



JULY 2022

5G FAPI: PHY API specification

DOCUMENT 222.10.05



Small Cell Forum develops the technical and commercial enablers to accelerate small cell adoption and support the digital transformation of enterprises and communities.

Broad roll-out of small cells will make high-grade mobile connectivity accessible and affordable for industries, enterprises and for rural and urban communities. That, in turn, will drive new business opportunities for a widening ecosystem of service providers.

Our focus and work program reflects two key areas of diversification in the small cell ecosystem – the emergence of alternative deployment models such as neutral hosting and private networks, and Open RAN specifications enabling disaggregation of small cells at both component and network level.

We have driven the standardization of key elements of small cell technology including Iuh, FAPI, nFAPI, SON, services APIs, TR-069 evolution and the enhancement of the X2 interface. These specifications enable an open, multivendor platform and lower barriers to densification for all stakeholders.

Today our members are driving solutions that include:

- 5G components, products, networks
- Neutral host & multi-operator requirements
- Open RAN small cells & disaggregation
- Private networks & enterprise requirements
- Deployment and regulation
- Edge compute with small cell blueprint

All content in this document including links and references are for informational purposes only and is provided 'as is' with no warranties whatsoever including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample.

No license, express or implied, to any intellectual property rights is granted or intended hereby.

©2007-2022 All rights reserved in respect of articles, drawings, photographs etc published in hardcopy form or made available in electronic form by Small Cell Forum Ltd anywhere in the world.

The Small Cell Forum Release Program website can be found here: www.scf.io

If you would like more information about Small Cell Forum or would like to be included on our mailing list, please contact:

Email info@smallcellforum.org

Post Small Cell Forum, PO Box 23, GL11 5WA UK

Member Services memberservices@smallcellforum.org



Credits

Author	Company
Clare Somerville	Intel
Samel Celebi	Qualcomm
Jianwei Wu, Vicky Messer	Picocom
Zahi Zaltzman	Airspan
Andrei Radulescu (v3-v5)	Qualcomm Technologies, Inc
Jignesh Shah (v4)	Qualcomm Technologies, Inc
Srikant Jayaraman (v4)	Qualcomm Technologies, Inc
Abhilash Hegde (v4)	Qualcomm Technologies, Inc
Shmuel Vagner (v4-v5)	Qualcomm Technologies, Inc
Mark Wallace (v4-v5)	Qualcomm Technologies, Inc
Yaniv Elgarisi (v4)	Qualcomm Technologies, Inc
Lior Uziel (v4-v5)	Qualcomm Technologies, Inc
Doohyun Sung (v4-v5)	Qualcomm Technologies, Inc
Shubham Khandelwal (v5)	Qualcomm Technologies, Inc
Tao Zhong (v3-v5)	Picocom
Jianwei Wu (v3)	Picocom
Martin Lysejko (v5)	Picocom
Caroline Liang (v3)	NEC

Reviewer	Company
Tom Berko	Airspan
N Sreenivas	Nokia
Martin Lysejko (v1-v5)	Picocom
Francois Periard	Open Air Interface
Vicky Messer (v3, v4)	Picocom
Lior Uziel (v3)	Qualcomm Technologies, Inc
Yaniv Elgarisi (v3, v4)	Qualcomm Technologies, Inc
Reetika Aggarwal (v3, v4)	Qualcomm Technologies, Inc
Fatih Ulupinar (v3)	Qualcomm Technologies, Inc
Rudra Kumar Shivalingaiah (v3)	Qualcomm Technologies, Inc
Rashmi Tripathi (v3)	Qualcomm Technologies, Inc
Abhilash Hegde (v4-v5)	Qualcomm Technologies, Inc
Shubham Khandelwal (v4-v5)	Qualcomm Technologies, Inc
Yinan (Olivia) Li (v4)	Qualcomm Technologies, Inc
Jignesh Shah (v4-v5)	Qualcomm Technologies, Inc
Mike DeVico (v5)	Qualcomm Technologies, Inc
Murtuza Chhatriwala (v5)	Qualcomm Technologies, Inc
Shmuel Vagner (v4-v5)	Qualcomm Technologies, Inc
Loksiva Paruchuri (v5)	Qualcomm Technologies, Inc
Nitzan Ofer (v5)	Qualcomm Technologies, Inc
Ganesh Shenbagaraman (v3)	Radysis (contributor)
Tao Zhong (v3-v5)	Picocom
Vikas Dixit (v3-v5)	Jio (contributor)
Prabhakar Chitrapu	Small Cell Forum



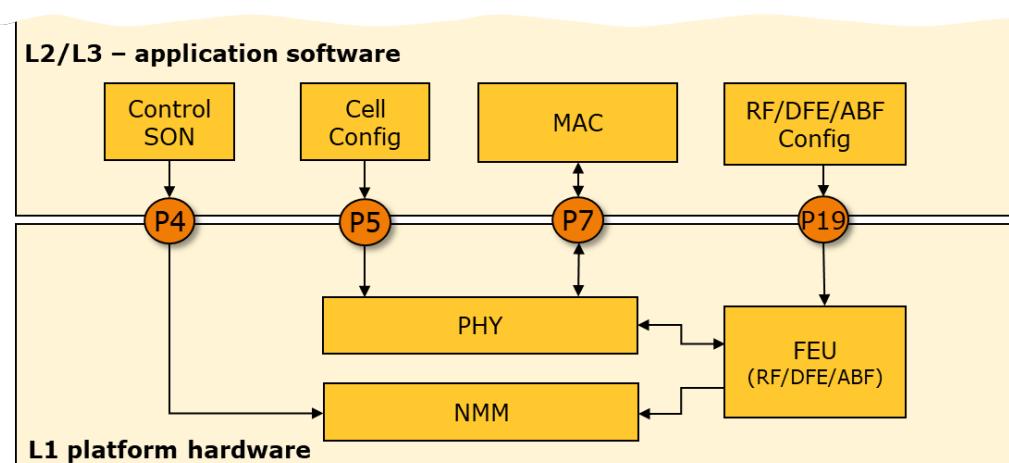
Scope

The functional application platform interface (FAPI) is an initiative within the small cell industry released by the Small Cell Forum, which establishes interoperability and innovation among suppliers of platform hardware, platform software and application software. It does so by providing a common API around which suppliers can create a competitive ecosystem. In doing this, we support a long and distinguished engineering tradition of providing an 'interchangeability of parts' to ensure that the systems vendors can take advantage of the latest innovations in silicon and software with minimum barriers to entry, and the least amount of custom re-engineering. Hence the specification helps support an innovative and competitive ecosystem for vendors of 5G small cell hardware, software and equipment.

For 5G, the FAPI suite comprises four specification documents covering the following APIs:

- '5G FAPI: PHY API' – main data path (P7) and PHY mode control (P5) interface [[SCF222](#)]
- '5G FAPI: RF and Digital Front End Control API' – (P19) for Frontend Unit control [[SCF223](#)]
- 'Network Monitor Mode API' – (P4) for 2G/3G/4G/5G [[SCF224](#)]
- '5G nFAPI Specification' [[SCF225](#)]

Small cell internal architecture



SON (Self Organising Networks), MAC (Medium Access Control), NMM (Network Monitor Mode)
FEU (Front End Unit) including DFE (Digital Front End) and ABF (Analog Beam Forming)

The 5G FAPI defines internal interfaces and is agnostic to RAN architecture.

Historical placeholder APIs P1, P2, P3, P6 and M1 are outside the scope of this document.

The table below summarizes all of SCF's FAPI API specifications, covering 5G, 4G, 3G and 2G radio access technologies (RATs).



			SCF FAPI Support				
Brand name		3GPP RAT Type [3GPP TS 29.274]	PHY API	Network Monitor Mode	RF / Digital Front End	network FAPI PHY/MAC split	Small cell (PNF/RU) management model
2G	GSM	GERAN		[SCF224]			
3G	UMTS	UTRAN	[SCF048]	[SCF224]			
3G	HSPA	HSPA Evolution	[SCF048]	[SCF224]			
4G	LTE	EUTRAN (WB-E-UTRAN)	[SCF082]	[SCF224]	[SCF082]	[SCF082]	[SCF167]
4G	LTE-NB-IoT	EUTRAN-NB-IoT	[SCF082]	[SCF224]	[SCF082]	[SCF082]	[SCF167]
4G	LTE-M	LTE-M	[SCF082]	[SCF224]	[SCF082]	[SCF082]	[SCF167]
5G	5G NR	NR	[SCF222]	[SCF224]	[SCF223]	[SCF225]	[SCF227] ¹

SCF FAPI support for different radio access technologies – the green boxes indicate new for 5G

¹ Currently in scoping



5G PHY API executive summary

This document provides the 5G PHY API specification for an application platform interface (API) between the MAC and PHY protocol layers within the small cell ecosystem. This functional application platform interface (FAPI) is an internal interface within an integrated or disaggregated small cell.

This version is for release 15 of 3GPP specification, as well select release 16 features. Among others, it supports:

- DL and UL SU-MIMO and MU-MIMO (including Massive-MIMO), based on SRS
- eMIMO[23], including mTRP, low-PAPR RS, and MU-CSI.
 - MU-CSI support: UCI payloads can be transparent to FAPI.
- 2-step RACH [24][25].
- NR positioning [27]

It also supports optimized processing:

- Control Plane/User Plane separation
- Word-aligned padding

Features relevant to multiple split architectures are also supported:

- Delay management with or without timestamps
- L2 awareness of the L1 interface between PHY and FEU
 - Number of spatial streams
 - Connectivity and synchronization events

The 5G PHY API specification defines a control interface (P5) and a user plane or data path interface (P7). The document is structured as follows:

- **Introduction:** This section provides the high-level overview and governing principles of the API.
- **PHY API Procedures:** This section details procedures to configure and control a PHY entity.
- **PHY API Messages:** This section defines the FAPI specific messages and structures to implement these procedures



Contents

1. Introduction	1
1.1 5G new radio (NR).....	1
1.2 PHY API.....	2
2. PHY API Procedures	4
2.1 Configuration Procedures.....	4
2.1.1 Initialization.....	5
2.1.2 Termination	9
2.1.3 Restart.....	10
2.1.4 Reset.....	11
2.1.5 Reconfigure	11
2.1.6 Query	13
2.1.7 Notification	14
2.1.8 Protocol Negotiation.....	14
2.1.9 PHY Instantiation.....	15
2.1.10 Common Context	18
2.1.11 PHY/FEU Interface	18
2.2 Slot Procedures	19
2.2.0 Delay Management.....	20
2.2.1 SLOT signal	22
2.2.2 SFN/SL Synchronization	23
2.2.3a Slot Numerology	28
2.2.3 API message order	29
2.2.4 Downlink	32
2.2.5 Uplink	42
2.2.6 Spatial Multiplexing	53
2.2.7 Error sequences	59
3. PHY API Messages.....	62
3.1 General Message Format	62
3.1.1 Simplified Message Format	62
3.1.2 nFAPI Message Format.....	63
3.1.3 Padding	64
3.2 Message Types.....	64
3.3 Configuration Messages.....	65
3.3.1 PARAM	65
3.3.2 CONFIG.....	112
3.3.3 Vendor TLVs	140



3.3.4	START.....	140
3.3.5	STOP	140
3.3.6	PHY Notifications	141
3.3.7	Storing Precoding and Beamforming Tables	143
3.3.8	RESET.....	144
3.4	Slot Messages	145
3.4.1	Slot.indication.....	145
3.4.1a	TIMING.indication.....	145
3.4.2	DL_TTI.request	146
3.4.2b	DL_TTI.response	181
3.4.3	UL_TTI.request	182
3.4.4	UL_DCI.request.....	219
3.4.5	SLOT errors	220
3.4.6	TX_DATA.request	220
3.4.7	RX_DATA.indication	223
3.4.8	CRC.indication.....	224
3.4.9	UCI.indication	227
3.4.10	SRS.indication.....	237
3.4.11	RACH.indication	245
3.4.12	DL Node Sync	247
3.4.13	UL Node Sync	247
3.4.14	Timing Info.....	247
References	248

Tables

Table 2-1	PHY API configuration request messages valid in each PHY state.....	5
Table 2-2	Information returned by the PHY during a PARAM message exchange	6
Table 2-3	SRS-based reporting & layer selection, per direction & usage type ..	55
Table 3-1	General PHY API message format.....	62
Table 3-2	PHY API message header	62
Table 3-3	General PHY API message structure	62
Table 3-4	Adaptation of nFAPI structure.....	63
Table 3-5	PHY API message types	65
Table 3-6	PARAM.request message body	65
Table 3-7	PARAM.response message body	66
Table 3-8	Error codes for PARAM.response	66
Table 3-9	TLV format.....	66
Table 3-10	PARAM.request TLVs	67
Table 3-11	PARAM.response TLVs	67
Table 3-12	Signal and Reference RS.....	69



