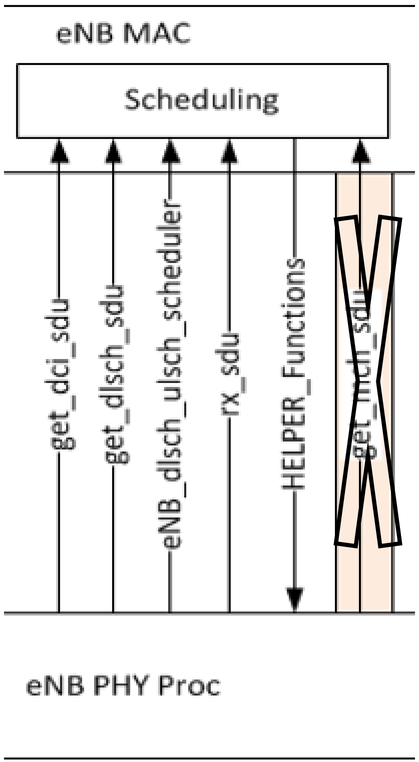
Scheduling



- **[Re-use]**get_dci_sdu:
  - retrieve result of scheduling (DCI) in current subframe. Can be called an arbitrary number of times after eNB_dlsch_ulsch_scheduler.
- **[Re-use]**get_dlsch_sdu:
  - PHY get downlink data from MAC layer.
- **[Re-use]**eNB_dlsch_ulsch_scheduler:
  - Function to trigger the eNB scheduling procedure.
- **[Re-use]**rx_sdu:
  - MAC get Uplink data from PHY layer.

BMWLAB

HELPER Function



- HELPER Function:
  - All the primitives for helping scheduling(Ex: get_TBS_UL)

HELPER Function

PHY Helper Function

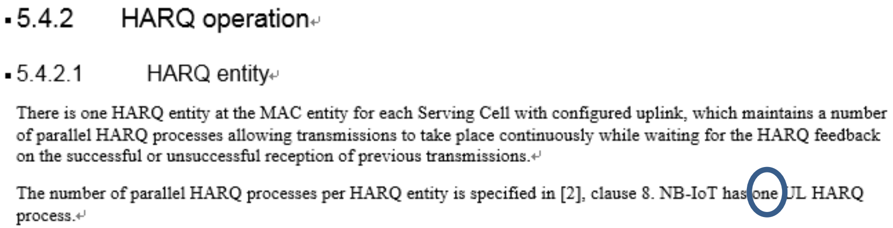
- computeRIV
- get_TBS_DL
- get_TBS_UL
- get_ue_active_harq_pid
- get_nCCE_max
- get_nCCE_offset
- get_nb_rb
- get_prb
- get_transmission_mode
- get_rballoc

MAC Helper Function

- get_eNB_UE_stats
- get_lte_frame_parms
- get_mu_mimo_mode:
- get_hundred_times_delta_TF:
- get_target_pusch_rx_power
- get_target_pucch_rx_power
- get_prach_prb_offset
- is_prach_subframe
- get_SB_size

PHY Helper Function

- [Delete]computeRIV
  - RIV computation from PHY
- [Modify]get_TBS_DL
  - Downlink TBS table lookup from PHY
- [Modify]get_TBS_UL
  - Uplink TBS table lookup from PHY
- [Modify]get_ue_active_harq_pid
  - Function to retrieve the HARQ round index for a particular UL/DLSCH and harq_pid



- 5.4.2 HARQ operation

- 5.4.2.1 HARQ entity

There is one HARQ entity at the MAC entity for each Serving Cell with configured uplink, which maintains a number of parallel HARQ processes allowing transmissions to take place continuously while waiting for the HARQ feedback on the successful or unsuccessful reception of previous transmissions.

The number of parallel HARQ processes per HARQ entity is specified in [2], clause 8. NB-IoT has one UL HARQ process.

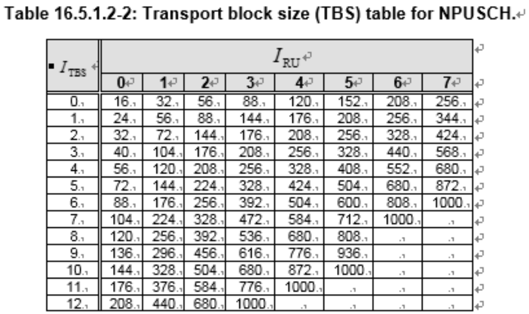


Table 16.5.1.2-2: Transport block size (TBS) table for NPUSCH.

I_{TBS}	I_{RU}							
	0	1	2	3	4	5	6	7
0	16	32	56	88	120	152	208	256
1	24	56	88	144	176	208	256	344
2	32	72	144	176	208	256	328	424
3	40	104	176	208	256	328	440	568
4	56	120	208	256	328	408	552	680
5	72	144	224	328	424	504	680	872
6	88	176	256	392	504	600	808	1000
7	104	224	328	472	584	712	1000	
8	120	256	392	536	680	808		
9	136	296	456	616	776	936		
10	144	328	504	680	872	1000		
11	176	376	584	776	1000			
12	208	440	680	1000				

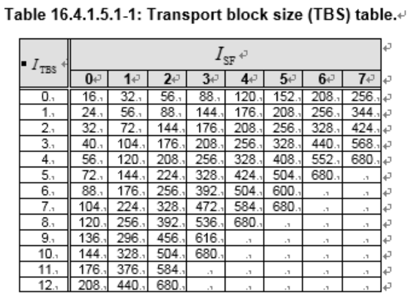


Table 16.4.1.5.1-1: Transport block size (TBS) table.

