OpenAirInterface

MAC

SAP Design Specification

# Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Modified by | Date | Version | Comments |
| Paul | 2015/11/04 | 0.0 | Add RRC example SAP |
| Tim | 2015/12/02 | 1.0 | [Add Sample for ifdef Rel 10](#_5.1.4_ifdefine_Rel) |
| Daron | 2015/12/02 | 2.0 | 1. [Add Primitive’s Source](#_4.1.2_Source)   [Remove the previous parameters section, combine in primitives section](#_4.1.4_Parameters) |
| Nina | 2015/12/09 | 2.1 | Add 3GPP Specification Mapping in Module section |
| Cherry | 2015/12/23 | 2.2 | 1. Modify module table 2. Delete ifdef Rel 10 item |
| Eric | 2015/12/30 | 2.3 | Add ifdef Rel 10 in Parameters |
| Eric | 2016/04/07 | 2.4 | Add IND/REQ/CON/RES in Primitives Table |
| Daron | 2016/05/31 | 2.5 | Modify primitives table |
| Eric | 2016/11/09 | 2.6 | 1. Add Return Value on chapter 3.x.5  2. Add Table 3.2 Data Type in MAC SAP Primitives |
| Kenny | 2016/11/19 | 2.7 | Primitive rrc\_mac\_config\_req() |
| Eric | 2016/12/31 | 2.8 | Complete Primitives 3.11~3.14 |
| Nick | 2016/1/5 | 2.8 | 1.Complete primitives 3.15~3.19 3.22 3.26  2.Update primitives 3.3 3.9. |
| Xavier | 2016/1/6 | 2.8 | 1. 3.9 mac\_rlc\_data\_req  2. 3.10 mac\_rlc\_status\_ind |

# Table of Content

[Document Revision History 2](#_Toc474318005)

[Table of Content 3](#_Toc474318006)

[1 Reference 9](#_Toc474318007)

[2 Introduction 10](#_Toc474318008)

[2.1 Abbreviations 10](#_Toc474318009)

[2.2 Overview 10](#_Toc474318010)

[3 Primitives 12](#_Toc474318011)

[3.1 mac\_rrc\_data\_req 15](#_Toc474318012)

[3.1.1 Description 15](#_Toc474318013)

[3.1.2 Direction 15](#_Toc474318014)

[3.1.3 Source 15](#_Toc474318015)

[3.1.4 Parameters 15](#_Toc474318016)

[3.1.5 Return Value 15](#_Toc474318017)

[3.2 rrc\_mac\_config\_req 17](#_Toc474318018)

[3.2.1 Description 17](#_Toc474318019)

[3.2.2 Direction 17](#_Toc474318020)

[3.2.3 Source 17](#_Toc474318021)

[3.2.4 Parameters 17](#_Toc474318022)

[3.2.5 Return Value 18](#_Toc474318023)

[3.3 mac\_rrc\_data\_ind 19](#_Toc474318024)

[3.3.1 Description 19](#_Toc474318025)

[3.3.2 Direction 19](#_Toc474318026)

[3.3.3 Source 19](#_Toc474318027)

[3.3.4 Parameters 19](#_Toc474318028)

[3.3.5 Return Value 19](#_Toc474318029)

[3.4 mac\_eNB\_get\_rrc\_status 20](#_Toc474318030)

[3.4.1 Description 20](#_Toc474318031)

[3.4.2 Direction 20](#_Toc474318032)

[3.4.3 Source 20](#_Toc474318033)

[3.4.4 Parameters 20](#_Toc474318034)

[3.4.5 Return Value 20](#_Toc474318035)

[3.5 rrc\_mac\_remove\_ue 21](#_Toc474318036)

[3.5.1 Description 21](#_Toc474318037)

[3.5.2 Direction 21](#_Toc474318038)

[3.5.3 Source 21](#_Toc474318039)

[3.5.4 Parameters 21](#_Toc474318040)

[3.5.5 Return Value 21](#_Toc474318041)

[3.6 mac\_eNB\_rrc\_ul\_in\_sync 22](#_Toc474318042)

[3.6.1 Description 22](#_Toc474318043)

[3.6.2 Direction 22](#_Toc474318044)

[3.6.3 Source 22](#_Toc474318045)

[3.6.4 Parameters 22](#_Toc474318046)

[3.6.5 Return Value 22](#_Toc474318047)

[3.7 rrc\_rx\_tx 23](#_Toc474318048)

[3.7.1 Description 23](#_Toc474318049)

[3.7.2 Direction 23](#_Toc474318050)

[3.7.3 Source 23](#_Toc474318051)

[3.7.4 Parameters 23](#_Toc474318052)

[3.7.5 Return Value 23](#_Toc474318053)

[3.8 mac\_rlc\_data\_req 24](#_Toc474318054)

[3.8.1 Description 24](#_Toc474318055)

[3.8.2 Direction 24](#_Toc474318056)

[3.8.3 Source 24](#_Toc474318057)

[3.8.4 Parameters 24](#_Toc474318058)

[3.8.5 Return Value 25](#_Toc474318059)

[3.9 mac\_rlc\_data\_ind 26](#_Toc474318060)

[3.9.1 Description 26](#_Toc474318061)

[3.9.2 Direction 26](#_Toc474318062)

[3.9.3 Source 26](#_Toc474318063)

[3.9.4 Parameters 26](#_Toc474318064)

[3.9.5 Return Value 27](#_Toc474318065)

[3.10 mac\_rlc\_status\_ind 28](#_Toc474318066)

[3.10.1 Description 28](#_Toc474318067)

[3.10.2 Direction 28](#_Toc474318068)

[3.10.3 Source 28](#_Toc474318069)

[3.10.4 Parameters 28](#_Toc474318070)

[3.10.5 Return Value 28](#_Toc474318071)

[3.11 eNB\_dlsch\_ulsch\_scheduler 30](#_Toc474318072)

[3.11.1 Description 30](#_Toc474318073)

[3.11.2 Direction 30](#_Toc474318074)

[3.11.3 Source 30](#_Toc474318075)

[3.11.4 Parameters 30](#_Toc474318076)

[3.11.5 Return Value 30](#_Toc474318077)

[3.12 get\_dlsch\_sdu 31](#_Toc474318078)

[3.12.1 Description 31](#_Toc474318079)

[3.12.2 Direction 31](#_Toc474318080)

[3.12.3 Source 31](#_Toc474318081)

[3.12.4 Parameters 31](#_Toc474318082)

[3.12.5 Return Value 31](#_Toc474318083)

[3.13 get\_dci\_sdu 32](#_Toc474318084)

[3.13.1 Description 32](#_Toc474318085)

[3.13.2 Direction 32](#_Toc474318086)

[3.13.3 Source 32](#_Toc474318087)

[3.13.4 Parameters 32](#_Toc474318088)