Table 3-13	Cell and PHY parameters	71
Table 3-14	Carrier parameters.....	73
Table 3-15	PDCCH parameters	73
Table 3-16	PUCCH parameters	74
Table 3-17	PDSCH parameters	79
Table 3-18	PRS parameters.....	79
Table 3-19	PUSCH parameters.....	82
Table 3-20	MsgA-PUSCH parameters.....	84
Table 3-21	PRACH parameters.....	87
Table 3-22	MsgA-PRACH parameters	88
Table 3-23	Measurement parameters	92
Table 3-24	UCI parameters	92
Table 3-25	Capability validity scope	93
Table 3-26	PHY Support.....	98
Table 3-27	PHY / DFE Profile Validity Map	99
Table 3-28	Delay management parameters	99
Table 3-29	Rel-16 mTRP parameters	100
Table 3-30	User Plane and Encapsulation parameters	102
Table 3-31	SRS parameters	105
Table 3-32	Spatial Multiplexing and MIMO Capabilities.....	112
Table 3-33	CONFIG.request message body	112
Table 3-34	CONFIG.response message body	113
Table 3-35	Error codes for CONFIG.response	113
Table 3-36	CONFIG TLVs	114
Table 3-37	PHY configuration	116
Table 3-38	Carrier configuration	118
Table 3-39	Cell configuration	119
Table 3-40	SSB power and PBCH configuration	119
Table 3-41	PRACH configuration	123
Table 3-42	Multi-PRACH configuration table	123
Table 3-43	MsgA-PUSCH configuration table.....	126
Table 3-44	MsgA-PRACH-to-(PRU & DMRS) mapping structure.....	129
Table 3-45	Multi-MsgA-PUSCH configuration table.....	130
Table 3-46	SSB resource configuration table	132
Table 3-47	Multi-SSB resource configuration table	132
Table 3-48	TDD table	133
Table 3-49	Measurement configuration	133
Table 3-50	Beamforming and Precoding.....	134
Table 3-51	UCI configuration	135
Table 3-52	PRB-symbol rate match patterns bitmap (non-CORESET) configuration	135
Table 3-53	CORESET rate match patterns bitmap configuration	136
Table 3-54	LTE-CRS rate match patterns configuration	137
Table 3-55	PUCCH semi-static configuration	138



Table 3-56	PDSCH semi-static configuration	138
Table 3-57	Delay management configuration	139
Table 3-58	Rel-16 mTRP configuration	139
Table 3-59	Error codes for START.request	140
Table 3-60	Error codes for STOP.request	141
Table 3-61	ERROR.indication message body	141
Table 3-62	Extended Status format	142
Table 3-63	PHY API error codes	142
Table 3-64	Extended Status format	143
Table 3-65	CONNECTIVITY.indication message body	143
Table 3-66	Digital beam table (DBT) PDU	144
Table 3-67	Precoding matrix (PM) PDU	144
Table 3-68	Error codes for RESET.request	145
Table 3-69	Slot indication message body	145
Table 3-70	Timing indication message body	146
Table 3-71	DL_TTI.request message body	149
Table 3-72	DL_TTI.request top-level rate match signaling	152
Table 3-73	PDCCH PDU	154
Table 3-74	DL DCI PDU	156
Table 3-75	PDCCH PDU maintenance FAPIv3	156
Table 3-76	PDCCH PDU parameters FAPIv4	157
Table 3-77	PDSCH PDU	163
Table 3-78	PDSCH PTRS parameters	164
Table 3-79	PDSCH maintenance parameters V3	168
Table 3-80	PDSCH PTRS maintenance parameters V3	168
Table 3-81	Rel-16 PDSCH parameters V3	169
Table 3-82	PDSCH parameters FAPIv4	171
Table 3-83	PDSCH parameters FAPIv5	171
Table 3-84	CSI-RS PDU	173
Table 3-85	CSI-RS PDU maintenance parameters from FAPIv3	173
Table 3-86	CSI-RS PDU parameters for FAPIv4	174
Table 3-87	SSB/PBCH PDU	175
Table 3-88	SSB/PBCH PDU maintenance FAPIv3	176
Table 3-89	MAC generated MIB PDU	176
Table 3-90	PHY generated MIB PDU	177
Table 3-91	SSB PDU parameters for FAPIv4	177
Table 3-92	PRS PDU	179
Table 3-93	Tx precoding and beamforming PDU	180
Table 3-94	Tx precoding and beamforming PDU	181
Table 3-95	Tx generalized precoding PDU	181
Table 3-96	DL_TTI.response message body	182
Table 3-97	UL_TTI.request message body	184
Table 3-98	PRACH PDU	186



Table 3-99 PRACH maintenance FAPIv3	187
Table 3-100 Uplink spatial stream assignment FAPIv4	187
Table 3-101 MsgA-PRACH to MsgA-PUSCH map signalling in FAPIv4	189
Table 3-102 PUSCH PDU.....	195
Table 3-103 PUSCH maintenance FAPIv3	197
Table 3-104 Uci information for determining UCI Part1 to Part2 correspondence, added in FAPIv3	197
Table 3-105 Optional puschData information.....	198
Table 3-106 Optional puschUci information	199
Table 3-107 Optional puschPtrs information	200
Table 3-108 Optional dftsOfdm information.....	201
Table 3-109 PUSCH parameters V4.....	202
Table 3-110 MsgA-PUSCH PDU	206
Table 3-111 PUCCH PDU.....	211
Table 3-112 PUCCH basic extension for FAPIv3	211
Table 3-113 PUCCH basic extension for FAPIv4	212
Table 3-114 214	
Table 3-115 FAPIv4 SRS Parameters.....	217
Table 3-116 FAPIv5 SRS Parameters.....	217
Table 3-117 Rx beamforming PDU	218
Table 3-118 SRS-based Rx beamforming PDU	219
Table 3-119 UL_DCI.request message body.....	220
Table 3-120 Error codes for ERROR.indication generated by DL_TTI.request	220
Table 3-121 Error codes for ERROR.indication generated by UL_TTI.request	220
Table 3-122 Error codes for ERROR.indication generated by UL_DCI.request	220
Table 3-123 TX_DATA.request message.....	221
Table 3-124 TBS-TLV structure for TX_DATA.request	222
Table 3-125 Error codes for ERROR.indication.....	222
Table 3-126 RX_DATA.indication message body	224
Table 3-127 TBS-TLV structure for RX_DATA.indication	224
Table 3-128 CRC.indication message body.....	227
Table 3-129 UCI.indication message body.....	228
Table 3-130 UCI PUSCH PDU.....	229
Table 3-131 UCI PUCCH format 0 or 1 PDU	231
Table 3-132 UCI PUCCH format 2, 3 or 4 PDU	232
Table 3-133 SR PDU for format 0 or 1	233
Table 3-134 HARQ PDU for format 0 or 1	233
Table 3-135 SR PDU for format 2, 3 or 4	234
Table 3-136 HARQ PDU for format 2, 3 or 4 or for PUSCH.....	234
Table 3-137 UCI/CSI Inline Transport PDU.....	235
Table 3-138 UCI/CSI TLV transport PDU (for Part2)	237



Table 3-139	SRS.indication message body	238
Table 3-140	SrsReport-TLV structure	238
Table 3-141	FAPIv3 Beamforming report, with PRG-level resolution	239
Table 3-142	Normalized Channel I/Q Matrix	240
Table 3-143	Channel SVD Representation	242
Table 3-144	SRS-based positioning report	243
Table 3-145	RACH.indication message body	247

Figures

Figure 1-1	5G architecture.....	1
Figure 1-2	5G FAPI architecture	2
Figure 1-3	5G RAN protocol model [from 3GPP TS 38.401]	3
Figure 1-4	PHY API interactions.....	3
Figure 2-1	PHY layer state transactions on PHY API configuration messages	5
Figure 2-2	Initialization procedure	6
Figure 2-3	PARAM message exchange.....	7
Figure 2-4	CONFIG message exchange	8
Figure 2-5	START message exchange	9
Figure 2-6	Termination procedure	10
Figure 2-7	Restart procedure	11
Figure 2-8	Reset procedure	11
Figure 2-9	Major reconfiguration procedure	12
Figure 2-10	Minor reconfigure procedure.....	13
Figure 2-11	Query procedure.....	13
Figure 2-12	Notification procedures	14
Figure 2-13	Protocol negotiation	15
Figure 2-14	PHY instantiation procedure	17
Figure 2-15	PHY ID state machine.....	18
Figure 2-16	CONNECTIVITY.Indication call flow.....	19
Figure 2-17	Delay management timing sequence without timestamps, with DL_TTI example	22
Figure 2-18	Periodicity of SLOT.indication messages, illustrated for $\mu = 0, 1, 2, 3$	23
Figure 2-19	SFN/SL synchronization start-up with L2/L3 master	24
Figure 2-20	SFN/SL synchronization with L2/L3 master.....	25
Figure 2-21	SFN/SL synchronization start-up with L1 master.....	26
Figure 2-22	SFN/SL synchronization with L1 master	28
Figure 2-23	DL message order.....	30
Figure 2-24	UL message order	32
Figure 2-25	Time and frequency structure of the synchronization signal block (SSB)	33
Figure 2-26	Possible SS block locations for FR1 (sub6 GHz) deployments	34
Figure 2-27	Possible SS block locations for FR2 deployments.....	34



Figure 2-28	PCH procedure	35
Figure 2-29	DLSCH procedure	36
Figure 2-30	2-layer DLSCH procedure.....	37
Figure 2-31	Multi-slot DL-SCH procedure	38
Figure 2-32	Downlink reference signals.....	40
Figure 2-33	ProfileNR: Transmit power levels	41
Figure 2-34	ProfileSSS: Transmit power levels.....	42
Figure 2-35	RACH procedure	43
Figure 2-36	: 2-Step RACH Procedure	44
Figure 2-37	: 2-Step RACH Mga-A PRACH and MsgA-PUSCH illustration	46
Figure 2-38	: MsgA-PRACH → MsgA-PUSCH mapping illustration	47
Figure 2-39	ULSCH procedure.....	48
Figure 2-40	Multi-slot UL-SCH procedure	49
Figure 2-41	SRS procedure	50
Figure 2-42	CSI procedure	51
Figure 2-43	SR procedure	52
Figure 2-44	Uplink reference signals.....	53
Figure 2-45	Precoding and digital beamforming flow.....	54
Figure 2-46	Overview of digital beam table (DBT)	54
Figure 2-47	Dynamic Precoding flow (Downlink)	56
Figure 2-48	Dynamic Combiner flow (Uplink).....	56
Figure 2-49	MU-MIMO Groups.....	57
Figure 2-50	Example Call flow for MU-MIMO DL Scheduling	58
Figure 2-51	UL_TTI.request error sequence	60
Figure 2-52	DL_TTI.request error sequence	60
Figure 2-53	UL_DCI.request error sequence	61
Figure 2-54	TX_DATA.request error sequence	61



1. Introduction

1.1 5G new radio (NR)

5G NR is standardized by 3GPP (<http://www.3gpp.org>) and designed as an evolution to the current 4G LTE wireless network. Requirements for 5G include higher bandwidth, lower latency, improved reliability and increased density of users/devices; although not necessarily all at the same time. Instead, three service types are defined: enhanced mobile broadband (eMBB), ultra-reliable low latency (URLLC) and massive internet of things (mIoT). This wide range of use cases results in flexibility, which is a key feature of 5G.

Figure 1-1 shows the architecture of a 5G network. It consists of only two elements: the 5G Core (consisting of the access and mobility function (AMF), user plane function (UPF), etc.) and the 5G Node B (gNB). The 5G FAPI resides within the gNB element. The standardized interfaces in 5G network are called NG, Xn and F1. The L1 is not involved in these interfaces.

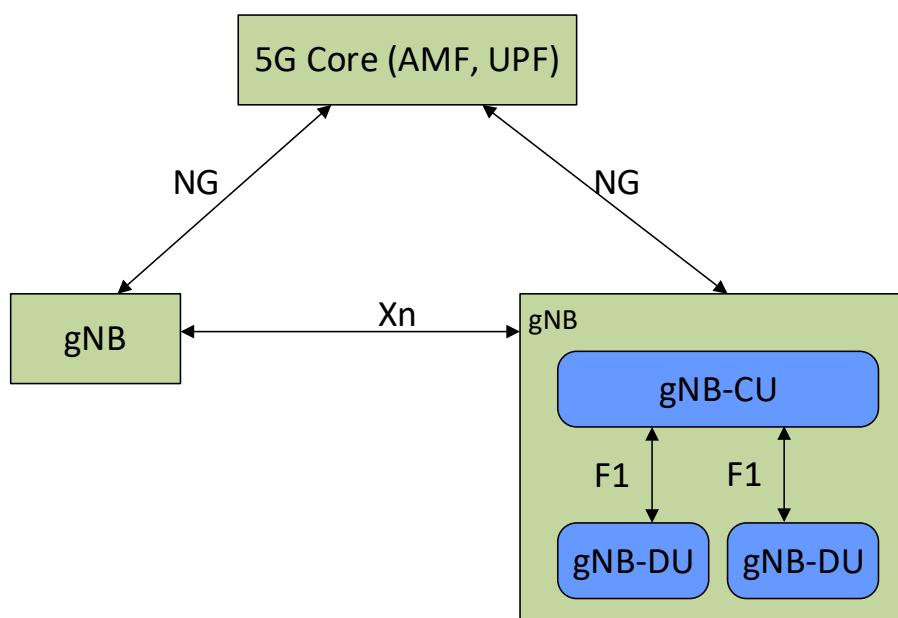


Figure 1-1 5G architecture

For 5G FAPI the MAC and PHY reside in a single element, resulting in FAPI existing as an internal interface within the gNB, as shown in Figure 1-2.