I_{TBS}	I_{SF}							
	0	1	2	3	4	5	6	7
0	16	32	56	88	120	152	208	256
1	24	56	88	144	176	208	256	344
2	32	72	144	176	208	256	328	424
3	40	104	176	208	256	328	440	568
4	56	120	208	256	328	408	552	680
5	72	144	224	328	424	504	680	
6	88	176	256	392	504	600		
7	104	224	328	472	584	680		
8	120	256	392	536	680			
9	136	296	456	616				
10	144	328	504	680				
11	176	376	584					
12	208	440	680					

BMWLAB

PHY Helper Function

- **[Delete]**get_nCCE_max
  - Function to retrieve number of CCE
- **[Delete]**get_nCCE_offset
  - Function to retrieve offset number of CCE
- **[Delete]**get_nb_rb
  - Function to retrieve number of PRB in an rb_alloc
- **[Delete]**get_prb
  - Function to convert VRB to PRB for distributed allocation

PHY Helper Function

- **[Delete]**get_transmission_mode
  - Function to retrieve transmission mode for UE
- **[Delete]**get_rballoc
  - Function to retrieve rb_alloc bitmap from dci rballoc field and VRB type

MAC Helper Function

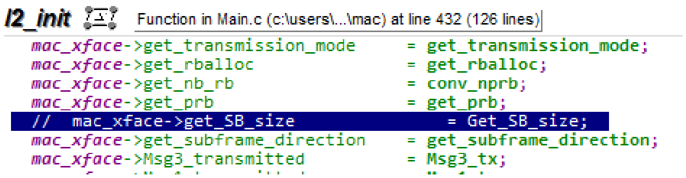
- [Modify]get_eNB_UE_stats
  - Function for MAC to get the UE stats from the PHY
  - Also used in NB-IoT because it need to get the UE stats from PHY
- [Modify]get_lte_frame_parms
  - get the frame parameters from the PHY
  - Not use in NB-IoT, it may need to design new primitives, like get_NB_frame_parm
- [Delete]get_mu_mimo_mode:
  - get the Multiuser mimo mode
  - Not use in OAI

MAC Helper Function

- **[Delete]**get_hundred_times_delta_TF:
  - get the delta TF for Uplink Power Control Calculation
  - Not use in OAI
- **[Re-use]**get_target_pusch_rx_power
  - get target PUSCH received power(this is the normalized RX power and this should be constant regardless of mcs)
  - Also used in NB-IoT because it need to know RX constant power in eNB PHY.
- **[Delete]**get_target_pucch_rx_power
  - get target PUSCH received power
  - Not use in NB-IoT

MAC Helper Function

- **[Delete]**get_prach_prb_offset:
  - Return PRACH frequency offset
- **[Delete]**is_prach_subframe:
  - Determine is PRACH subframe or not according table 5.7.1-2 from 36.211
- **[Delete]**get_SB_size
  - ICIC algos



l2_init Function in Main.c (c:\users\...\mac) at line 432 (126 lines)

```
  mac_xface->get_transmission_mode      = get_transmission_mode;
  mac_xface->get_rballoc                = get_rballoc;
  mac_xface->get_nb_rb                  = conv_nprb;
  mac_xface->get_prb                    = get_prb;
//  mac_xface->get_SB_size                  = Get_SB_size;
  mac_xface->get_subframe_direction     = get_subframe_direction;
  mac_xface->Msg3_transmitted           = Msg3_tx;
```

EX: For preamble format 0-3

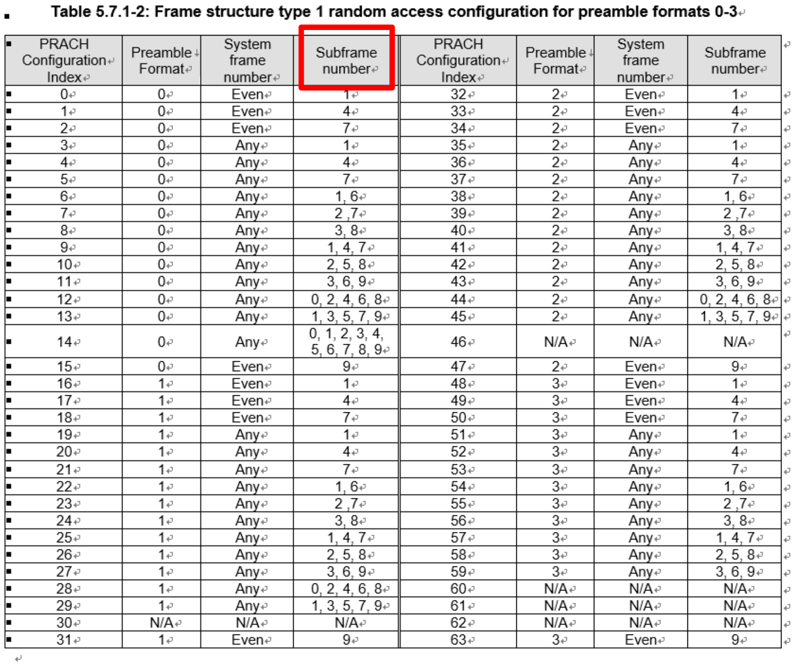
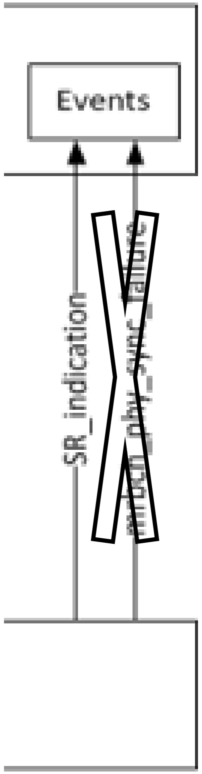