[3.13.5 Return Value 32](#_Toc474318089)

[3.14 get\_mch\_sdu 33](#_Toc474318090)

[3.14.1 Description 33](#_Toc474318091)

[3.14.2 Direction 33](#_Toc474318092)

[3.14.3 Source 33](#_Toc474318093)

[3.14.4 Parameters 33](#_Toc474318094)

[3.14.5 Return Value 33](#_Toc474318095)

[3.15 initiate\_ra\_proc 34](#_Toc474318096)

[3.15.1 Description 34](#_Toc474318097)

[3.15.2 Direction 34](#_Toc474318098)

[3.15.3 Source 34](#_Toc474318099)

[3.15.4 Parameters 34](#_Toc474318100)

[3.15.5 Return Value 34](#_Toc474318101)

[3.16 fill\_rar 35](#_Toc474318102)

[3.16.1 Description 35](#_Toc474318103)

[3.16.2 Direction 35](#_Toc474318104)

[3.16.3 Source 35](#_Toc474318105)

[3.16.4 Parameters 35](#_Toc474318106)

[3.16.5 Return Value 35](#_Toc474318107)

[3.17 cancel\_ra\_proc 36](#_Toc474318108)

[3.17.1 Description 36](#_Toc474318109)

[3.17.2 Direction 36](#_Toc474318110)

[3.17.3 Source 36](#_Toc474318111)

[3.17.4 Parameters 36](#_Toc474318112)

[3.17.5 Return Value 36](#_Toc474318113)

[3.18 rx\_sdu 37](#_Toc474318114)

[3.18.1 Description 37](#_Toc474318115)

[3.18.2 Direction 37](#_Toc474318116)

[3.18.3 Source 37](#_Toc474318117)

[3.18.4 Parameters 37](#_Toc474318118)

[3.18.5 Return Value 37](#_Toc474318119)

[3.19 get\_eNB\_UE\_stats 38](#_Toc474318120)

[3.19.1 Description 38](#_Toc474318121)

[3.19.2 Direction 38](#_Toc474318122)

[3.19.3 Source 38](#_Toc474318123)

[3.19.4 Parameters 38](#_Toc474318124)

[3.19.5 Return Value 38](#_Toc474318125)

[3.20 SR\_indication 39](#_Toc474318126)

[3.20.1 Description 39](#_Toc474318127)

[3.20.2 Direction 39](#_Toc474318128)

[3.20.3 Source 39](#_Toc474318129)

[3.20.4 Parameters 39](#_Toc474318130)

[3.20.5 Return Value 39](#_Toc474318131)

[3.21 mrbch\_phy\_sync\_faliure 40](#_Toc474318132)

[3.21.1 Description 40](#_Toc474318133)

[3.21.2 Direction 40](#_Toc474318134)

[3.21.3 Source 40](#_Toc474318135)

[3.21.4 Parameters 40](#_Toc474318136)

[3.21.5 Return Value 40](#_Toc474318137)

[3.22 get\_rballoc 41](#_Toc474318138)

[3.22.1 Description 41](#_Toc474318139)

[3.22.2 Direction 41](#_Toc474318140)

[3.22.3 Source 41](#_Toc474318141)

[3.22.4 Parameters 41](#_Toc474318142)

[3.22.5 Return Value 41](#_Toc474318143)

[3.23 computeRIV 42](#_Toc474318144)

[3.23.1 Description 42](#_Toc474318145)

[3.23.2 Direction 42](#_Toc474318146)

[3.23.3 Source 42](#_Toc474318147)

[3.23.4 Parameters 42](#_Toc474318148)

[3.23.5 Return Value 42](#_Toc474318149)

[3.24 get\_ue\_active\_harq\_pid 43](#_Toc474318150)

[3.24.1 Description 43](#_Toc474318151)

[3.24.2 Direction 43](#_Toc474318152)

[3.24.3 Source 43](#_Toc474318153)

[3.24.4 Parameters 43](#_Toc474318154)

[3.24.5 Return Value 43](#_Toc474318155)

[3.25 get\_TBS 44](#_Toc474318156)

[3.25.1 Description 44](#_Toc474318157)

[3.25.2 Direction 44](#_Toc474318158)

[3.25.3 Source 44](#_Toc474318159)

[3.25.4 Parameters 44](#_Toc474318160)

[3.25.5 Return Value 44](#_Toc474318161)

[3.26 lte\_frame\_params 45](#_Toc474318162)

[3.26.1 Description 45](#_Toc474318163)

[3.26.2 Direction 45](#_Toc474318164)

[3.26.3 Source 45](#_Toc474318165)

[3.26.4 Parameters 45](#_Toc474318166)

[3.26.5 Return Value 45](#_Toc474318167)

[3.27 get\_transmission\_mode 46](#_Toc474318168)

[3.27.1 Description 46](#_Toc474318169)

[3.27.2 Direction 46](#_Toc474318170)

[3.27.3 Source 46](#_Toc474318171)

[3.27.4 Parameters 46](#_Toc474318172)

[3.27.5 Return Value 46](#_Toc474318173)

[3.28 get\_nCCE\_max 47](#_Toc474318174)

[3.28.1 Description 47](#_Toc474318175)

[3.28.2 Direction 47](#_Toc474318176)

[3.28.3 Source 47](#_Toc474318177)

[3.28.4 Parameters 47](#_Toc474318178)

[3.28.5 Return Value 47](#_Toc474318179)

[3.29 phy\_config\_sib1\_eNB 48](#_Toc474318180)

[3.29.1 Description 48](#_Toc474318181)

[3.29.2 Direction 48](#_Toc474318182)

[3.29.3 Source 48](#_Toc474318183)

[3.29.4 Parameters 48](#_Toc474318184)

[3.29.5 Return Value 48](#_Toc474318185)

[3.30 phy\_config\_sib2\_eNB 49](#_Toc474318186)

[3.30.1 Description 49](#_Toc474318187)

[3.30.2 Direction 49](#_Toc474318188)

[3.30.3 Source 49](#_Toc474318189)

[3.30.4 Parameters 49](#_Toc474318190)

[3.30.5 Return Value 49](#_Toc474318191)

[3.31 phy\_config\_sib13\_eNB 51](#_Toc474318192)

[3.31.1 Description 51](#_Toc474318193)

[3.31.2 Direction 51](#_Toc474318194)

[3.31.3 Source 51](#_Toc474318195)

[3.31.4 Parameters 51](#_Toc474318196)

[3.31.5 Return Value 51](#_Toc474318197)

[3.32 phy\_config\_dedicated\_eNB 52](#_Toc474318198)

[3.32.1 Description 52](#_Toc474318199)

[3.32.2 Direction 52](#_Toc474318200)

[3.32.3 Source 52](#_Toc474318201)

[3.32.4 Parameters 52](#_Toc474318202)

[3.32.5 Return Value 52](#_Toc474318203)

# 1 Reference

|  |  |  |  |
| --- | --- | --- | --- |
| Spec. | Release /Version | page | Related work |
| 3GPP 36.321 | Release v8.6 |  |  |
| 3GPP 36.321 | Release v13.2 |  |  |
|  |  |  |  |
|  |  |  |  |

Related work will be the chapter in this document.