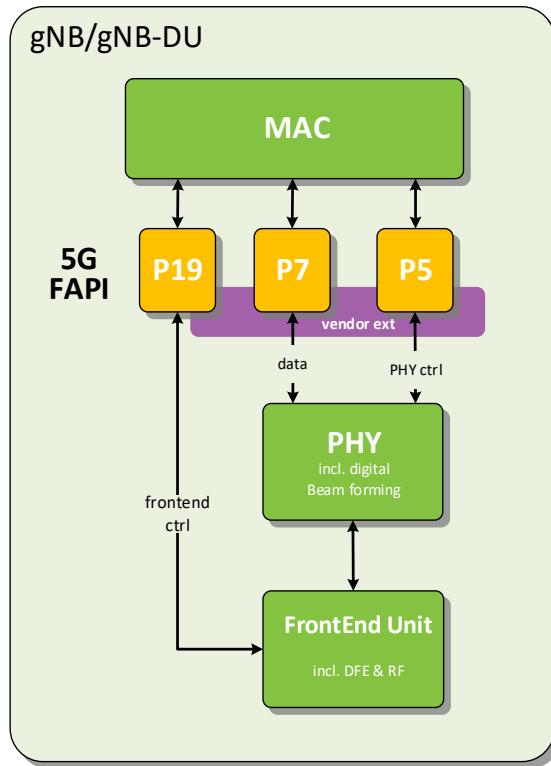


Figure 1-2 5G FAPI architecture

The introduction of 5G nFAPI for split 6 – with MAC and PHY residing in different physical locations, a distributed unit (S-DU) and radio unit (S-RU) – is outside the scope of this document.

1.2 PHY API

The 5G FAPI PHY API, defined in this document, resides within the gNB/gNB-DU component and Figure 1-3 shows the protocol model for the gNB defined in the 5G-RAN architectural standard 3GPP TS 38.401 [1]. It highlights the separation of control- and data-plane information, which is maintained throughout the 5G NR network. Both control- and data-plane information is passed through the PHY API. However, each API message contains either control- or data-plane information, but never both.

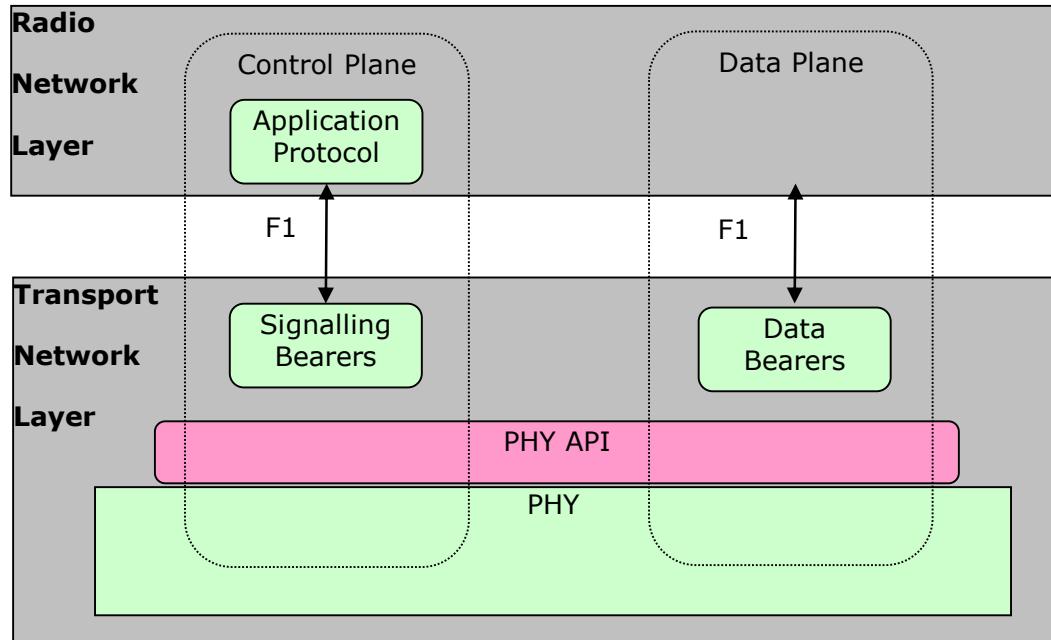


Figure 1-3 5G RAN protocol model [from 3GPP TS 38.401]

Figure 1-4 provides an example of how the different L2/L3 protocol layers will interact with the PHY API. In this example, a PHY control entity is responsible for configuration procedures (P5). The MAC layer is responsible for the exchange of data-plane messages with the PHY (P7). The PHY configuration sent over the P5 interface may be determined using SON techniques, information model parameters sent from an OAM system, or a combination of both methods. If carrier aggregation is supported, then one instance of the PHY API exists for each component carrier, as defined in 3GPP TS 38.104 [7].

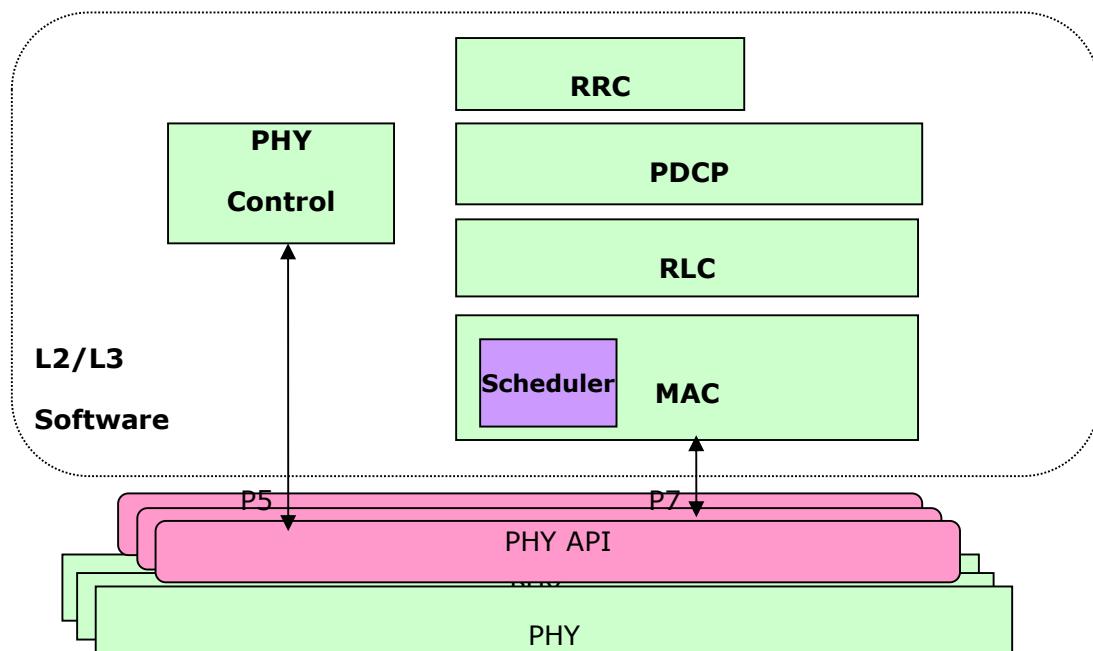


Figure 1-4 PHY API interactions



2. PHY API Procedures

This section gives an overview of the procedures which use the PHY API. These procedures are split into two groups: configuration procedures and slot procedures. Configuration procedures handle the management of the PHY layer and are expected to occur infrequently. Slot procedures determine the structure of each slot and operate with a periodicity based on the subcarrier spacing numerology, namely 125us, 250us, 500us or 1ms periodicity. Unless otherwise noted, these procedures apply per PHY. When defined, PHY ID #0 corresponds to a supervisory entity (e.g. an SoC-wide concept, hosting a non-PHY specific, common context) that cannot be in PHY RUNNING state and cannot terminate P7. In nFAPI, as specified in SCF-225 [10], PNF also terminates messages with PHY ID#0. Digital beamforming (for uplink and downlink channels) and digital precoding (only for PDSCH, in this version of the specification) relies on both configuration and slot procedures:

- Configuration procedure: L2/L3 configures a table of digital beam weights (DBT: digital beam table) applicable to baseband ports, as well as a table of precoder weights (PMT: precoder matrix table) applicable to logical ports.
- Slot procedures: L2/L3 signals per-slot, per-channel beam indices. For PDSCH, L2/L3 also signals precoder matrix indices.

RF beamforming is supported in P19 API, per SCF-223 [9].

2.1 Configuration Procedures

The configuration procedures supported by PHY API (except for API handled at common context) are:

- Initialization
- Termination
- Restart
- Reset
- Query
- Error notification

PHY APIs applicable to the common context are described in 2.1.10

These procedures will move the PHY layer through the IDLE, CONFIGURED and RUNNING states, as shown in Figure 2-1. A list of the PHY API configuration messages which are valid in each state is given in Table 2-1.

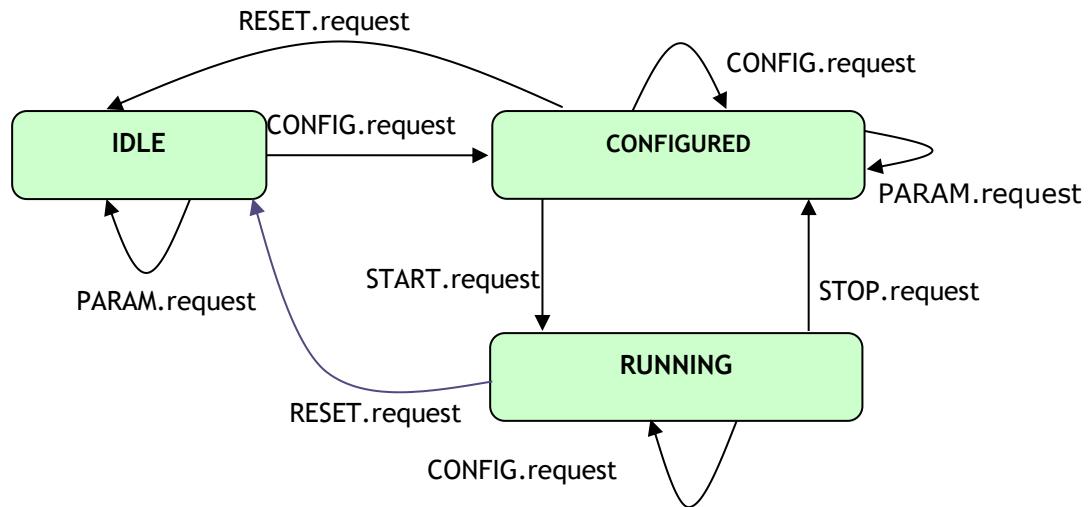


Figure 2-1 PHY layer state transactions on PHY API configuration messages

Idle state	Configured state	Running state
PARAM.request	PARAM.request	CONFIG.request
CONFIG.request	CONFIG.request	STOP.request
	START.request	RESET.request
	RESET.request	

Table 2-1 PHY API configuration request messages valid in each PHY state

When timeouts for P5 request procedures are supported, P5 procedures (from request to response/indication that signals the end of the procedure) are expected to last no more than the time value declared in TLV 0x015F. If a procedure lasts more than that amount, L2/L3 software shall trigger a `RESET.request` for those P5 procedures that change PHY state.

2.1.1 Initialization

The initialization procedure moves the PHY from the IDLE state to the RUNNING state, via the CONFIGURED state. An overview of this procedure is given in Figure 2-2, the different stages are:

- The PARAM message exchange procedure
- The CONFIG message exchange procedure
- The START message exchange procedure

For SFN/SL Synchronization, the initialization procedure is completed when the PHY sends the L2/L3 software a `SLOT.indication` message.

For Delay Management, the initialization procedure is completed when the PHY sends the L2/L3 software a `START.response` message.

The remainder of this section describes the PARAM, CONFIG and START message exchange procedures.

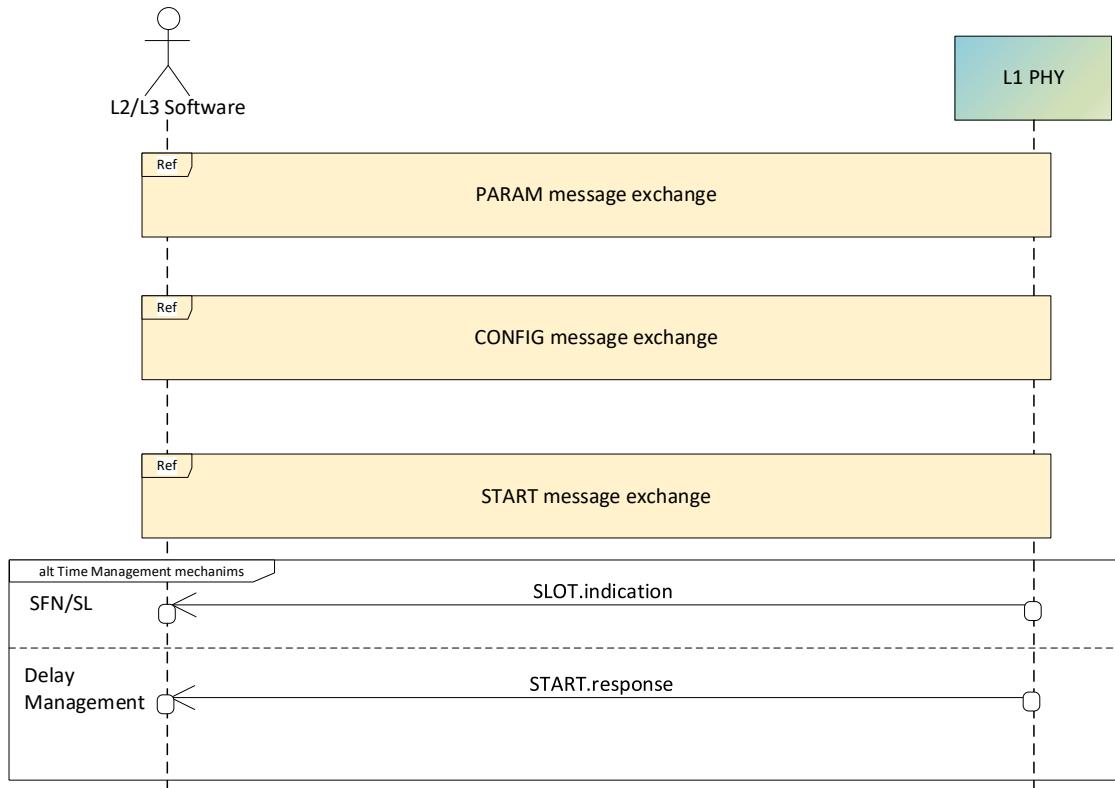


Figure 2-2 Initialization procedure

2.1.1.1 Param message exchange procedure

The PARAM message exchange procedure is shown in

Figure 2-3. Its purpose is to allow the L2/L3 software to collect information about the PHY configuration and current state. The information returned by the PHY depends on its state, and is described in Table 2-2. The PARAM message exchange is optional.