Table 5.7.1-2: Frame structure type 1 random access configuration for preamble formats 0-3

PRACH Configuration Index	Preamble Format	System frame number	Subframe number	PRACH Configuration Index	Preamble Format	System frame number	Subframe number
0	0	Even	1	32	2	Even	1
1	0	Even	4	33	2	Even	4
2	0	Even	7	34	2	Even	7
3	0	Any	1	35	2	Any	1
4	0	Any	4	36	2	Any	4
5	0	Any	7	37	2	Any	7
6	0	Any	1, 6	38	2	Any	1, 6
7	0	Any	2 ,7	39	2	Any	2 ,7
8	0	Any	3, 8	40	2	Any	3, 8
9	0	Any	1, 4, 7	41	2	Any	1, 4, 7
10	0	Any	2, 5, 8	42	2	Any	2, 5, 8
11	0	Any	3, 6, 9	43	2	Any	3, 6, 9
12	0	Any	0, 2, 4, 6, 8	44	2	Any	0, 2, 4, 6, 8
13	0	Any	1, 3, 5, 7, 9	45	2	Any	1, 3, 5, 7, 9
14	0	Any	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	46	N/A	N/A	N/A
15	0	Even	9	47	2	Even	9
16	1	Even	1	48	3	Even	1
17	1	Even	4	49	3	Even	4
18	1	Even	7	50	3	Even	7
19	1	Any	1	51	3	Any	1
20	1	Any	4	52	3	Any	4
21	1	Any	7	53	3	Any	7
22	1	Any	1, 6	54	3	Any	1, 6
23	1	Any	2 ,7	55	3	Any	2 ,7
24	1	Any	3, 8	56	3	Any	3, 8
25	1	Any	1, 4, 7	57	3	Any	1, 4, 7
26	1	Any	2, 5, 8	58	3	Any	2, 5, 8
27	1	Any	3, 6, 9	59	3	Any	3, 6, 9
28	1	Any	0, 2, 4, 6, 8	60	N/A	N/A	N/A
29	1	Any	1, 3, 5, 7, 9	61	N/A	N/A	N/A
30	N/A	N/A	N/A	62	N/A	N/A	N/A
31	1	Even	9	63	3	Even	9

Event



- **[Re-use]**SR_indication:
  - Set UE is ready to be scheduled flag for Uplink scheduling.
  - In NB-IoT, the control plane solution didn’t use SR procedure, but user plane solution support.