# 2 Introduction

## 2.1 Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| MAC | Media access control |
| RRC | Radio Resource Control |
| PDCP | Packet Data Convergence Protocol |
| RLC | Radio Link Control |
| PHY | Physical layer |

## 2.2 Overview

Figure 1.1 shows the primitives implemented in the MAC layer and the SAPs to RRC, RLC, PHY layers

In this chapter, need to point out :

1. Other layers that is connected with MAC layer

2. The upper or lower relationship between layers

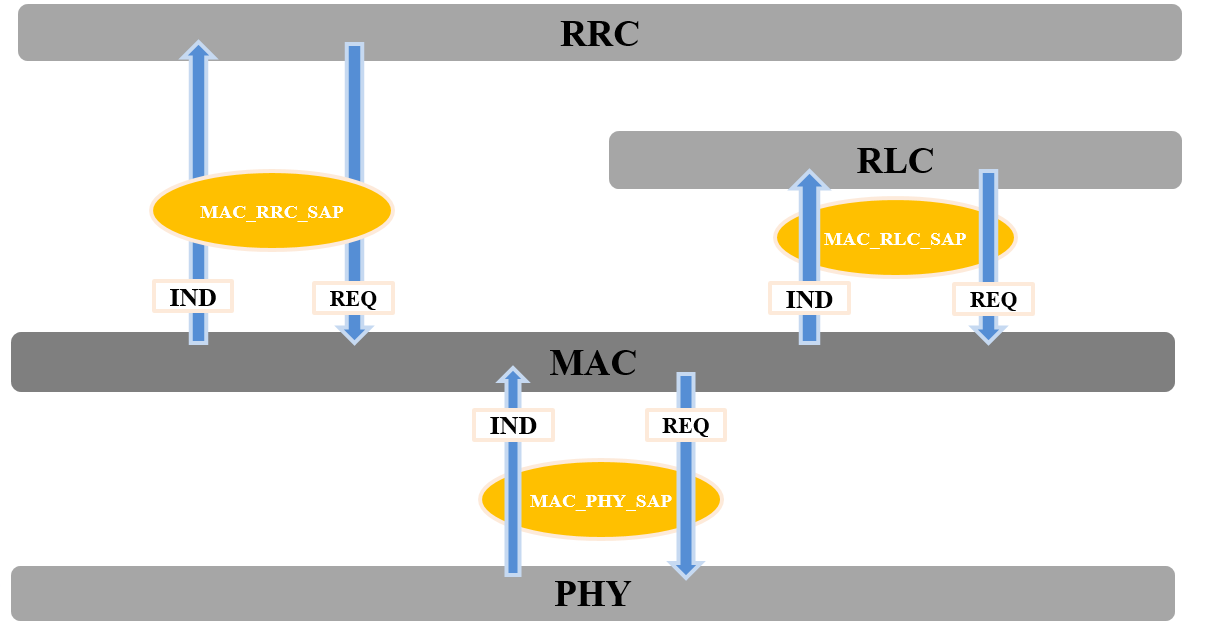
****

Fig. 1.1 Primitives and SAPs architecture.

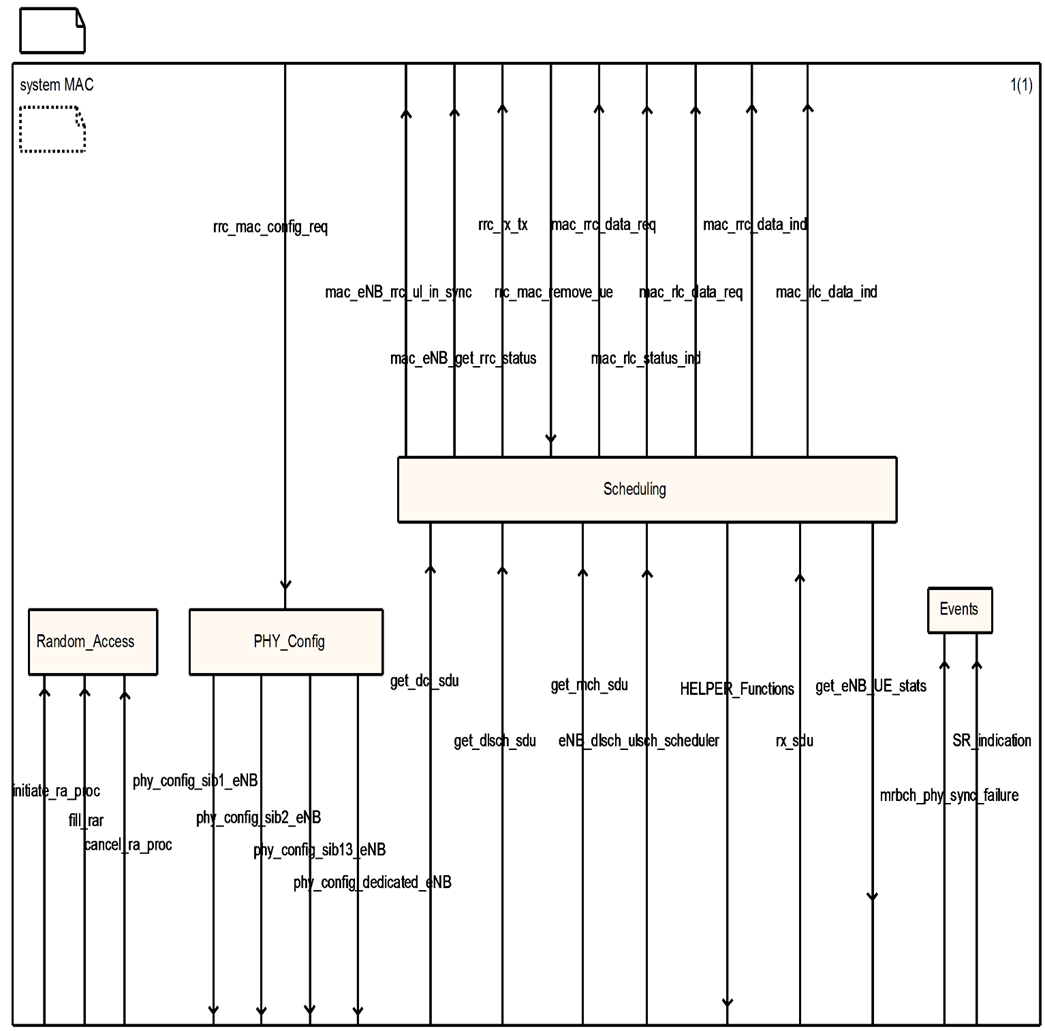
****

Fig. 1.2 Primitives and SAPs for MAC layer.