PHY state	Information returned by PHY
IDLE	The PHY indicates which capabilities it supports
CONFIGURED	The PHY returns its current configuration
RUNNING	The PHY returns invalid state

Table 2-2 Information returned by the PHY during a PARAM message exchange

From

Figure 2-3, it can be seen that the PARAM message exchange procedure is initiated by the L2/L3 software sending a `PARAM.request` message to the PHY. It is recommended that the L2/L3 software starts a guard timer to wait for the response from the PHY. If the PHY is operating correctly it will return a `PARAM.response` message. In the IDLE and CONFIGURED states, this message will include the current PHY state and a list of configuration information, as described in Table 2-2. In the RUNNING state, this message will indicate an `INVALID_STATE` error. To determine the PHY capabilities it must be moved to the CONFIGURED state using the termination procedure. If the guard timer expires before the PHY responds, this indicates the PHY is not operating correctly. This must be rectified before further PHY API commands are used; the rectification method is outside the scope of this document.

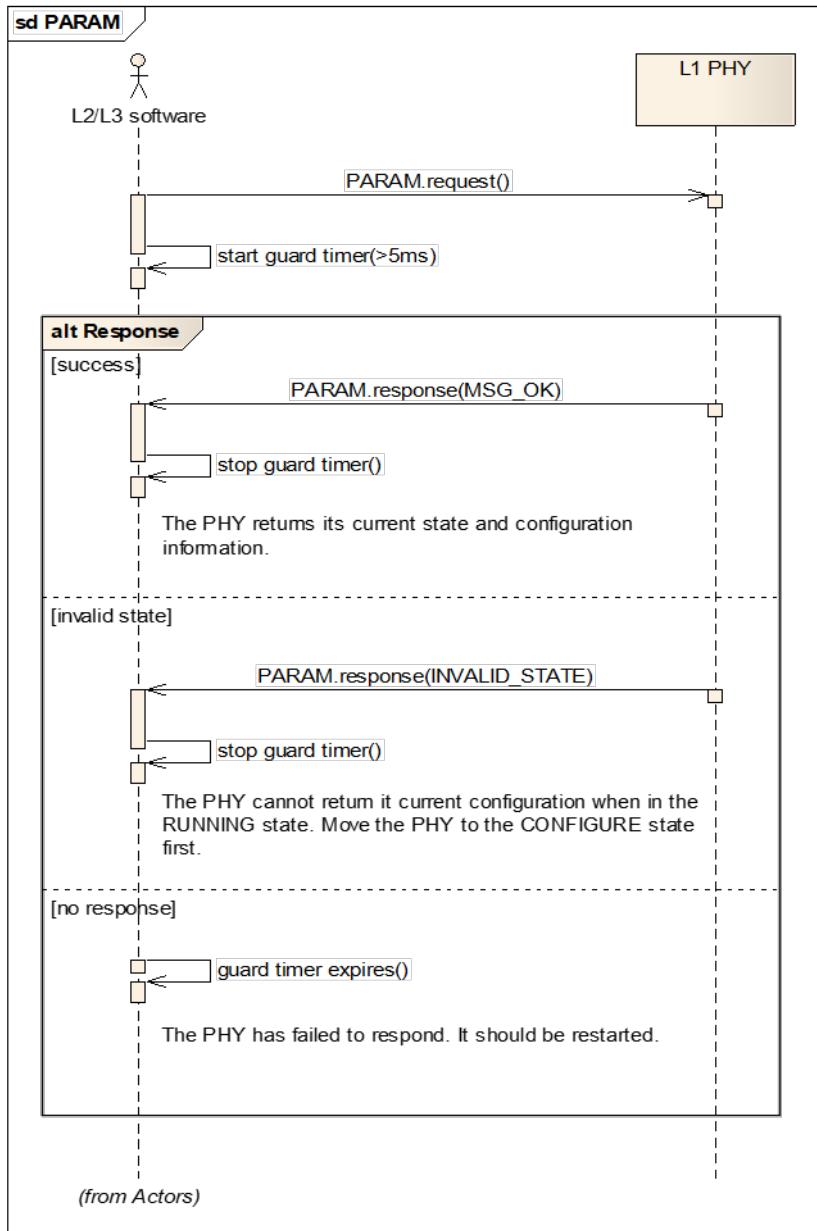


Figure 2-3 PARAM message exchange

2.1.1.2 Config message exchange procedure

The CONFIG message exchange procedure is shown in Figure 2-4. Its purpose is to allow the L2/L3 software to configure the PHY. It can be used when the PHY is in any state. The procedure has slight differences depending on the PHY state. For clarity, each case is described separately.

If the PHY is in the IDLE state, the `CONFIG.request` message, sent by the L2/L3 software, must include all mandatory TLVs. The mandatory TLVs are indicated by the PHY in the `PARAM.response` message. If all mandatory TLVs are included, and set to values supported by the PHY, L1 will return a `CONFIG.response` message, indicating it is successfully configured and has moved to the CONFIGURED state. If the `CONFIG.request` message has missing mandatory TLVs, invalid TLVs, or unsupported TLVs, the PHY will return a `CONFIG.response` message, indicating an incorrect



configuration. In this case, it will remain in the IDLE state and all received TLVs will be ignored.

If the PHY is in the CONFIGURED state, the CONFIG.request message, sent by the L2/L3 software, may include only the TLVs that are required to change the PHY to a new configuration. If the PHY supports these new values, it will return a CONFIG.response message, indicating it has been successfully configured. However, if the CONFIG.request message includes invalid TLVs, or unsupported TLVs, the PHY will return a CONFIG.response message, indicating an incorrect configuration. In this case, all received TLVs will be ignored and the PHY will continue with its previous configuration. In both cases, if the PHY receives a CONFIG.request while in the CONFIGURED state it will remain in the CONFIGURED state.

If the PHY is in the RUNNING state, then a limited subset of CONFIG TLVs may be sent in a CONFIG.request message. The permitted TLVs are indicated by the PHY in PARAM.response. If the CONFIG.request message has invalid TLVs, or TLVs which must not be reconfigured in the RUNNING state, the PHY will return a CONFIG.response message, indicating an incorrect configuration. In this case, it will remain in the RUNNING state and all received TLVs will be ignored.

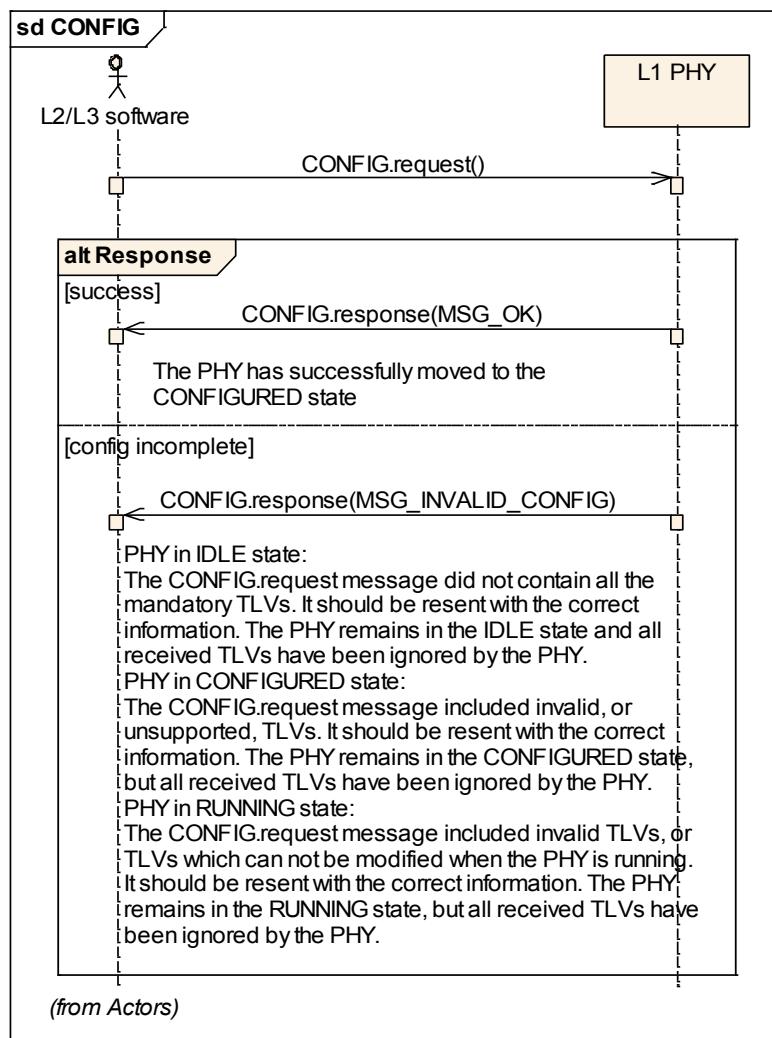


Figure 2-4 CONFIG message exchange



2.1.1.3 Start message exchange procedure

The START message exchange procedure is shown in Figure 2-5. Its purpose is to instruct a configured PHY to start transmitting as a gNB. The L2/L3 software initiates this procedure by sending a START.request message to the PHY.

If the PHY is in the CONFIGURED state, and it supports SFN/SL-based synchronization, it will issue a SLOT indication. After the PHY has sent its first SLOT.indication message, it enters the RUNNING state.

If the PHY is in the CONFIGURED state, and it supports Delay Management, it will issue a START.response, after which it enters the RUNNING state.

If the PHY receives a START.request in either the IDLE or RUNNING state, it will return an ERROR.indication including an INVALID_STATE error.

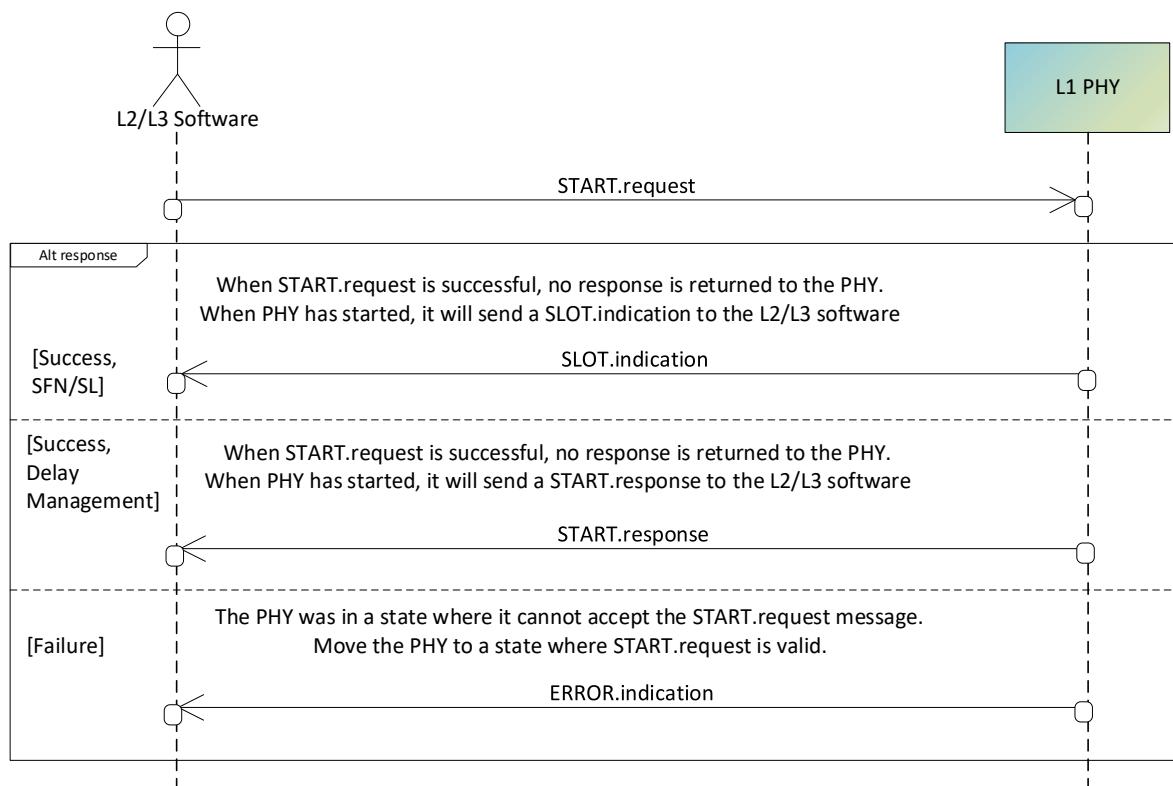


Figure 2-5 START message exchange

2.1.2 Termination

The termination procedure is used to move the PHY from the RUNNING state to the CONFIGURED state. This stops the PHY transmitting as a gNB. The termination procedure is shown in Figure 2-6 and initiated by the L2/L3 software sending a STOP.request message.