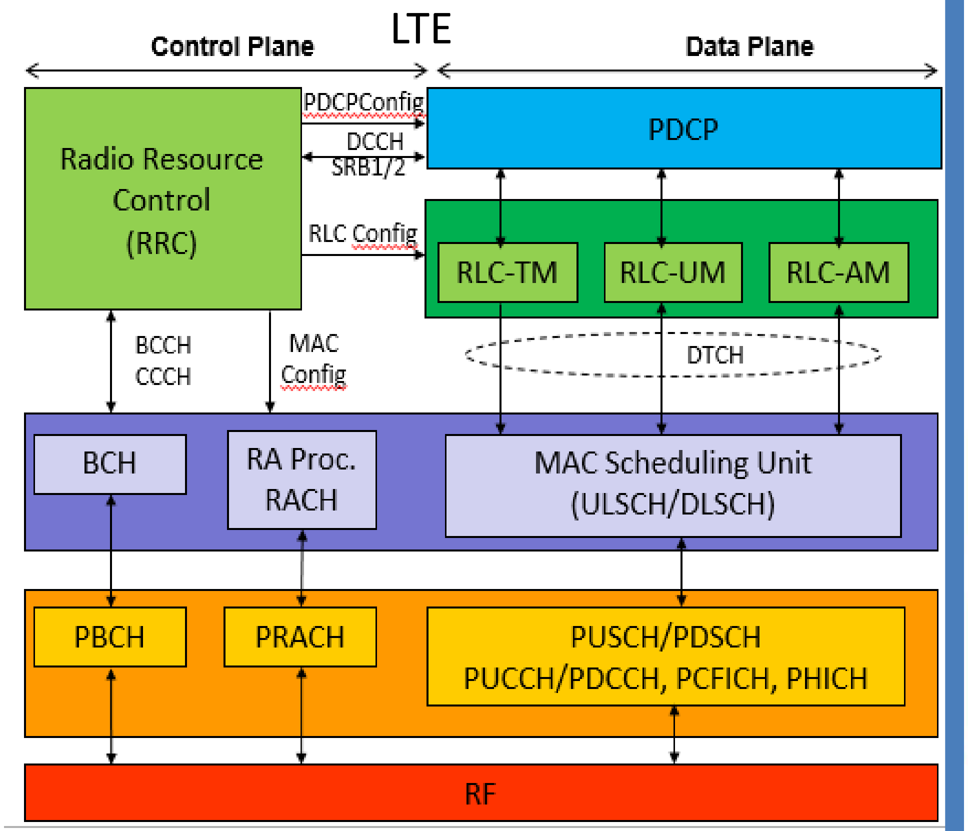
BMWLAB

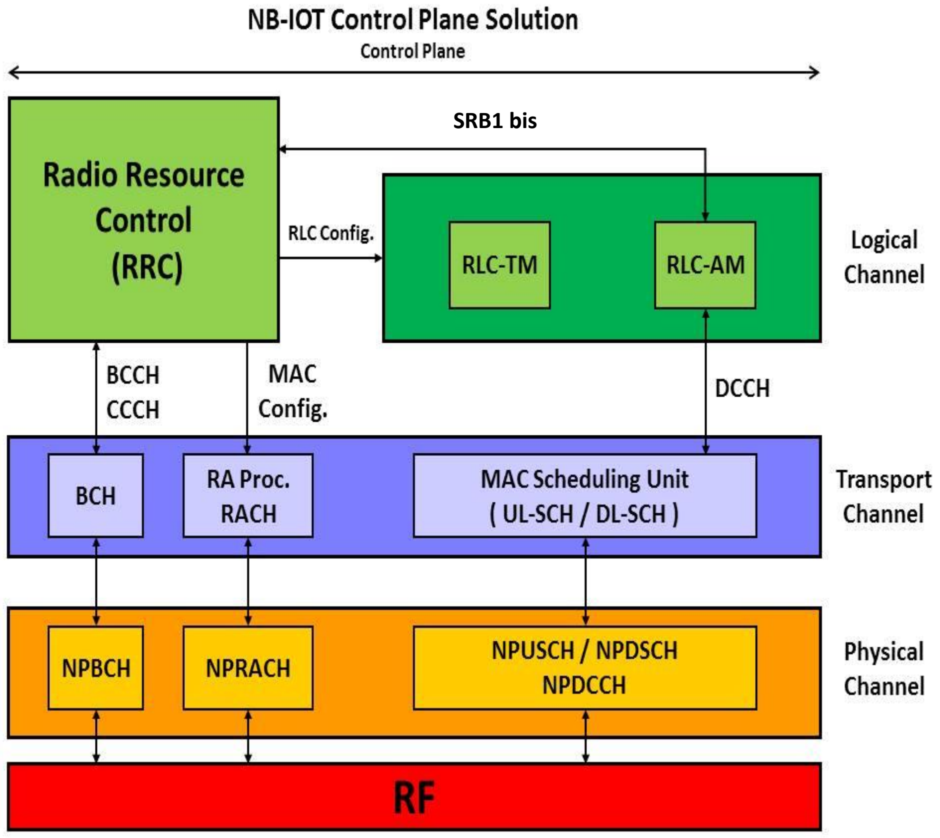
Outline

- eNB Layer 2 Interface for NB-IoT
  - MAC-RLC
  - MAC-RRC
- eNB MAC-PHY Interface
  - Random Access
  - Phy Config
  - Scheduling
  - Helper Function
  - Event
- Protocol for NB-IoT based on OAI
  - Protocol stack (LTE vs NB-IoT)
  - Packet structure
  - Downlink Packet Transfer
  - Uplink Packet Transfer
  - OAI MAC scheduler
- MAC System Overview & Modification Phase
  - System MAC
  - Partition OAI module & Primitives
  - Partition MAC Module in detail
  - Modification Phase from OAI to NB-IoT

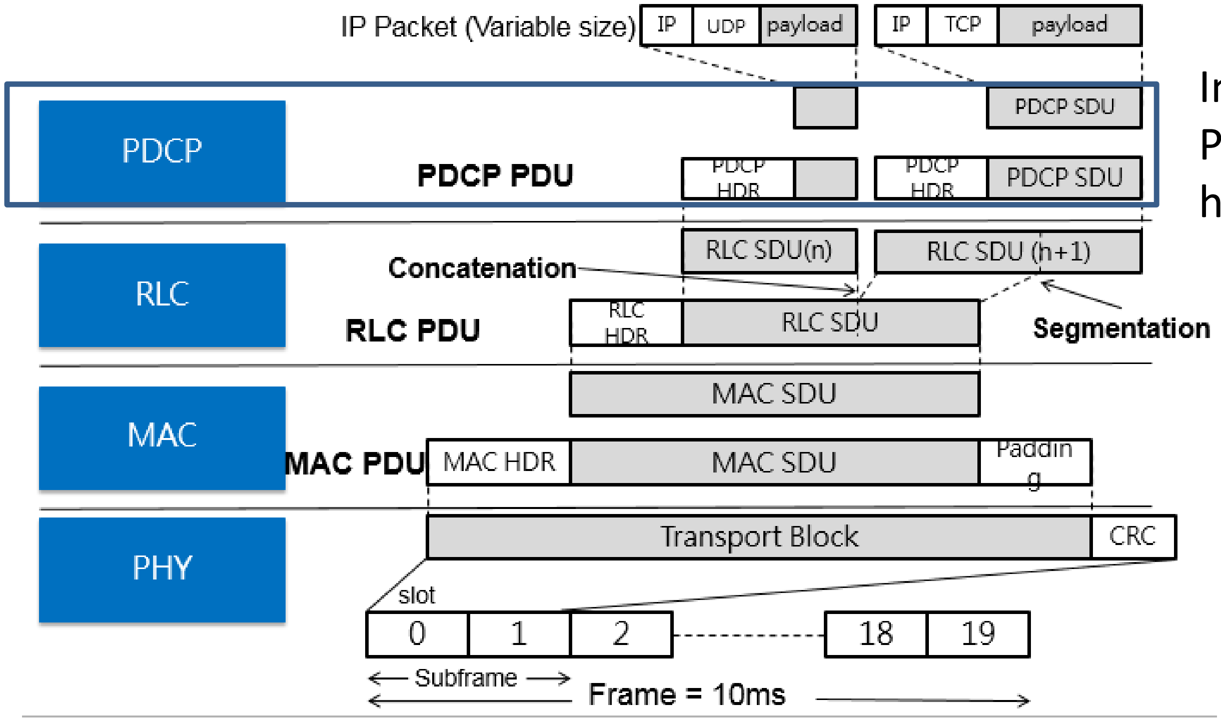
BMWLAB

Protocol stack (LTE vs NB-IoT)