# 3 Primitives

Table 3.1 MAC SAP Primitives

|  |  |  |
| --- | --- | --- |
| Primitive | Direction | Document |
| **MAC\_RRC SAP** | | |
| 3.1 mac\_rrc\_data\_req | RRC🡪MAC | openair2\RRC\L2\_INTERFACE\  openair\_rrc\_L2\_interface.c |
| 3.2 rrc\_mac\_config\_req | RRC🡪MAC | openair2\LAYER2\MAC\  Config.c |
| 3.3 mac\_rrc\_data\_ind | MAC🡪RRC | openair2\PHY\_INTERFACE\  defs.h |
| 3.4 mac\_eNB\_get\_rrc\_status | MAC🡪RRC | openair2\RRC\LITE\  L2\_interface.c |
| 3.5 rrc\_mac\_remove\_ue | RRC🡪MAC | openair2\RRC\LITE\  L2\_interface.c |
| 3.6 mac\_eNB\_rrc\_ul\_in\_sync | MAC🡪RRC | openair2\RRC\LITE\  L2\_interface.c |
| 3.7 rrc\_rx\_tx | MAC🡪RRC | openair2\RRC\LITE\  rrc\_common.c |
| **MAC\_RLC SAP** | | |
| 3.8 mac\_rlc\_data\_req | RLC🡪MAC | openair2\LAYER2\RLC\  rlc\_mac.c |
| 3.9 mac\_rlc\_data\_ind | MAC🡪RLC | openair2\LAYER2\RLC\  rlc\_mac.c |
| 3.10 mac\_rlc\_status\_ind | MAC🡪RLC | openair2\LAYER2\RLC\  rlc\_mac.c |
| **MAC\_PHY SAP** | | |
| 3.11 eNB\_dlsch\_ulsch\_scheduler | PHY🡪MAC | openair2\LAYER2\MAC\  eNB\_dlsch\_ulsch\_scheduler.c |
| 3.12 get\_dlsch\_sdu | PHY🡪MAC | openair2\LAYER2\MAC\  eNB\_scheduler\_dlsch.c |
| 3.13 get\_dci\_sdu | PHY🡪MAC | openair2\LAYER2\MAC\  eNB\_scheduler\_primitives.c |
| 3.14 get\_mch\_sdu | PHY🡪MAC | openair2\LAYER2\MAC\  eNB\_scheduler\_mch.c |
| 3.15 initiate\_ra\_proc | PHY🡪MAC | openair2\LAYER2\MAC\  eNB\_scheduler\_RA.c |
| 3.16 fill\_rar | PHY🡪MAC | openair2\LAYER2\MAC\  rar\_tools.c |
| 3.17 cancel\_ra\_proc | PHY🡪MAC | openair2\LAYER2\MAC\  eNB\_scheduler\_RA.c |
| 3.18 rx\_sdu | PHY🡪MAC | openair2\LAYER2\MAC\  eNB\_scheduler\_ulsch.c |
| 3.19 get\_eNB\_UE\_stats | MAC🡪PHY | openair2\PHY\_INTERFACE\  defs.h |
| 3.20 SR\_indication | PHY🡪MAC | openair2\LAYER2\MAC\  eNB\_scheduler\_primitives.c |
| 3.21 mrbch\_phy\_sync\_faliure | PHY🡪MAC | openair2\LAYER2\MAC\  main.c |
| 3.22 get\_rballoc | MAC🡪PHY | openair2\PHY\_INTERFACE\  defs.h |
| 3.23 computeRIV | MAC🡪PHY | openair2\PHY\_INTERFACE\  defs.h |
| 3.24 get\_ue\_active\_harq\_pid | MAC🡪PHY | openair2\PHY\_INTERFACE\  defs.h |
| 3.25 get\_TBS | MAC🡪PHY | openair2\PHY\_INTERFACE\  defs.h |
| 3.26 lte\_frame\_params | MAC🡪PHY | openair2\PHY\_INTERFACE\  defs.h |
| 3.27 get\_transmission\_mode | MAC🡪PHY | openair2\PHY\_INTERFACE\  defs.h |
| 3.28 get\_nCCE\_max | MAC🡪PHY | openair2\PHY\_INTERFACE\  defs.h |
| 3.29 phy\_config\_sib1\_eNB | MAC🡪PHY | openair1\PHY\INIT\  lte\_init.c |
| 3.30 phy\_config\_sib2\_eNB | MAC🡪PHY | openair1\PHY\INIT\  lte\_init.c |
| 3.31 phy\_config\_sib13\_eNB | MAC🡪PHY | openair1\PHY\INIT\  lte\_init.c |
| 3.32 phy\_config\_dedicated\_eNB | MAC🡪PHY | openair2\PHY\_INTERFACE\  mac\_phy\_primitives.c |

Table 3.2 Data Type in MAC SAP Primitives

|  |  |
| --- | --- |
| **Data Type (Definition in OAI)** | **Declaration in C** |
| uint8\_t | unsigned char |
| int8\_t | signed char |
| uint16\_t | unsigned short |
| int16\_t | short |
| uint32\_t | unsigned int |
| int32\_t | int |
| rb\_id\_t | unsigned int |
| eNB\_flag\_t | signed char |
| sub\_frame\_t | unsigned int |
| module\_id\_t | unsigned char |
| frame\_t | unsigned long int |
| rnti\_t | unsigned short |
| **ASN.1** | |
| LogicalChannelConfig\_t | typedef struct LogicalChannelConfig |
| measGapConfig | typedef struct MeasGapConfig |
| TDD\_Config\_t | typedef struct TDD\_Config |
| MobilityControlInfo\_t | typedef struct MobilityControlInfo |
| struct MBSFN\_SubframeConfigList | typedef struct MBSFN\_SubframeConfigList |
| MBSFN\_AreaInfoList\_r9\_t | typedef struct MBSFN\_AreaInfoList\_r9 |
| PMCH\_InfoList\_r9\_t | typedef struct PMCH\_InfoList\_r9 |
| RadioResourceConfigCommonSIB\_t | typedef struct RadioResourceConfigCommonSIB |
| physicalConfigDedicated | typedef struct PhysicalConfigDedicated |
| ARFCN\_ValueEUTRA\_t | long |
| AdditionalSpectrumEmission\_t | long |
| SCellToAddMod\_r10\_t | typedef struct SCellToAddMod\_r10 |

## 3.1 mac\_rrc\_data\_req

### 3.1.1 Description

RRC have data to transmit to MAC layer according to different logical channels for NB-IoT.