If the STOP.request message is received by the PHY while operating in the RUNNING state, it will stop all TX and RX operations and return to the CONFIGURED state. When the PHY has completed its stop procedure, a STOP.indication message is sent to the L2/L3 software.



If the STOP.request message was received by the PHY while in the IDLE or CONFIGURED state, it will return an ERROR.indication message including an INVALID_STATE error. However, in this case the PHY was already stopped.

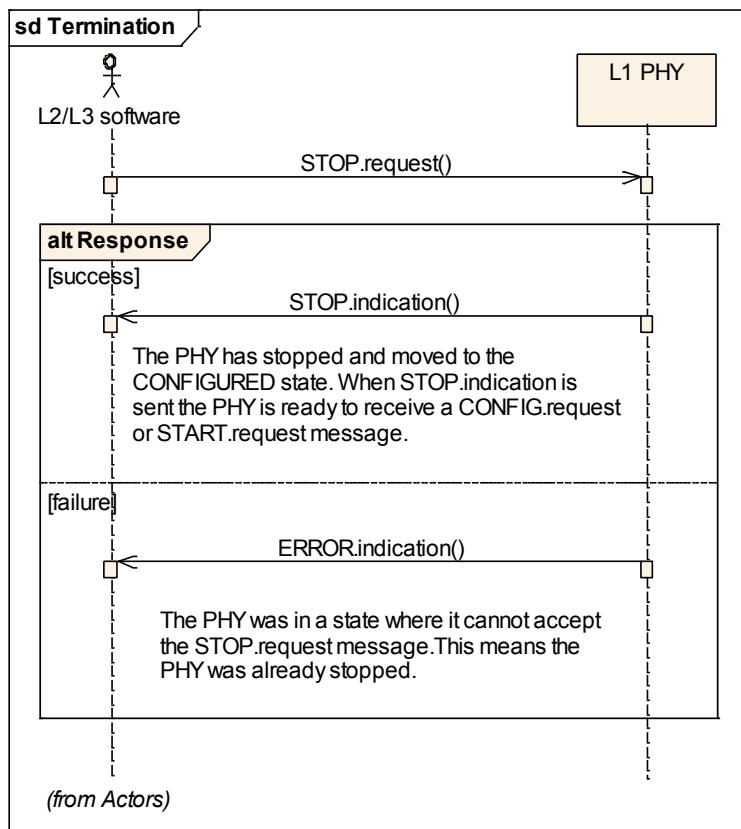


Figure 2-6 Termination procedure

2.1.3 Restart

The restart procedure is shown in Figure 2-7. It can be used by the L2/L3 software when it needs to stop transmitting, but later wants to restart transmission using the same configuration. To complete this procedure, the L2/L3 software can follow the STOP message exchange shown in Figure 2-6. This moves the PHY to the CONFIGURED state. To restart transmission, it should follow the START message exchange, shown in Figure 2-5, moving the PHY back to the RUNNING state.

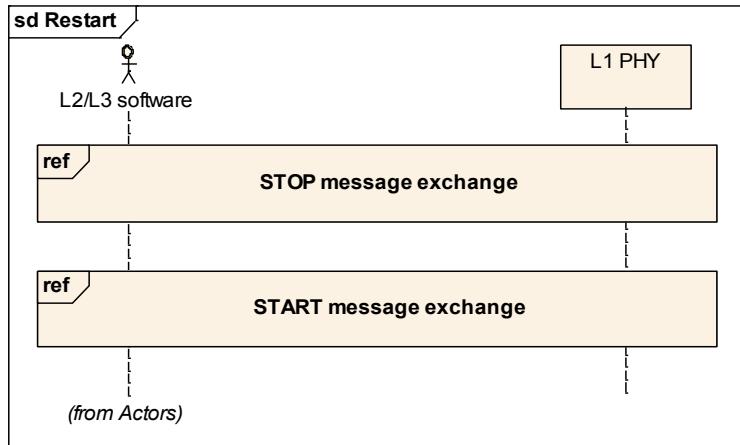


Figure 2-7 Restart procedure

2.1.4 Reset

The reset procedure is shown in Figure 2-8. This procedure is used when the L2/L3 software wants to return the PHY to the IDLE state. Under normal conditions, this can only be achieved by terminating the PHY (as shown in Figure 2-6) and then resetting the PHY.

When L2 detects P5 timeout, it instead invokes `RESET.request` directly.

The RESET procedure is complete when `RESET.indication` is received.

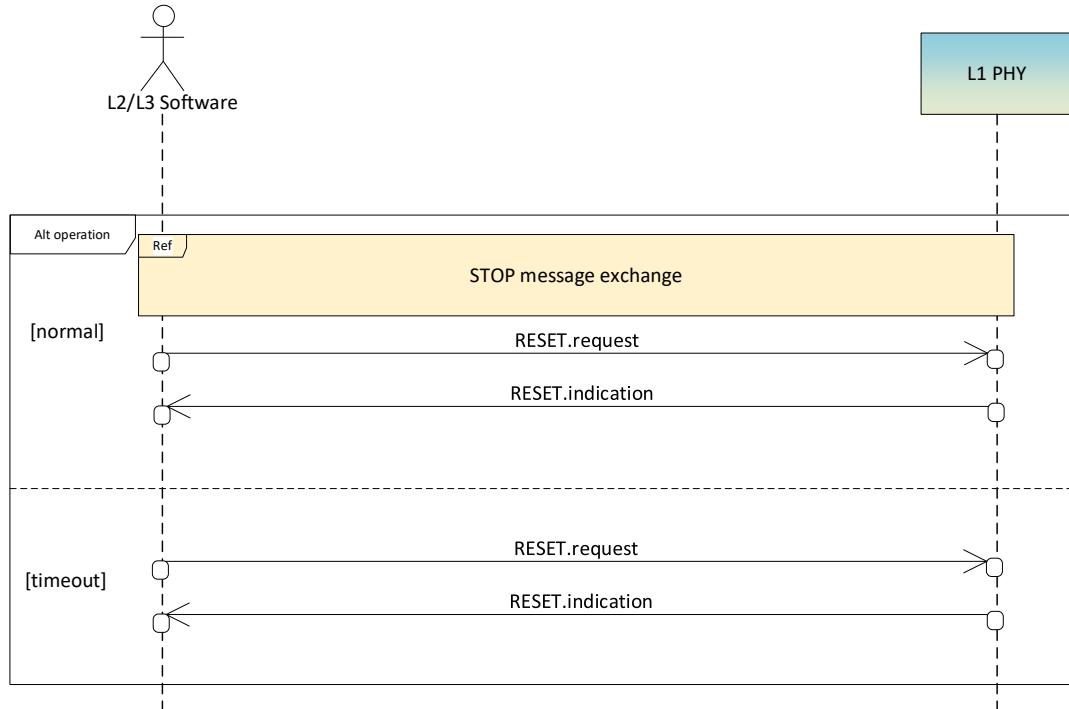


Figure 2-8 Reset procedure

2.1.5 Reconfigure

Two methods of reconfiguration are supported by the PHY:



- a major reconfiguration where the PHY is stopped; and
- a minor reconfiguration where the PHY continues running.

The major reconfigure procedure is shown in Figure 2-9. It is used when the L2/L3 software wants to make significant changes to the configuration of the PHY. The STOP message exchange, shown in Figure 2-6, is followed to halt the PHY and move it to the CONFIGURED state. The CONFIG message exchange, shown in Figure 2-4, is used to reconfigure the PHY. Finally, the START message exchange, shown in Figure 2-5, is followed to start the PHY and return it to the RUNNING state.

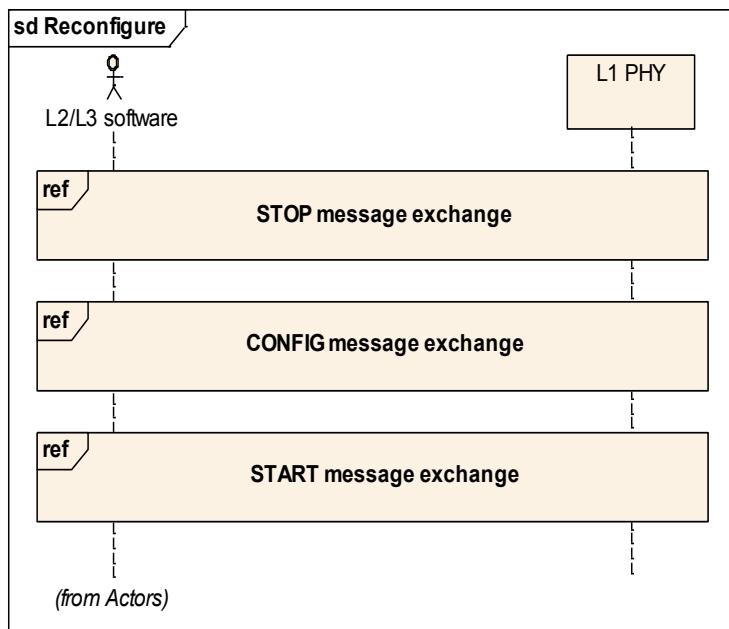


Figure 2-9 Major reconfiguration procedure

The minor reconfiguration procedure is shown in Figure 2-10. It is typically used in conjunction with a RRC system information update.

In the slot where the L2/L3 software requires the configuration change it sends the `CONFIG.request` message to the PHY. Only a limited subset of `CONFIG` TLVs may be sent, these are indicated by the PHY in `PARAM.response`. TLVs included in the `CONFIG.request` message for slot N will be applied at the SFN/SL given in the `CONFIG.request` message. Reconfiguring the PHY while in the RUNNING state has a further restriction, the `CONFIG.request` message must be sent before the `DL_TTI.request` and `UL_TTI.request` message.

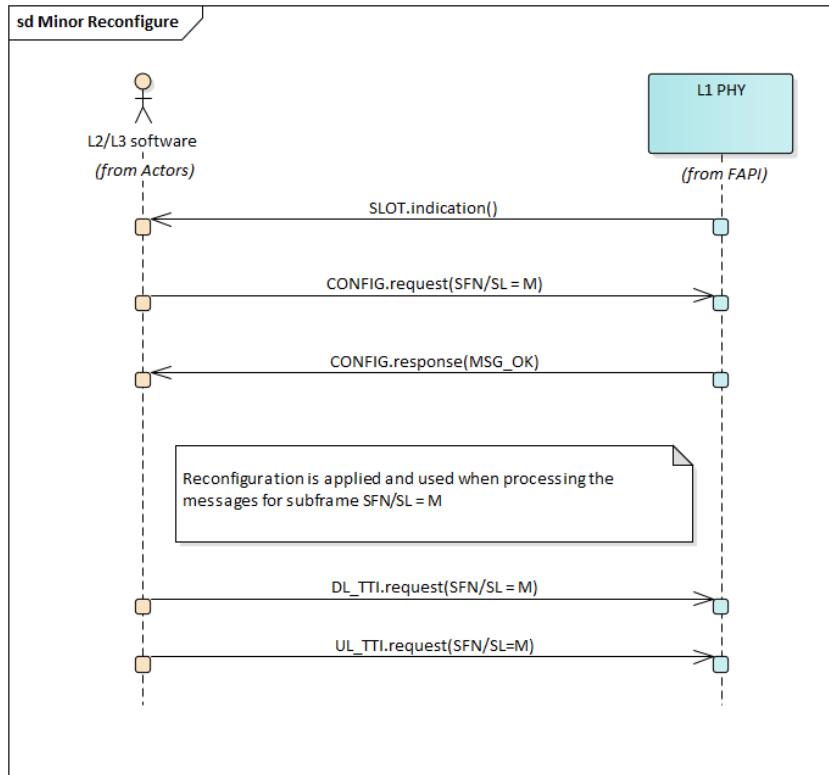


Figure 2-10 Minor reconfigure procedure

2.1.6 Query

The query procedure is shown in Figure 2-11. It is used by the L2/L3 software to determine the configuration and operational status of the PHY. The PARAM message exchange, shown in Figure 2-3, is used. This signalling sequence can be followed when the PHY is stopped, in the IDLE state and, optionally, the CONFIGURED state.

When terminated at Common Context (see section 2.1.10), the PARAM exchange procedure can return capabilities that are not dependent on particular PHY IDs or on PHY states, in particular the following capabilities:

- Any parameter in the Capability Validity table (Table 3-25)
- Any parameter in the PHY Support table (Table 3-26)
- Any parameter in the PHY/DFE Validity Map table (Table 3-27)
- Any other parameter that is valid the PHY ID, regardless of state. When the PHY ID is 0, the parameter applies to all the PHYs.