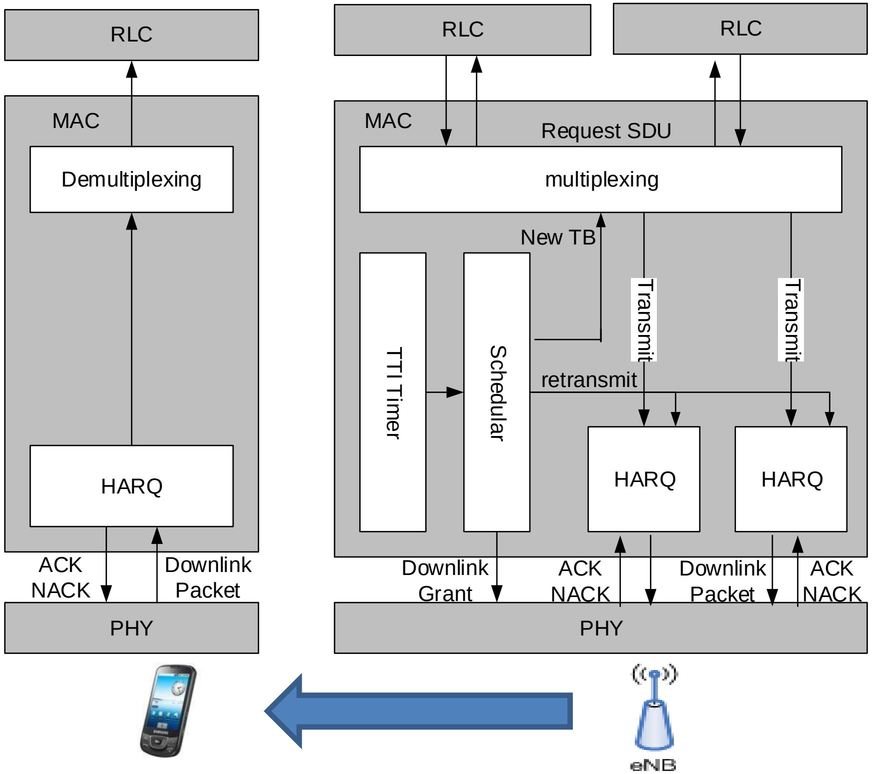


LTE packet structure



In NB-IoT Control Plane Solution, PDCP is just bypassing, not add header.

Downlink Packet Transfer



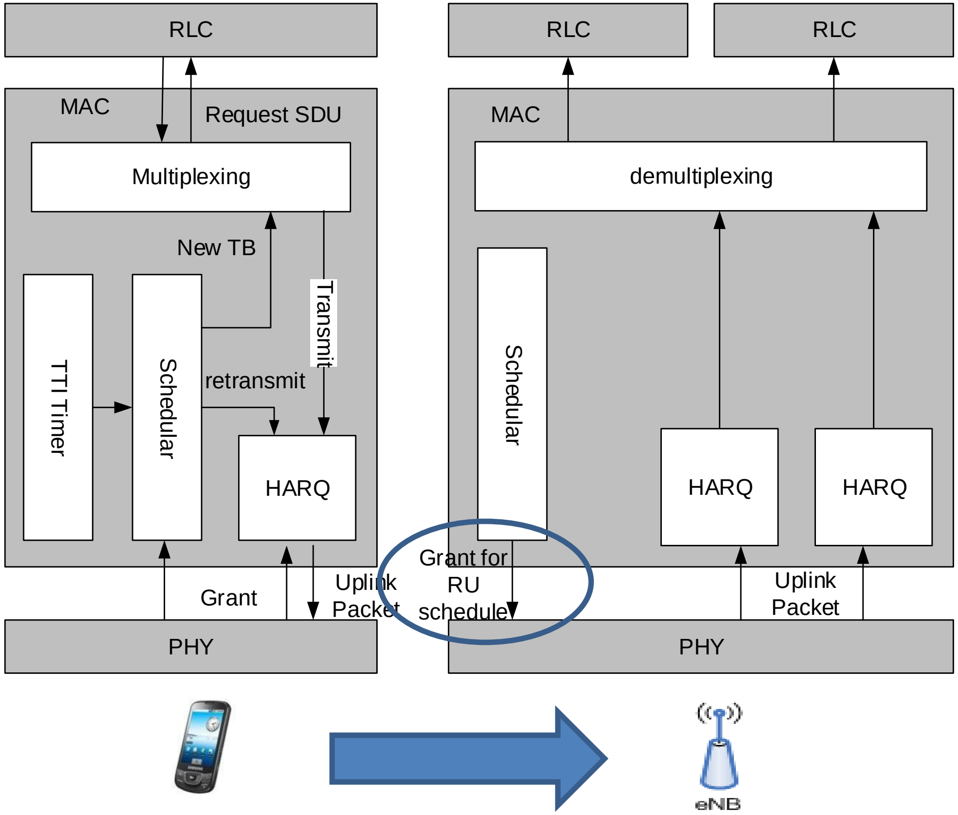
For NB-IoT:

1.In Downlink packet transmission, eNB use the new format of DCI to indicate UE the resource allocation, repetition times, resource and delay for reporting ACK/NACK and the scheduling delay.