### 3.1.2 Direction

RRC🡪MAC

### 3.1.3 Source

openair2\RRC\L2\_INTERFACE\openair\_rrc\_L2\_interface.c

### 3.1.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| Mod\_idP | Virtualized module identifier | module\_id\_t |
| CC\_id | Component carrier ID | int |
| frameP | Index of frame | frame\_t |
| Srb\_id | Index of Srb | rb\_id\_t |
| buffer\_pP | Pointer to received SDU | uint8\_t |
| enb\_flagP | Flag to indicate eNB or UE | eNB\_flag\_t |
| eNB\_index | Valid for UE indicating the index of connected eNB(s) | uint8\_t |
| **else** | | |
| Nb\_tb | Transport block | uint8\_t |
| mbsfn\_sync\_area | MBSFN sync area index | uint8\_t |

### 3.1.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| sizeof\_SIB1 | SIB1 size | uint8\_t |
| sizeof\_SIB23 | SIB23 size | uint8\_t |
| Sdu\_size | SDU size | uint8\_t |
| sizeof\_MCCH\_MESSAGE | MCCH size | uint8\_t |
| Ret\_size | Payload size | char |

## 3.2 rrc\_mac\_config\_req

### 3.2.1 Description

RRC transmit configuration parameter to MAC layer.

### 3.2.2 Direction

RRC 🡪 MAC

### 3.2.3 Source

openair2\LAYER2\MAC\Config.c

### 3.2.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| Mod\_id | Virtualized module identifier | uint8\_t |
| CC\_id | Component carrier ID | int |
| eNB\_flagP | Flag to indicate eNB or UE | eNB\_flag\_t |
| rntiP | UE RNTI number | uint16\_t |
| eNB\_index | valid for UE indicating the index of connected eNB(s) | uint8\_t |
| \*RadioResourceConfigCommonSIB-NB | Configuration parameter of radio Resource Config Common | RadioResourceConfigCommonSIB\_t |
| PhysicalConfigDedicated-NB | Configuration parameter for physical Config Dedicate | physicalConfigDedicated |
| \*MAC-MainConfig-NB | Configuration parameter for UE Mac main config | MAC\_MainConfig\_t |
| logicalChannelIdentity | If DRB-ToAddMod-NB-r13  =>Logic channel ID | LogicalChannelConfig\_t |
| \*LogicalChannelConfig-NB | Configuration parameter for logical channel | MAC\_MainConfig\_t |
| \*SIwindowsize | Length of SI window | uint8\_t |
| \*SIperiod | Period of SI message | uint16\_t |
| \*ul\_CarrierFreq | Uplink carried frequency | long |
| \*additionalSpectrumEmission | Additional spectrum emission | long |
| **else** | | |
| \*\*measObj | Measurement object to be added | MeasObjectToAddMod\_t |
| \*measGapConfig | Configuration parameter for measurement gap | MeasGapConfig\_t |
| \*tdd\_Config | Configuration parameter for TDD config | TDD\_Config\_t |
| \*mobilityControlInfo | Mobility control info received for Handover | MobilityControlInfo\_t |
| \*ul\_Bandwidth | Uplink bandwidth | long |
| \*mbsfn\_SubframeConfigList | List of MBSFN subframe config | struct MBSFN\_SubframeConfigList |
| \*sCellToAddMod\_r10 | Small cell to be added | SCellToAddMod\_r10\_t |
| MBMS\_Flag | Flag of MBMS | uint8\_t |
| \*mbsfn\_AreaInfoList | MBSFN area information list | MBSFN\_AreaInfoList\_r9\_t |
| \*pmch\_InfoList | PMCH information list | PMCH\_InfoList\_r9\_t |
| num\_active\_cba\_groups | Number of active CBA groups | uint8\_t |
| cba\_rnti | The RNTI of CBA | uint16\_t |

### 3.2.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.3 mac\_rrc\_data\_ind

### 3.3.1 Description

RRC receive data from different logical channel (MCCH, BCCH, CCCH) for NB-IoT.

### 3.3.2 Direction

MAC 🡪 RRC

### 3.3.3 Source

openair2\PHY\_INTERFACE\defs.h

### 3.3.4 Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **If define NBIoT\_R13** | | | |
| Parameter | Description | | Data Type |
| module\_idP | Virtualized module identifier | | uint8\_t |
| CC\_idP | Component carrier ID | | int |
| frameP | NB-IoT frame number | | uint32\_t |
| subframeP | NB-IoT subframe number | | uint32\_t |
| rntiP | Radio Network Temporary Identifier | | uint16\_t |
| srb\_idP | Index of Srb | | rb\_id\_t |
| \*sduP | Pointer of received Service data unit | | uint8\_t |
| sdu\_lenP | Service data unit length | | int32\_t |
| eNB\_flagP | Flag to indicate eNB or UE | | eNB\_flag\_t |
| eNB\_indexP | valid for UE indicating the index of connected eNB(s) | | uint8\_t |
| **else** | | | |
| mbsfn\_sync\_area | MBSFN sync area index | uint8\_t | |

### 3.3.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.4 mac\_eNB\_get\_rrc\_status

### 3.4.1 Description

MAC get status of UE from RRC.

### 3.4.2 Direction

MAC 🡪 RRC

### 3.4.3 Source

openair2\RRC\LITE\L2\_interface.c

### 3.4.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data Type |
| module\_idP | Virtualized module identifier | uint8\_t |
| rntiP | Radio Network Temporary Identifier | uint16\_t |
| **else** | | |
| N/A | N/A | N/A |

### 3.4.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| RRC\_status | Know if in RRC\_CONNECTED | int |

## 3.5 rrc\_mac\_remove\_ue

### 3.5.1 Description

Brief Function to remove UE when UE UL failure

### 3.5.2 Direction

RRC 🡪 MAC

### 3.5.3 Source

openair2\RRC\LITE\L2\_interface.c

### 3.5.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data type |
| mod\_idP | Virtualized module identifier | module\_id\_t |
| rntiP | UE identifier | rnti\_t |
| **else** | | |
| N/A | N/A | N/A |

### 3.5.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| 0 | if (UE\_id == -1 or ret == 0) | int |
| -1 | error in mac\_remove\_ue() | int |

## 3.6 mac\_eNB\_rrc\_ul\_in\_sync

### 3.6.1 Description

Brief Function to remove UE when UE UL failure

### 3.6.2 Direction

RRC 🡪 MAC

### 3.6.3 Source

openair2\RRC\LITE\L2\_interface.c

### 3.6.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data type |
| Mod\_instP | Virtualized module identifier | const module\_id\_t |
| CC\_idP | Component carrier ID | const int |
| frameP | NB-IoT frame number | const frame\_t |
| subframeP | NB-IoT subframe number | const sub\_frame\_t |
| rntiP | UE identifier | const rnti\_t |
| **else** | | |
| N/A | N/A | N/A |