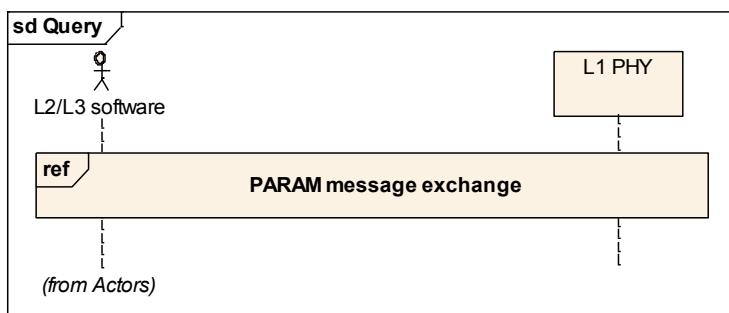


Figure 2-11 Query procedure



2.1.7 Notification

The notification procedure is shown in Figure 2-12. The PHY sends a notification message when it has an event of interest for the L2/L3 software. Currently, there is one notification message called `ERROR.indication`.

The `ERROR.indication` message has already been mentioned in multiple procedures. It is used by the PHY to indicate that the L2/L3 software has sent invalid information to the PHY.

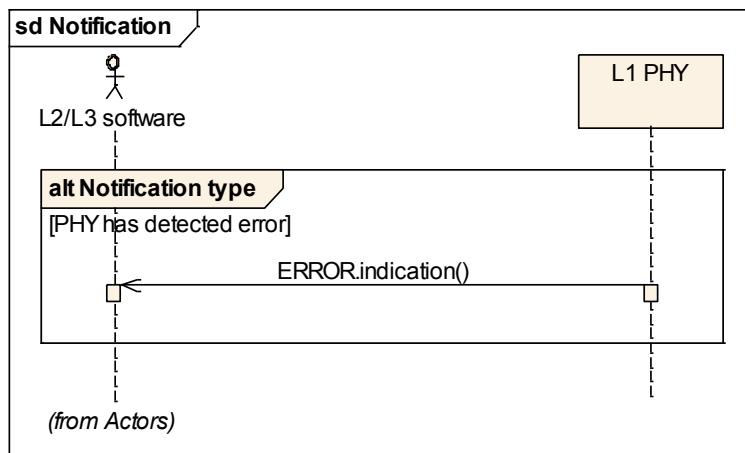


Figure 2-12 Notification procedures

2.1.8 Protocol Negotiation

The protocol negotiation procedure is shown in Figure 2-13. The procedure allows L2/L3 and PHY to ensure that the same FAPI version is used. In particular:

- Protocol negotiation is triggered by L2/L3 sending a `PARAM.request` message (step 1), with a *protocolVersion* indicating a FAPI protocol version supported by L2/L3.
 - Note: typically, this would be the highest FAPI version that L2/L3 supports.
- PHY replies to MAC with a `PARAM.response` message (step 2) that contains:
 - phyFapiProtocolVersion*: the highest FAPI protocol version that PHY supports
 - phyFapiNegotiatedProtocolVersion*: the FAPI version that PHY assumes in subsequent P5/P7 exchanges with L2/L3.

After step 2, both L2/L3 and PHY are assumed to use the same protocol version, as indicated by *phyFapiNegotiatedProtocolVersion*.

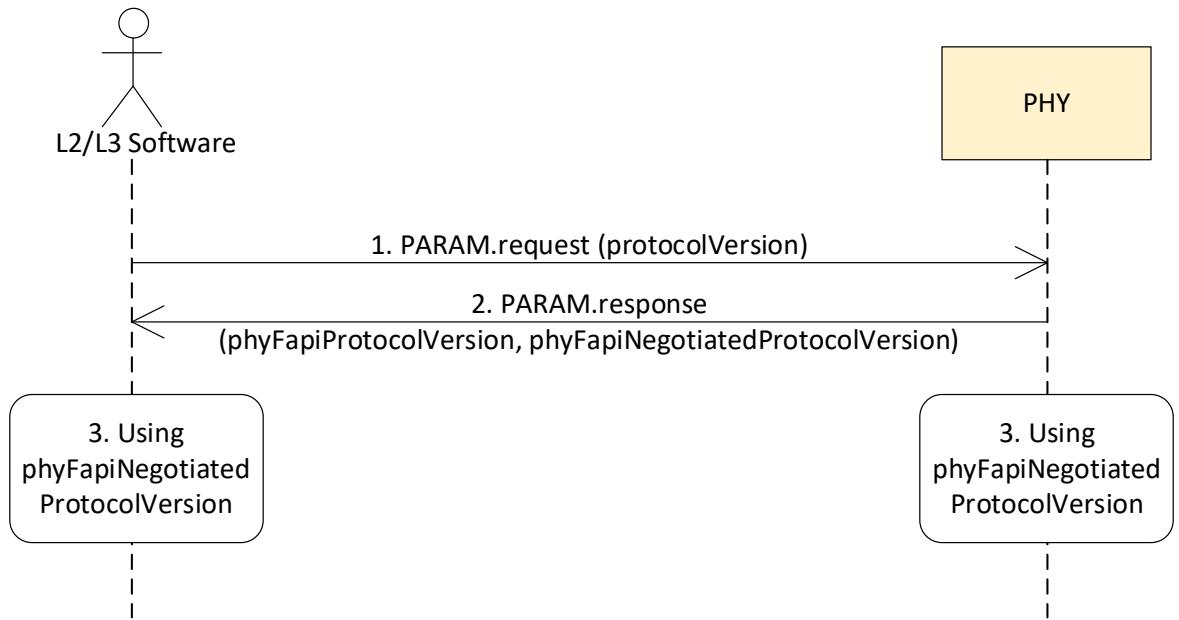


Figure 2-13 Protocol negotiation

Regardless of *phyFapiNegotiatedProtocolVersion*, PHY whose *phyFapiProtocolVersion* is higher than or equal to this FAPI release can always follow the above procedure to negotiate the protocol version.

Protocol negotiation can only happen when all defined PHYs are in IDLE state. Regardless of whether any PHYs are defined, FAPI protocol negotiation can occur with respect to the common context. When supervisory PHY (i.e. PHY ID 0, or PNF in the nFAPI context) is defined, the common context is hosted by supervisory PHY, but common context does not require a supervisory PHY.

The *phyFapiProtocolVersion* is assumed for all exchanges with any PHY IDs sharing the common context.

The PHY handling of any past or future P5 exchanges is only defined if the P5 exchanges are performed according to *phyFapiNegotiatedProtocolVersion*.

2.1.9 PHY Instantiation

This process describes the PHY Profile discovery and selection process, which leads to defining PHYs (and corresponding PHY IDs). It is illustrated in Figure 2-14.

The PHY Instantiation procedure has the following steps:

PHY Profile Discovery: L2/L3 uses the PHY Query Procedure (steps 1 and 2 in the figure) towards Common Context (see section 2.1.10), to discover the supported PHY Profiles and a map indicating which PHY and DFE profiles are pairable.

- Note: Some PHY implementations may pre-define PHY IDs (e.g. > 0), in which case this procedure is not needed.

DFE Profile Discovery: L2/L3 uses the DFE Query Procedure in SCF-223 [9] (steps 3 and 4 in the figure) towards DFE, to discover the supported DFE Profiles



- Note: The order in which DFE and PHY Profile Query procedures are invoked is not subject to this specification.

PHY Profile Selection: L2/L3 selects a PHY profile, via the `CONFIG.request` message (step 5) towards Common Context.

- Note: Common Context does not support PHY (IDLE, CONFIGURED, RUNNING) states, so the state diagram from Figure 2-1 does not apply to PHY ID#0.

PHY Definition: PHY IDs > 0 are defined (but cannot be configured), based on the selected profile (step 6).

- `CONFIG.response` (step 7) announces successful configuration, if PHY Definition succeeds. If unsuccessful, an error code will indicate the failure cause and the PHY Instantiation fails.

DFE Profile Selection: L2/L3 selects a DFE profile, via the `DFE CONFIG.request` message (step 8).

- Note: The order in which DFE and PHY Profile Selections are invoked is not subject to this specification.
- `DFE CONFIG.response` (step 9) announces successful configuration. If unsuccessful, an error code will indicate the failure cause the PHY Instantiation fails.

FEU Configuration and Initialization: L2/L3 executes the Configuration and Initialization procedures towards DFE and RF (step 8)

- The Configuration and Initialization procedures towards FEU components is documented in SCF-225 [10]

PHY Instantiation: When the DFE and RF have both entered RUNNING state, PHY IDs defined through Profile selection become configurable (step 11), after successful DFE and PHY Profile selection.

- If newly defined, these PHY IDs may be considered IDLE.

If PHY IDs existed in any state (IDLE/CONFIGURED/RUNNING) prior to any of the PHY Profile Selection and DFE Profile Selection steps, they remain in the same states, as long as none of their capabilities or configurations are not impacted by the PHY or DFE Profile Selections, otherwise `ERROR.indication` will be triggered by these PHY IDs.

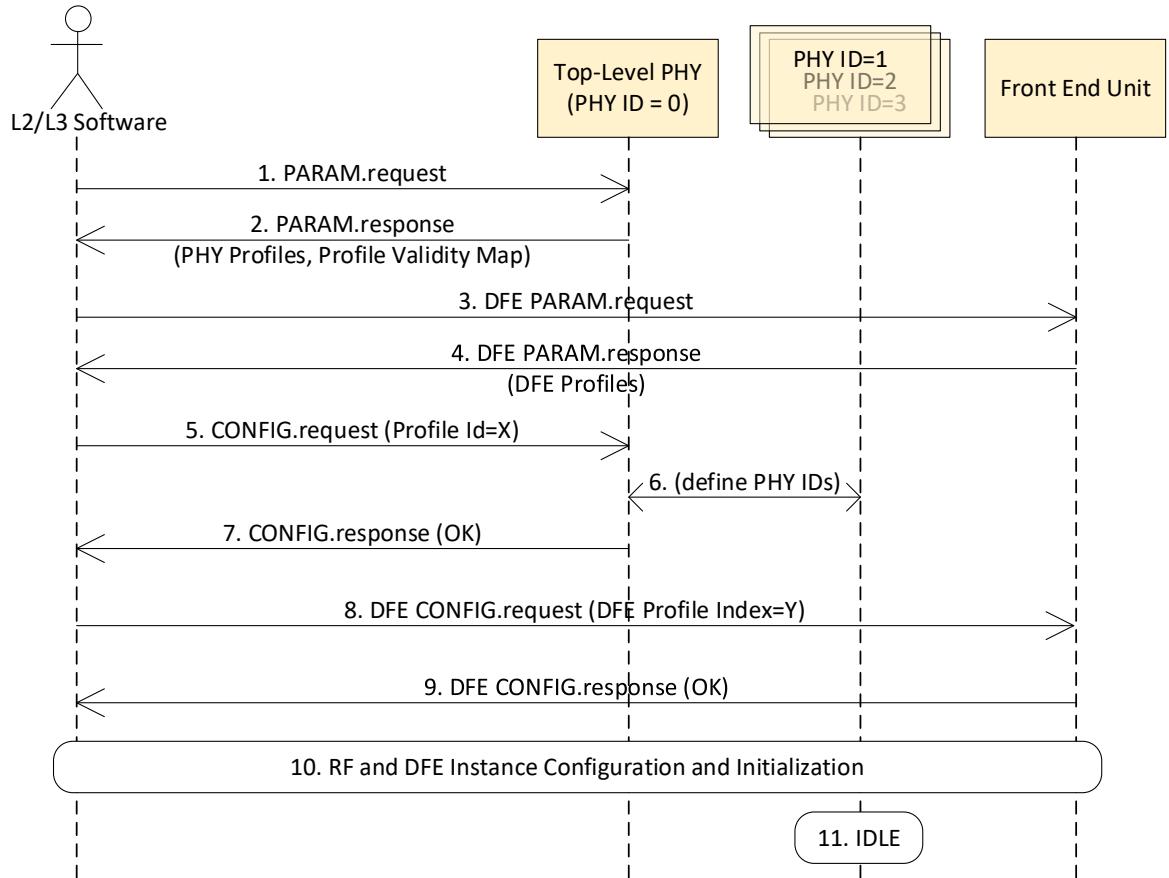


Figure 2-14 PHY instantiation procedure

In general, PHY instantiation is a procedure that controls the definition and for each PHY ID and instantiation of each PHY, as illustrated in Figure 2-15. In particular, a particular PHY ID x :

- PHY ID x is defined when L2/L3 selects a PHY profile that includes x
- A defined PHY ID x is associated with a PHY state machine for the PHY associated with PHY ID x .
 - o For a newly defined PHY ID, the corresponding PHY state machine is in IDLE state.
- A PHY (with a defined PHY ID) can be assumed CONFIGURABLE when it is associated FEU and when all the FEUs (DFE and RF instances) that it associated with are initialized
- A PHY ID stops being defined when L2/L3 selects a PHY profile that excludes x .
 - o this can only happen if PHY Idx is in CONFIGURED or IDLE state, otherwise the profile selection is rejected.

In addition, L1 rejects any PHY profile if such a change results in any capability, configuration or FEU mapping for a PHY ID x that is in PHY RUNNING state.