2. Between the DCI and Downlink Packet, there is a scheduling delay.

3. Just use NPUSCH to report ACK NACK.

Uplink Packet Transfer



For NB-IoT:

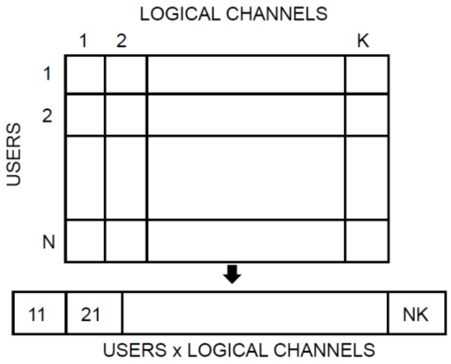
1.In Uplink packet transmission, eNB use the new format of DCI to indicate UE the resource allocation, repetition times and the scheduling delay.

2. Delete PHICH Channel. When eNB decode data, it won’t report ACK/NACK, but use the NDI field in DCI to indicate a new transmission or retransmission.

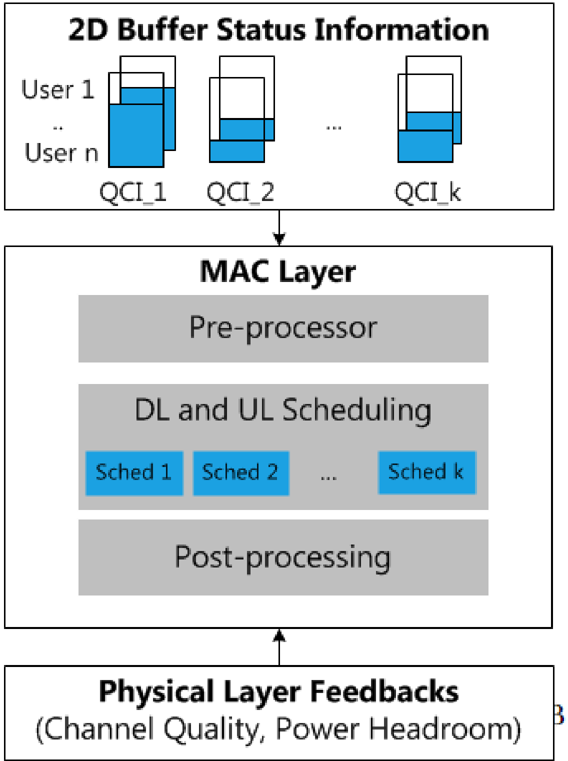
3. Use Resource Unit (RU) as a resource allocation Unit.

4. All the retransmission need control information(DCI).

OAI MAC scheduler



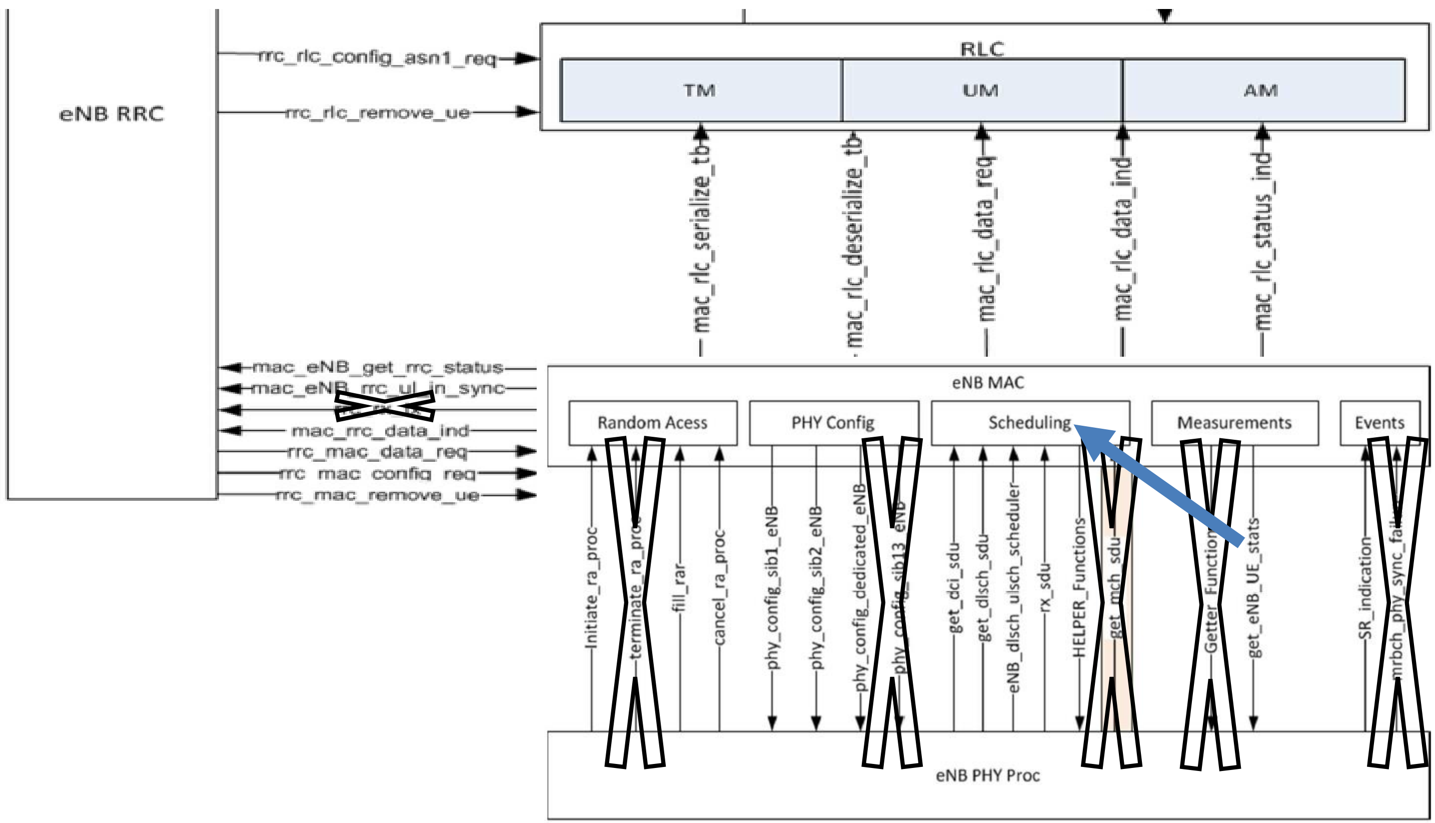
- Preprocessor
  - Converts two-dimensional buffer of users x logical channels into a single dimension as shown below
- Scheduler
  - Sorting of the blocks
- Post-processor
  - Converts the scheduling decision into the PHY format (i.e. DCI for LTE)
- Resource allocation for NB-IoT
  - Use just 1 PRB to allocate resource
  - Add scheduling delay due to the lower capability CPU
  - New format of DCI
  - Resource config between different CE level