### 3.6.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.7 rrc\_rx\_tx

### 3.7.1 Description

Brief Function to update timers every subframe. For UE it updates T300,T304 and T310.**(For now there’re no handover feature defined in NB-IoT Spec)**

### 3.7.2 Direction

MAC 🡪 RRC

### 3.7.3 Source

openair2\RRC\LITE\rrc\_common.c

### 3.7.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data type |
| ctxt\_pP | Virtualized module identifier | protocol\_ctxt\_t\* |
| enb\_indexP | UE identifier | const uint8\_t |
| CC\_id | Index of corresponding eNB/CH | const int |
| **else** | | |
| N/A | N/A | N/A |

### 3.7.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| RRC\_status\_t | Get RRC status when handover | RRC\_status\_t |

## 3.8 mac\_rlc\_data\_req

### 3.8.1 Description

Interface with MAC layer, map data request to the RLC corresponding to the radio bearer

### 3.8.2 Direction

RLC 🡪 MAC

### 3.8.3 Source

openair2\LAYER2\RLC\rlc\_mac.c

### 3.8.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data type |
| mod\_idP | Virtualized module identifier | uint8\_t |
| rntiP | UE identifier | uint16\_t |
| eNB\_index | Index of corresponding eNB/CH | eNB\_index\_t |
| frameP | Frame index | uint32\_t |
| enb\_flagP | Flag to indicate eNB (1) or UE (0) | eNB\_flag\_t |
| channel\_idP | Which logical channel MAC want to use | unsigned int |
| \*buffer\_pP | Memory area to fill with the bytes requested by MAC  Put MAC transport channel buffer to this parameter | char |
| **else** | | |
| MBMS\_flagP | Flag to indicate whether this is the MBMS service (1) or not (0) | MBMS\_flag\_t |

### 3.8.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| ret\_tb\_size | Transport block size | tbs\_size\_t |

## 3.9 mac\_rlc\_data\_ind

### 3.9.1 Description

Interface with MAC layer and RLC layer deserialize the transport blocks sent by MAC, then map data indication to the RLC instance corresponding to the radio bearer identifier

### 3.9.2 Direction

MAC 🡪 RLC

### 3.9.3 Source

openair2\LAYER2\RLC\rlc\_mac.c

### 3.9.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data type |
| mod\_idP | Virtualized module identifier | uint8\_t |
| rntiP | UE identifier | uint16\_t |
| eNB\_index |  | uint8\_t |
| frameP | Frame index | uint32\_t |
| enb\_flagP | Flag to indicate eNB (1) or UE (0) | eNB\_flag\_t |
| channel\_idP | Channel identifier | unsigned int |
| \*buffer\_pP | Memory area to fill with the bytes requested by MAC | char |
| tb\_sizeP | Size of a transport block in bits | int32\_t |
| num\_tbP | Number of transport blocks | unsigned int |
| \*crcs\_pP | Array of CRC decoding | unsigned int |
| **else** | | |
| MBMS\_flagP | Flag to indicate whether this is the MBMS service (1) or not (0) | MBMS\_flag\_t |

### 3.9.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.10 mac\_rlc\_status\_ind

### 3.10.1 Description

Interface with MAC layer and RLC layer for NB-IoT, request and set the number of bytes scheduled for transmission by the RLC instance corresponding to the radio bearer identifier

### 3.10.2 Direction

MAC 🡪 RLC

### 3.10.3 Source

openair2\LAYER2\RLC\rlc\_mac.c

### 3.10.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data Type |
| mod\_idP | Virtualized module identifier | uint8\_t |
| rntiP | UE identifier | uint16\_t |
| eNB\_index | Index of eNB | uint8\_t |
| frameP | Frame index | uint32\_t |
| enb\_flagP | Flag to indicate eNB (1) or UE (0) | eNB\_flag\_t |
| channel\_idP | Channel identifier | unsigned int |
| tb\_sizeP | Size of a transport block in bits | int32\_t |
| **else** | | |
| MBMS\_flagP | Flag to indicate whether this is the MBMS service (1) or not (0) | MBMS\_flag\_t |

### 3.10.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| mac\_rlc\_status\_resp | Status response by rlc | Structure of  mac\_rlc\_status\_resp\_t |

## 3.11 eNB\_dlsch\_ulsch\_scheduler

### 3.11.1 Description

Function to trigger the eNB scheduling procedure

### 3.11.2 Direction

PHY → MAC

### 3.11.3 Source

openair2\LAYER2\MAC\eNB\_dlsch\_ulsch\_scheduler.c

### 3.11.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| Mod\_id | Virtualized module identifier | uint8\_t |
| cooperation\_flag | Flag to indicated that this cell has cooperating nodes | uint8\_t |
| frameP | Index to current frame | frame\_t |
| subframeP | Index to current sub-frame | sub\_frame\_t |
| **else** | | |
| N/A | N/A | N/A |

### 3.11.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.12 get\_dlsch\_sdu

### 3.12.1 Description

PHY get downlink data from MAC layer.

### 3.12.2 Direction

PHY 🡪 MAC

### 3.12.3 Source

openair2\LAYER2\MAC\eNB\_scheduler\_dlsch.c

### 3.12.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| module\_idP | Mod\_id Instance ID of eNB | module\_id\_t |
| CC\_id | Component carrier ID | int |
| frameP | NB-IoT Frame number | frame\_t |
| rntiP | Scrambling type | rnti\_t |
| TBindex | Transblock size index | uint8\_t |
| **else** | | |
| N/A | N/A | N/A |

### 3.12.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| BCCH\_pdu.payload[0] | BCCH PDU payload | uint8\_t |
| DLSCH\_pdu.payload[0] | DLSCH PDU payload | int8\_t |

## 3.13 get\_dci\_sdu

### 3.13.1 Description

retrieve result of scheduling (DCI) in current subframe. Can be called an arbitrary numeber of times after eNB\_dlsch\_ulsch\_scheduler

### 3.13.2 Direction

PHY 🡪 MAC

### 3.13.3 Source

openair2\LAYER2\MAC\eNB\_scheduler\_primitives.c

### 3.13.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| module\_idP | Mod\_id Instance ID of eNB | module\_id\_t |
| CC\_id | Component carrier ID | int |
| **else** | | |
| frameP | NB-IoT Frame number | frame\_t |
| subframeP | Subframe number | Subframe\_t |

### 3.13.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| DCI\_PDU | The return value of this primitve | Struture of DCI\_PDU |

## 3.14 get\_mch\_sdu

### 3.14.1 Description

Get MCH sdu and corresponding MCS for particular MBSFN subframe.