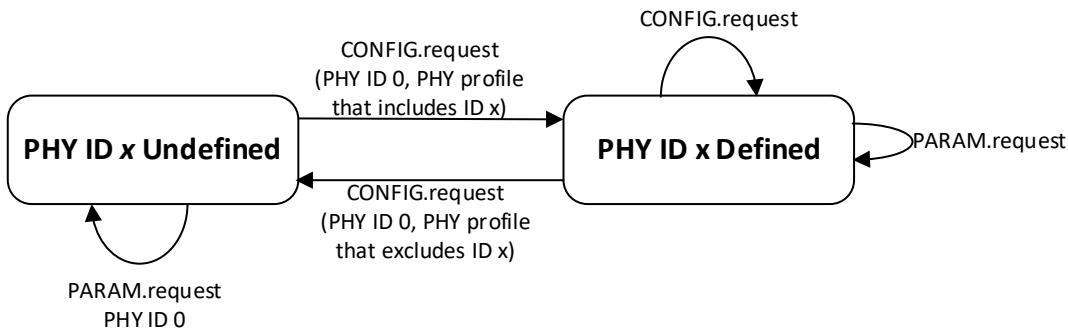


Figure 2-15 PHY ID state machine

2.1.10 Common Context

As discussed in 2.1, some P5 APIs can terminate in the common context, rather than in a specific PHY. When a supervisory PHY concept (e.g. PHY ID 0 or PNF in SCF-225 [10]) is supported, the common context is supported in this supervisory PHY.Common Context:

- only terminates P5, but not P7
- cannot be in PHY RUNNING state
 - i.e., cannot terminate the P5 START or STOP procedures
- can be queried (see section 2.1.6)
- can be configured

For configuration, common context can terminate or initiate the following procedures:

- CONFIG exchange, for the PHY Config TLVs (see Table 3-37)
- If PNF is supported, any PNF_xxx procedures from SCF-225 [10]

2.1.11 PHY/FEU Interface

In some cases (e.g. split 7.2x, as documented in section 4.5.3.2 of [26]), the interface between PHY and FEU may be subject to disconnection or synchronization events. These events may be observable at L2/L3 software, via PHY (P5 interface), or via FEU (P19-C interface in SCF-223 [9]).

An L2 that can observe such events, shall take the following actions:

- When a PHY's associated FEU is determined to have become disconnected:
 - L2/L3 software shall transition the corresponding PHY and FEU to Idle, via the mechanisms in section 2.1.
 - L2/L3 software shall re-query the FEU's capability (as well as any associated PHY's capability), after the FEU is determined to have been reconnected.
- When a PHY or its associated FEU is determined to have become desynchronized:
 - L2/L3 software shall stop any RUNNING PHY, via the mechanisms in section 2.1
 - When the PHY and its associated FEU are both determined to be synchronized, L2 may transition the PHY to RUNNING state, via the mechanisms in section 2.1



When PHY/FEU interface disconnection or synchronization events are observable by PHY, respectively FEU, the observations shall be made known to L2/L3 software via the P5 CONNECTIVITY.Indication (section 3.3.6.3) and P19-C Connectivity Indication (in SCF-223 [9]) APIs, as illustrated in Figure 2-16.

Figure 2-16 illustrates a disconnection/reconnection scenario and a synchronization loss/recovery scenario. In the case of reconnection, the reconnection observation can be coupled with a synchronization loss, or with a synchronization gain.

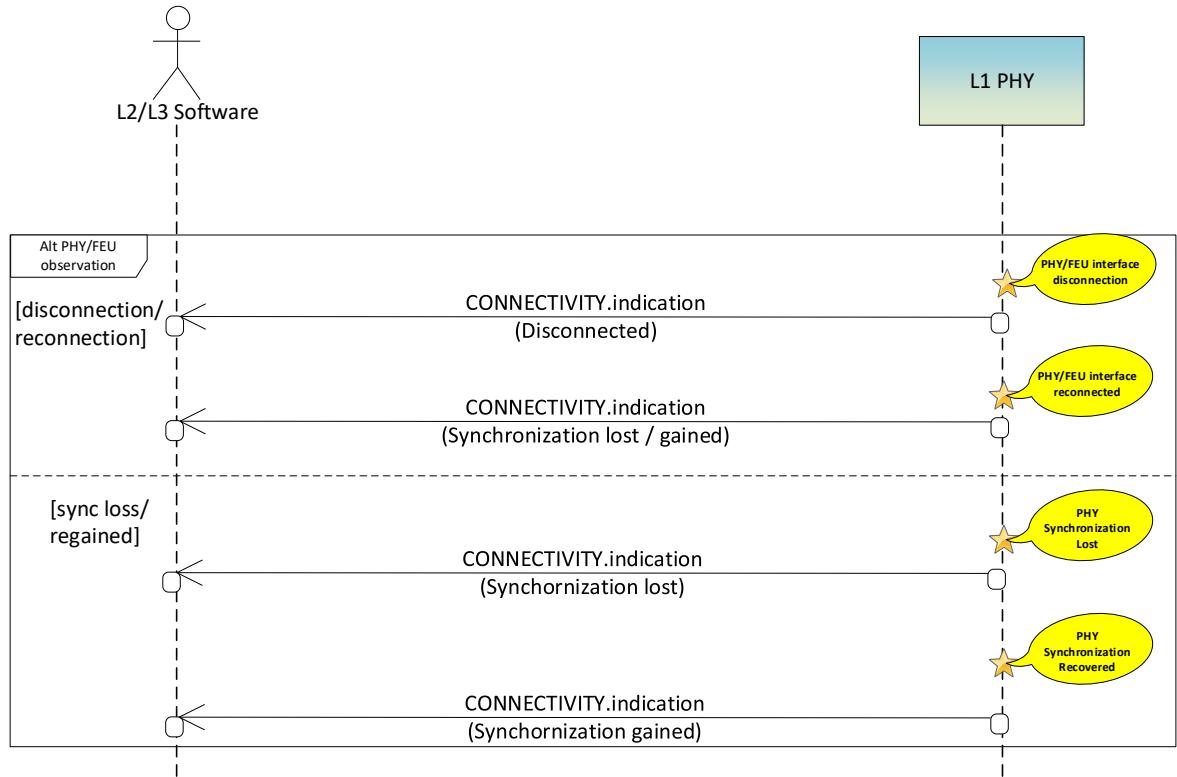


Figure 2-16 CONNECTIVITY.Indication call flow

2.2 Slot Procedures

The slot procedures have two purposes. Firstly, they are used to control the DL and UL frame structures. Secondly, they are used to transfer the slot data between the L2/L3 software and PHY. The slot procedures supported by the PHY API are:

- Transmission of a 62.5us, 125us, 250us, 500us or 1ms SLOT message
- Synchronization of SFN/Slot between the L2/L3 software and PHY
- Transmission of the BCH transport channel
- Transmission of the PCH transport channel
- Transmission of the DL-SCH transport channel
- Transmission of the downlink control information (DCI)
- Transmission of the CSI reference signal
- Reception of the RACH transport channel
- Reception of the UL-SCH transport channel
- Reception of the uplink control information (UCI)
- Reception of the sounding reference signal



2.2.0 Delay Management

This section describes two delay management mechanisms, based on receive window maintained at L1, per message type.

Note that receive windows are anchored to slots. The slot being anchored to is specific to each of the PDU(s) carried in the message.

- e.g. if a `DL_TTI.request` carries a 15 kHz PDSCH PDU and a 30 kHz SSB PDU, the 15 PDSCH PDU reception shall be handled according to the Rx window for the 15 kHz slot and the SSB PDU reception shall be handled according to the Rx window for the 30 kHz slot, relevant to the `DL_TTI.request` message.

Special slot anchoring is assumed, as follows:

- PRACH PDUs for Long preambles shall be anchored to the 15 kHz PRACH numerology.
- If supported (see capability TLV *DL_TTI RxWindow for SSB*) and configured, SSB PDUs shall target the `DL_TTI.request` Rx Window for the highest numerology for PDSCH, if lower (in kHz) than the SSB numerology.
- If supported (see capability TLV *DL_TTI RxWindow for PRACH*) and configured, PRACH PDUs shall target the `UL_TTI.request` Rx Window for highest numerology for PUSCH, if lower (in kHz) than the PRACH numerology.

2.2.0.1 Delay Management with Timestamps

For PHYs supporting both Delay Management and the nFAPI message format, the supported slot procedures are as documented in section 2.1.3 of SCF-225 [10].

2.2.0.2 Delay Management without Timestamps

Where L2 and L1 cannot rely on time stamps, it is still possible to support Delay Management, as described in this section, to ensure that P7 slot procedures occur in a timely fashion.

Similarly to Delay Management procedure documented in 2.2.0.1:

- a. a PHY instance expects P7 slot-based messages from VNF to reach the instance in a Receive Time Window interval, that PHY maintains to buffer and process time-critical P7 messages (`DL_TTI.request`, `UL_TTI.request`, `UL_DCI.request`, `TX_DATA.request`), to apply at their targeted slots.
- b. a Receive Timing Window is characterized by the following parameters, illustrated in Figure 2-11 of SCF-225 [10]:
 - A. a size (*Timing Window*) and
 - B. offset prior to the targeted slot ($<\text{msg}>$ *Timing offset*, where $<\text{msg}>$ is one of the above-listed time-critical messages)
- c. relevant P7 messages for Slot S received by PHY Receive Window mechanism:
 - A. inside the Timing Window interval: are processed for transmission, reception or configuration at Slot S;
 - B. outside the Timing Window interval: are marked by PHY as “too early” or “too late”, as appropriate, and used to construct a *timing report* to L2.

Unlike the Delay Management procedure documented in 2.2.0.1:

- d. the headers of P7 messages are not assumed to signal any time stamps.



- e. the L1 → L2 *timing report* from c.B above is TIMING.indication, as described in section 03.4.1A, which requires no time stamps in P7 messages.
 - A. TIMING.indication may be event-driven (by too-early, too-late events), or periodic, per configuration.
- f. an initial TIMING.indication message is triggered by transition of PHY into RUNNING state.
 - A. DL/UL Node Sync messages are not needed for the timestamp-free Delay Management mechanism in this section
 - B. slot-level timing awareness between L2 and L1 may make use of TIMING.indication
 - C. L2 can adjust its transmission window based on TIMING.indication

Delay Management without timestamps is illustrated in Figure 2-17, for `DL_TTI.request` messages

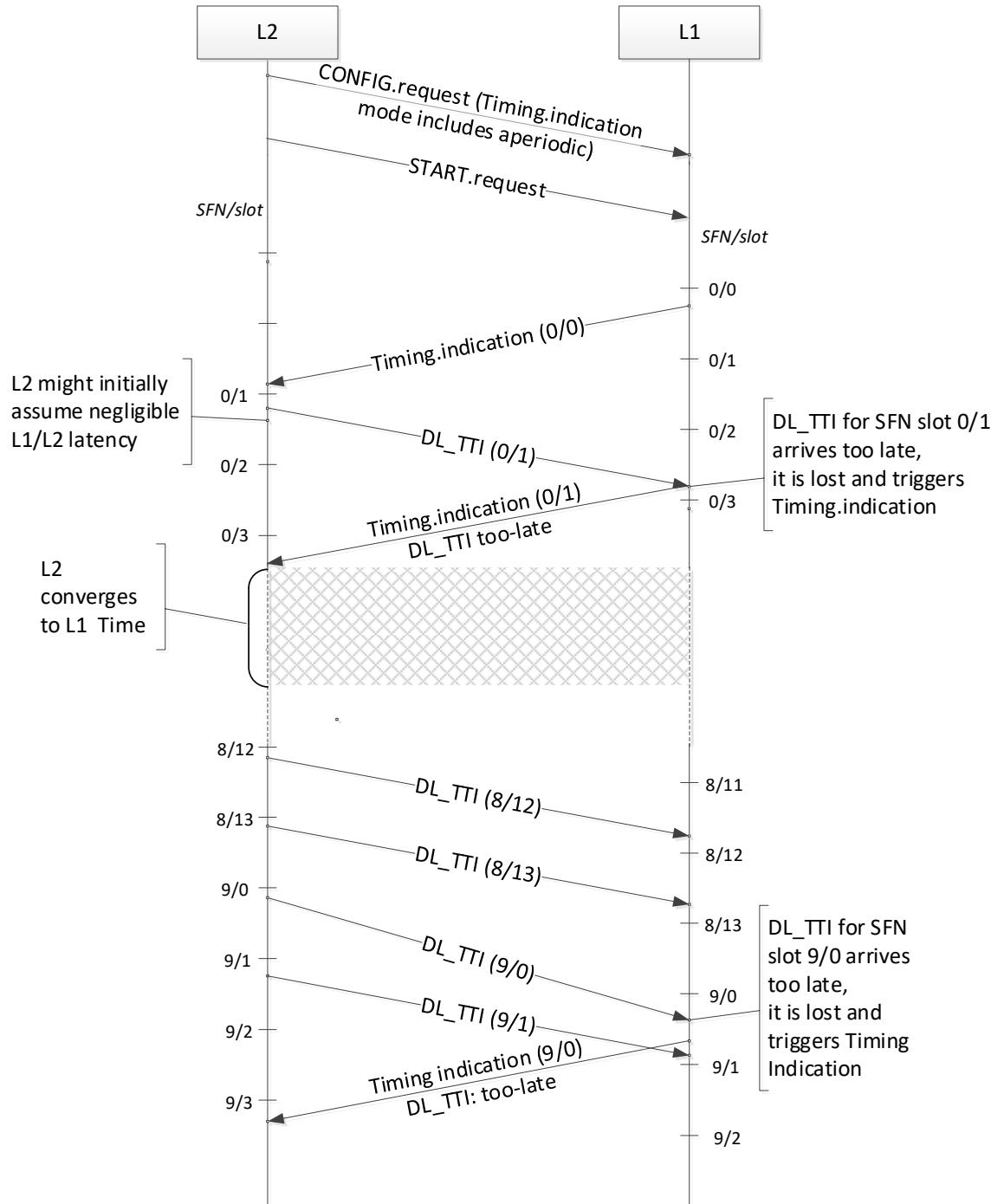


Figure 2-17 Delay management timing sequence without timestamps, with DL_TTI example

2.2.1 SLOT signal

This section only applies to PHYs supporting SFN/SL synchronization, as documented in section 2.2.2.