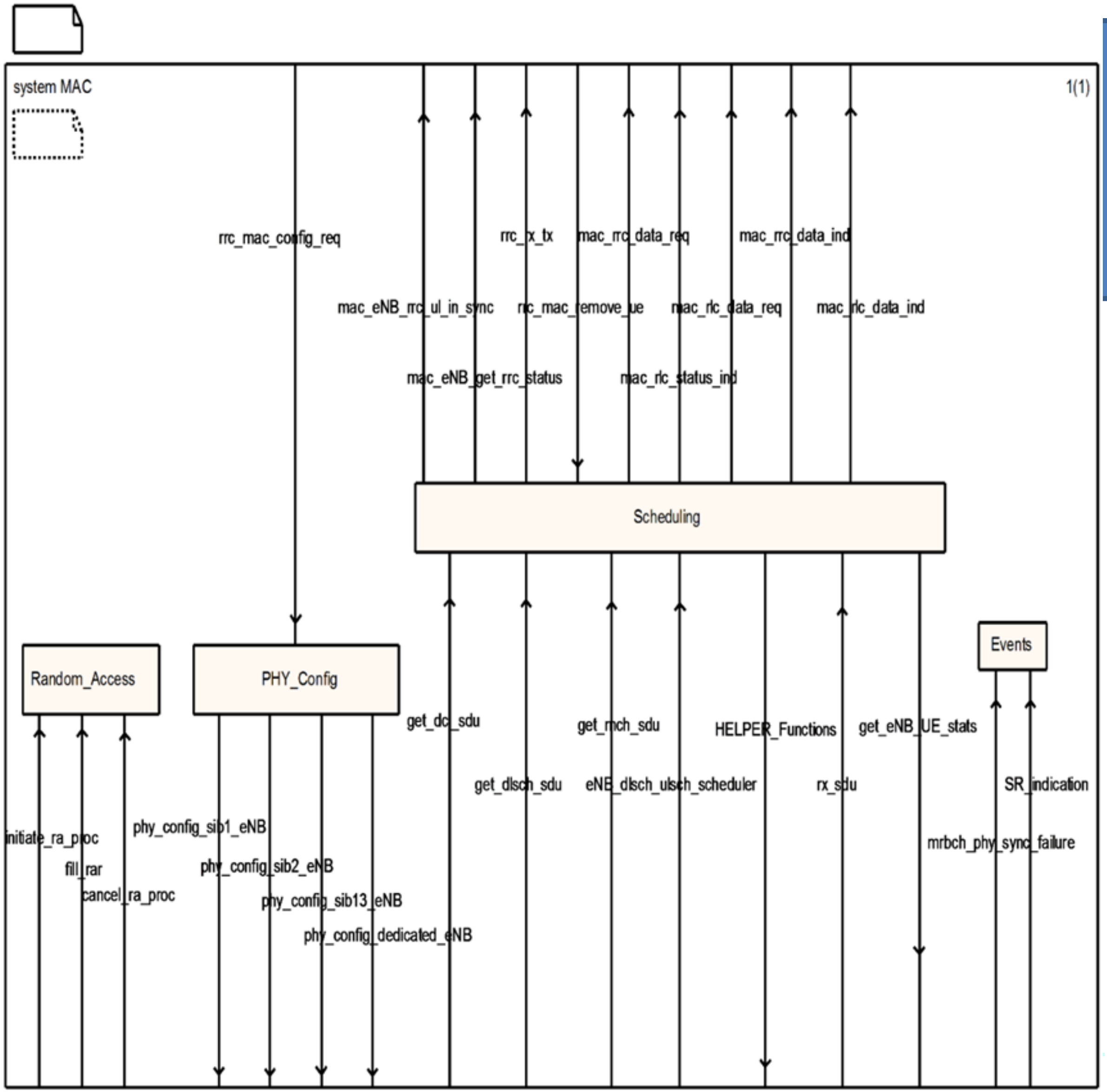
Outline

- eNB Layer 2 Interface for NB-IoT
  - MAC-RLC
  - MAC-RRC
- eNB MAC-PHY Interface
  - Random Access
  - Phy Config
  - Scheduling
  - Helper Function
  - Event
- Protocol for NB-IoT based on OAI
  - Protocol stack (LTE vs NB-IoT)
  - Packet structure
  - Downlink Packet Transfer
  - Uplink Packet Transfer
  - OAI MAC scheduler
- MAC System Overview & Modification Phase
  - System MAC
  - Partition OAI module & Primitives
  - Partition MAC Module in detail
  - Modification Phase from OAI to NB-IoT