### 3.14.2 Direction

PHY 🡪 MAC

### 3.14.3 Source

openair2\LAYER2\MAC\eNB\_scheduler\_mch.c

### 3.14.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| module\_idP | Virtualized module identifier | module\_id\_t |
| CC\_id | Component carrier ID | int |
| frameP | Index of frame | frame\_t |
| subframeP | Index of subframe | sub\_frame\_t |
| **else** | | |
| N/A | N/A | N/A |

### 3.14.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| &eNB\_mac\_inst[module\_idP].common\_channels[CC\_id].MCH\_pdu | Get MACH PDU | MCH\_PDU \* |

## 3.15 initiate\_ra\_proc

### 3.15.1 Description

Indicate a received preamble on PRACH. It initiates the RA procedure.

### 3.15.2 Direction

PHY 🡪 MAC

### 3.15.3 Source

openair2\LAYER2\MAC\eNB\_scheduler\_RA.c

### 3.15.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| module\_idP | Virtualized module identifier | module\_id\_t |
| CC\_id | Component carrier ID | int |
| frameP | Index of frame | frame\_t |
| preamble\_index | index of the received RA request | uint16\_t |
| timing\_offset | Offset in samples of the received PRACH | int16\_t |
| **else** | | |
| subframeP | Index of current subframe | sub\_frame\_t |

### 3.15.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.16 fill\_rar

### 3.16.1 Description

Fill random access response sdu, passing timing advance, and returns the t-CRNTI.

### 3.16.2 Direction

PHY 🡪 MAC

### 3.16.3 Source

openair2\LAYER2\MAC\rar\_tools.c

### 3.16.4 Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **If define NBIoT\_R13** | | | |
| **Parameter** | **Description** | | **Data Type** |
| module\_idP | Virtualized module identifier | | module\_id\_t |
| CC\_id | Component carrier ID | | int |
| frameP | Index of frame | | frame\_t |
| dlsch\_buffer | Pointer to DLSCH input buffer | | uint8\_t |
| N\_RB\_UL | Number of UL resource blocks | | uint16\_t |
| **else** | | | |
| input\_buffer\_length | PDU buffer size | uint8\_t | |

### 3.16.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| rnti | TC-RNTI | rnti\_t |

## 3.17 cancel\_ra\_proc

### 3.17.1 Description

Cancel an ongoing RA procedure. Indicate a failed RA response. It removes all temporary variables related to the initial connection of a UE.

### 3.17.2 Direction

PHY 🡪 MAC

### 3.17.3 Source

openairinterface5g-FieldTestVersion\openair2\LAYER2\MAC\eNB\_scheduler\_RA.c

### 3.17.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| module\_idP | Virtualized module identifier | module\_id\_t |
| CC\_id | Component carrier ID | int |
| frameP | Index of frame | frame\_t |
| rnti | RNTI of UE transmitting the SR | rnti\_t |
| **else** | | |
| N/A | N/A | N/A |

### 3.17.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.18 rx\_sdu

### 3.18.1 Description

This function is called by PHY, received MSG3/Uplink Data and send to MAC, it will call MAC\_RLC or MAC\_RRC SAP to send data to upper layer.

### 3.18.2 Direction

PHY 🡪 MAC

### 3.18.3 Source

openair2\LAYER2\MAC\eNB\_scheduler\_RA.c

### 3.18.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data type |
| frameP, | NB-IoT Frame number | frame\_t |
| subframeP | NB-IoT Subframe number | subframe\_t |
| rntiP | Scrambling type | rnti\_t |
| sduP | Service Data Unit | uint8\_t |
| sdu\_lenP | SDU length | uint16\_t |
| harq\_pidP | HARQ process ID | int |
| \*msg3\_flagP | Flag for Msg3 or normal uplink data | uint8\_t |
| **else** | | |
| N/A | N/A | N/A |

### 3.18.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.19 get\_eNB\_UE\_stats

### 3.19.1 Description

Function for MAC to get the UE stats from the PHY

### 3.19.2 Direction

MAC 🡪 PHY

### 3.19.3 Source

openair2\LAYER2\MAC\eNB\_scheduler\_ulsch.c

### 3.19.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data Type |
| Mod\_id | Virtualized module identifier | module\_id\_t |
| CC\_id | Component ID | uint8\_t |
| rnti | Value of rnti | rnti\_t |
| **else** | | |
|  |  |  |

### 3.19.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| UE\_id | ID of UE | uint32\_t |

## 3.20 SR\_indication

### 3.20.1 Description

Indicate Scheduling Request from UE, not support in NBIoT

### 3.20.2 Direction

PHY 🡪 MAC

### 3.20.3 Source

openair2\PHY\_INTERFACE\defs.h

### 3.20.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data Type |
| N/A | N/A | N/A |
| **else** | | |
| Mod\_id | Virtualized module identifier | module\_id\_t |
| CC\_id | Component Carrier ID | uint8\_t |
| frameP | Frame number | frame\_t |
| rnti | Value of rnti | rnti\_t |
| subframeP | Subframe number | subframe\_t |

### 3.20.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.21 mrbch\_phy\_sync\_faliure

### 3.21.1 Description

PHY Sync failure

### 3.21.2 Direction

PHY 🡪 MAC

### 3.21.3 Source

openair2\LAYER2\MAC\eNB\_scheduler\_primitives.c

### 3.21.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data Type |
| Mod\_id | Virtualized module identifier | module\_id\_t |
| CC\_id | Component Carrier ID | uint8\_t |
| rnti | Value of rnti | rnti\_t |
| **else** | | |
| N/A | N/A | N/A |

### 3.21.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| UE\_id | ID of UE | uint32\_t |

## 3.22 get\_rballoc

### 3.22.1 Description

Function to retrieve rb\_alloc bitmap from dci rballoc field and VRB type. This function may be useless due to the change of resource mapping in NB-IoT.

### 3.22.2 Direction

MAC 🡪 PHY

### 3.22.3 Source

openair2\LAYER2\MAC\main.c

### 3.22.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data Type |
| N/A | N/A | N/A |
| **else** | | |
| vrb\_type | Type of VRB mapping | vrb\_t |
| rb\_alloc\_dci | Rb allocate information in DCI feild | uint16\_t |

### 3.22.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| localRIV2alloc\_LUT25[rb\_alloc\_dci] | RB allocate type | uint32\_t |

## 3.23 computeRIV

### 3.23.1 Description

RIV computation from PHY

### 3.23.2 Direction

MAC 🡪 PHY

### 3.23.3 Source

openair2\PHY\_INTERFACE\defs.h

### 3.23.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| Parameter | Description | Data Type |
| N\_RB\_DL | Parameter to compute RIV | uint16\_t |
| RBstart | Parameter to compute RIV | uint16\_t |
| Lcrbs | Parameter to compute RIV | uint16\_t |
| **else** | | |
|  |  |  |

### 3.23.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| RIV | Value of RIV | uint16\_t |

## 3.24 get\_ue\_active\_harq\_pid

### 3.24.1 Description

Function to retrieve the HARQ round index for a particular UL/DLSCH and harq\_pid