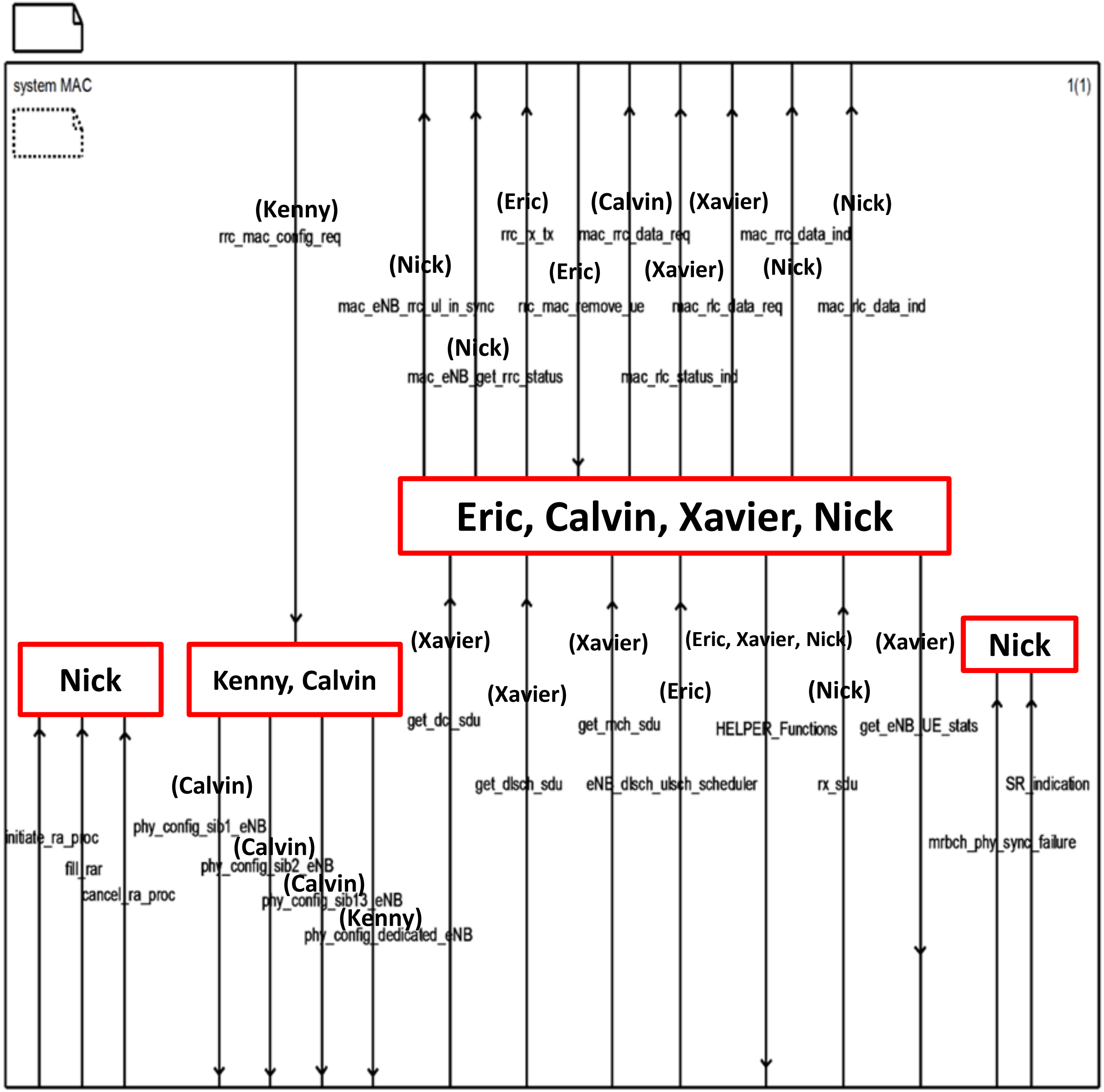
eNB layer 2 interface for NB-IoT



System MAC


system MAC
1(1)
rrc_mac_config_req
rrc_rx_tx
mac_rrc_data_req
mac_rrc_data_ind
mac_eNB_rrc_ul_in_sync
rrc_mac_remove_ue
mac_rlc_data_req
mac_rlc_data_ind
mac_eNB_get_rrc_status
mac_rlc_status_ind
Scheduling
Random_Access
PHY_Config
Events
get_dci_sdu
get_mch_sdu
HELPER_Functions
get_eNB_UE_stats
get_dlsch_sdu
eNB_dlsch_ulsch_scheduler
rx_sdu
SR_indication
mrbch_phy_sync_failure
initiate_ra_proc
fill_rar
cancel_ra_proc
phy_config_sib1_eNB
phy_config_sib2_eNB
phy_config_sib13_eNB
phy_config_dedicated_eNB

Partition OAI module & Primitives



B

Partition MAC Module in detail

MAC Module	Sub-Module		Member in charge	Note
Scheduling	UL	UL Scheduling	Eric, Nick	N/A
		UL HARQ	Nick, Kenny	
		Msg3 procedure	Nick, Eric	
	DL	DL Scheduling	Xavier, Calvin	N/A
		DL HARQ	Xavier, Calvin	
		DCI for Msg2 and schedule Msg4	Calvin, Xavier	
Random Access	Msg1 and Msg2 procedure		Nick	N/A
PHY Config	RRC Config MAC&PHY procedure except for SIB		Kenny	N/A
	SIB Config MAC&PHY procedure		Kenny	
Events	SR procedure related to UL scheduling		Nick	N/A