### 3.24.2 Direction

MAC 🡪 PHY

### 3.24.3 Source

openair2\PHY\_INTERFACE\defs.h

### 3.24.4 Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **If define NBIoT\_R13** | | | | |
| **Parameter** | **Description** | | **Data Type** | |
| Mod\_id | Virtualized module identifier | | module\_id\_t | |
| CC\_id | # of RB | | uint8\_t | |
| rnti | RNTI | | rnti\_t | |
| frame | Frame | | int | |
| subframe | Subframe | | uint8\_t | |
| round | HARQ round | | uint8\_t | |
| ul\_flag | UL or not | | uint8\_t | |
| **else** | | | | |
|  | |  | |  |

### 3.24.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| 0, -1 | 0:success  -1:faild | int |

## 3.25 get\_TBS

### 3.25.1 Description

Downlink TBS table lookup from PHY

### 3.25.2 Direction

MAC 🡪 PHY

### 3.25.3 Source

openair2\PHY\_INTERFACE\defs.h

### 3.25.4 Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **If define NBIoT\_R13** | | | | |
| **Parameter** | **Description** | | **Data Type** | |
| mcs | MCS | | uint8\_t | |
| nb\_rb | # of RB | | uint16\_t | |
| **else** | | | | |
|  | |  | |  |

### 3.25.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| TBS | Value of TB size | uint32\_t |

## 3.26 lte\_frame\_params

### 3.26.1 Description

get the frame parameters from the PHY

### 3.26.2 Direction

MAC 🡪 PHY

### 3.26.3 Source

openair2\PHY\_INTERFACE\defs.h

### 3.26.4 Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **If define NBIoT\_R13** | | | | |
| **Parameter** | **Description** | | **Data Type** | |
| Mod\_id | Virtualized module identifier | | module\_id\_t | |
| CC\_id | Component Carrier ID | | Uint8\_t | |
| **else** | | | | |
|  | |  | |  |

### 3.26.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| &PHY\_vars\_eNB\_g  [Mod\_id][CC\_id]->frame\_parms | Frame parameters | LTE\_DL\_FRAME\_PARMS |

## 3.27 get\_transmission\_mode

### 3.27.1 Description

Function to retrieve transmission mode for UE

### 3.27.2 Direction

MAC 🡪 PHY

### 3.27.3 Source

openair2\PHY\_INTERFACE\defs.h

### 3.27.4 Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **If define NBIoT\_R13** | | | | |
| **Parameter** | **Description** | | **Data Type** | |
| Mod\_id | Virtualized module identifier | | module\_id\_t | |
| CC\_id | CC\_ID | | uint8\_t | |
| rnti | RNTI | | nti\_t t | |
| **else** | | | | |
|  | |  | |  |

### 3.27.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| PHY\_vars\_eNB\_g[Mod\_id][CC\_id]->  transmission\_mode[UE\_id] | Transmission mode | uint8\_t |

## 3.28 get\_nCCE\_max

### 3.28.1 Description

Function to retrieve number of CCE

### 3.28.2 Direction

MAC 🡪 PHY

### 3.28.3 Source

openair2\PHY\_INTERFACE\defs.h

### 3.28.4 Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **If define NBIoT\_R13** | | | | |
| **Parameter** | **Description** | | **Data Type** | |
| Mod\_id | Virtualized module identifier | | module\_id\_t | |
| CC\_id | Component carrier ID | | uint8\_t | |
| num\_pdcch\_symbols | # of PDCCH symbol | | int | |
| subframe | Subframe | | uint8\_t | |
| **else** | | | | |
|  | |  | |  |

### 3.28.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| get\_nCCE | Get CCE function | Function return uint16\_t |

## 3.29 phy\_config\_sib1\_eNB

### 3.29.1 Description

Configure LTE\_DL\_FRAME\_PARMS with components of SIB1 (at eNB).

### 3.29.2 Direction

MAC 🡪 PHY

### 3.29.3 Source

openair1\PHY\INIT\lte\_init.c

### 3.29.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| Mod\_id | Virtualized module identifier | uint8\_t |
| CC\_id | Component carrier ID | int |
| SIwindowsize | Length of SI window | uint8\_t |
| SIPeriod | Period of SI message | uint16\_t |
| **else** | | |
| tdd\_Config | Configuration parameter for TDD | struct MBSFN\_SubframeConfigList |

### 3.29.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.30 phy\_config\_sib2\_eNB

### 3.30.1 Description

Configure LTE\_DL\_FRAME\_PARMS with components of SIB2 (at eNB).

### 3.30.2 Direction

MAC → PHY

### 3.30.3 Source

openair1\PHY\INIT\lte\_init.c

### 3.30.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| Mod\_id | Virtualized module identifier | uint8\_t |
| CC\_id | Component carrier ID | int |
| \*radioResourceConfigCommo | Radio Configuration from SIB2 | RadioResourceConfigCommonSIB\_t |
| \*ul\_CArrierFreq | UL carrier ARFCN, null if optional | ARFCN\_ValueEUTRA\_t |
| \*ul\_Bandwidth | UL bandwidth, null if optional | long |
| \*additionalSpectrumEmission | UL parameter | AdditionalSpectrumEmission\_t |
| **else** | | |
| mbsfn\_SubframeConfigList | Pointer to mbsfn subframe configuration list from SIB2 | struct MBSFN\_SubframeConfigList |

### 3.30.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.31 phy\_config\_sib13\_eNB

### 3.31.1 Description

Configure LTE\_DL\_FRAME\_PARMS with components of SIB13 (at eNB).

### 3.31.2 Direction

MAC → PHY

### 3.31.3 Source

openair1\PHY\INIT\lte\_init.c

### 3.31.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| **else** | | |
| Mod\_id, | Virtualized module identifier | uint8\_t |
| CC\_id, | Component carrier ID | int |
| mbsfn\_Area\_idx, | MBSFN area information index | int |
| mbsfn\_AreaId\_r9 | MBSFN area information ID | long |

### 3.31.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |

## 3.32 phy\_config\_dedicated\_eNB

### 3.32.1 Description

PHY get physicalConfigDedicated from MAC layer.

### 3.32.2 Direction

MAC 🡪 PHY

### 3.32.3 Source

openair2\PHY\_INTERFACE\mac\_phy\_primitives.c

### 3.32.4 Parameters

|  |  |  |
| --- | --- | --- |
| **If define NBIoT\_R13** | | |
| **Parameter** | **Description** | **Data Type** |
| Mod\_id | Virtualized module identifier | uint8\_t |
| CC\_id | Component carrier ID | int |
| rntiP | UE RNTI number | uint16\_t |
| PhysicalConfigDedicated-NB | Configuration parameter for physical Config Dedicate | physicalConfigDedicated |
| **else** | | |
| N/A | N/A | N/A |

### 3.32.5 Return Value

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| N/A | N/A | N/A |