A `SLOT.indication` message is sent from the PHY, to the L2/L3 software, indicating the start of a slot.

The periodicity of the `SLOT.indication` message is dependent on the numerology and illustrated in Figure 2-18, where the number of `SLOT.indication` messages per subframe is highlighted. The case where $\mu_{\max} = 4$ is not illustrated on that figure.

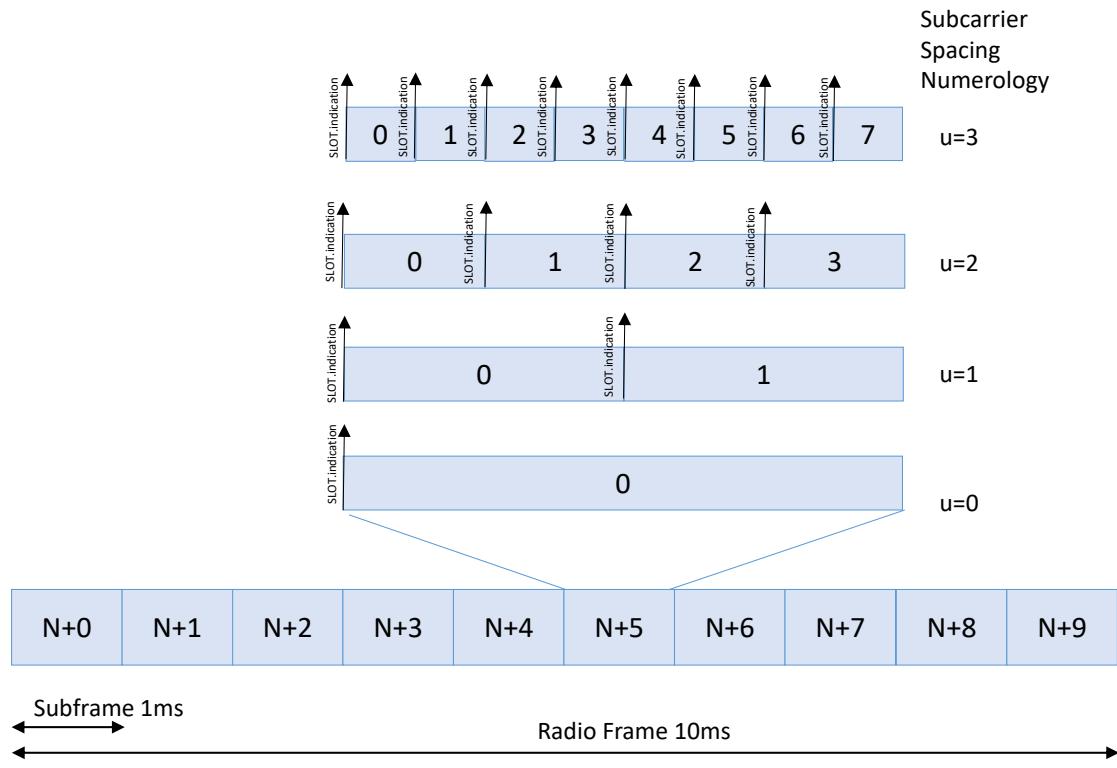


Figure 2-18 Periodicity of `SLOT.indication` messages, illustrated for $\mu = 0, 1, 2, 3$

2.2.2 SFN/SL Synchronization

For PHYs supporting Delay Management, see section 2.2.0.

The SFN/SL synchronization procedure is used to maintain a consistent SFN/SL value between the L2/L3 software and PHY. Maintaining this synchronization is important since different subframes and slots have different structures.

Two options are provided by the PHY API; the first option configures the PHY to use the SFN/SL value provided by the L2/L3 software. The second option configures the PHY to initialize the SFN/SL and ensure the L2/L3 software remains synchronous. The synchronization option is selected at compile time. For each option two procedures are described, the initial start-up synchronization and the maintenance of the synchronization.

2.2.2.1 L2/L3 software is master

The SFN/SL synchronization start-up procedure, where the L2/L3 software is master, is given in Figure 2-19. The start-up procedure followed is:

- After successful configuration the L2/L3 software sends a `START.request` message to move the PHY to the `RUNNING` state



- When the L2/L3 software is configured as master the initial PHY SFN/SL = M, where M could be any value. In the SLOT.indication message, SFN/SL = M
- The L2/L3 software sends a DL_TTI.request message to the PHY containing the correct SFN/SL = N
- The PHY uses the SFN/SL received from the L2/L3 software. It changes its internal SFN/SL to match the value provided by the L2/L3 software

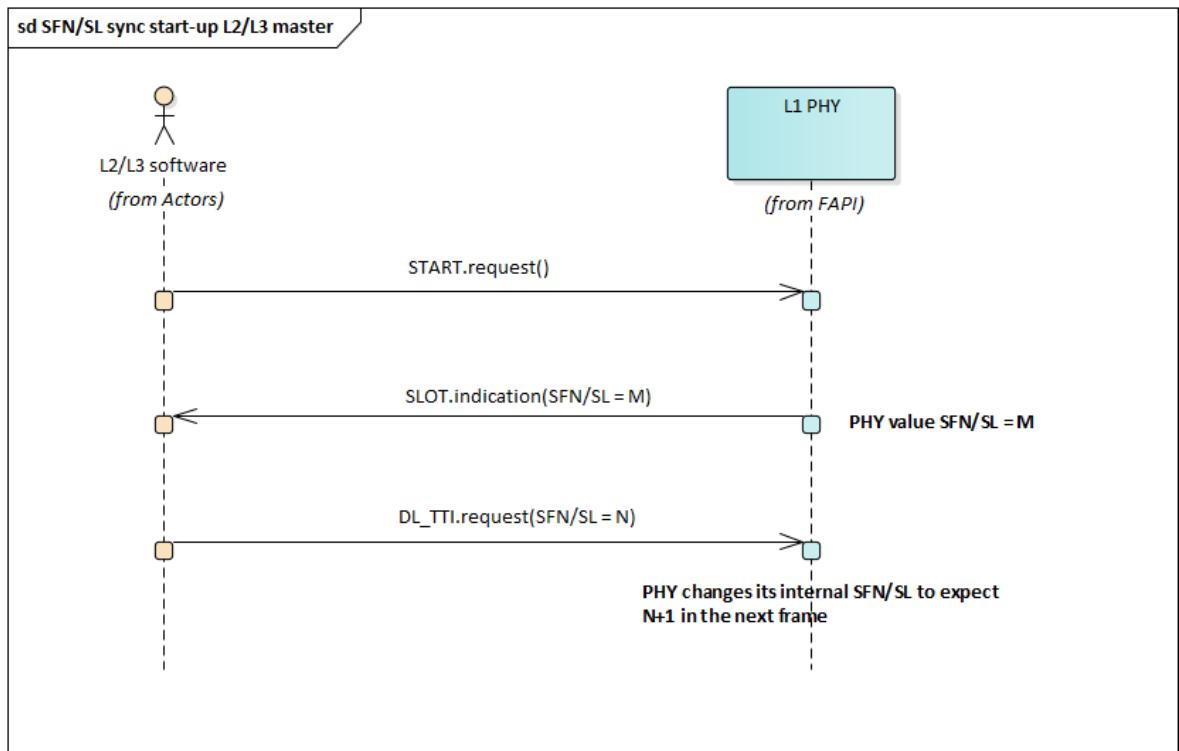


Figure 2-19 SFN/SL synchronization start-up with L2/L3 master

The SFN/SL synchronization maintenance procedure is shown in Figure 2-20. In this example, the L1 PHY is expecting the next DL_TTI.request to contain information regarding frame M. The procedure followed is:

- The PHY sends the SLOT.indication message with SFN/SL = M.
- The L2/L3 software sends a DL_TTI.request or UL_TTI.request message to the PHY containing SFN/SL = N
- If SFN/SL M = N
 - The PHY received the SFN/SL it was expecting. No SFN/SL synchronization is required
- If SFN/SL M ≠ N
 - The PHY received a different SFN/SL from the expected value. SFN/SL synchronization is required
 - The PHY uses the SFN/SL received from the L2/L3 software. It changes its internal SFN/SL to match the value provided by the L2/L3 software
 - The PHY returns an ERROR.indication message indicating the mismatch



This SFN/SL synchronization procedure assumes the L2/L3 software is always correct. However, it's possible the SFN/SL synchronization was unintended, and due to a L2/L3 software issue. The generation of an `ERROR.indication` message, with expected and received SFN/SL values, should allow the L2/L3 software to perform a correction with a further SFN/SL synchronization.

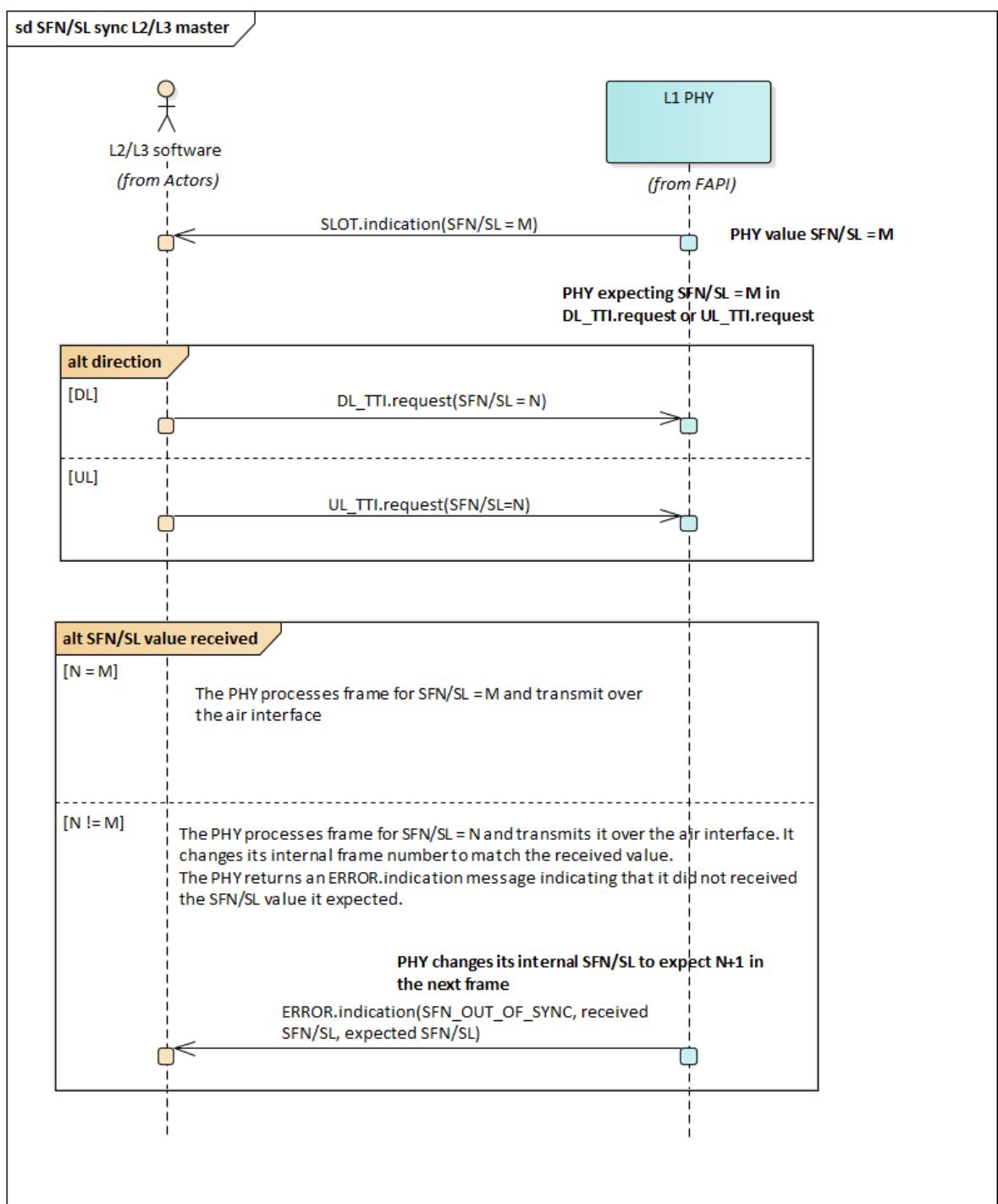


Figure 2-20 SFN/SL synchronization with L2/L3 master



2.2.2.2 L1 PHY is master

The SFN/SL synchronization start-up procedure, where the L1 software is master, is given in Figure 2-21. The start-up procedure followed is:

- After successful configuration the L2/L3 software sends a `START.request` message to move the PHY to the RUNNING state
- If the L1 software is configured as master, the initial PHY SFN/SL = M. The value of M is not deterministic, and could have been set by an external mechanism, such as GPS. The PHY sends a `SLOT.indication` message to the L2/L3 software, with SFN/SL = M. The L2/L3 software uses the SFN/SL received from the PHY. It changes its internal SFN/SL to match the value provided by the PHY
- The L2/L3 software sends a `DL_TTI.request` or `UL_TTI.request` message to the PHY containing SFN/SL = M

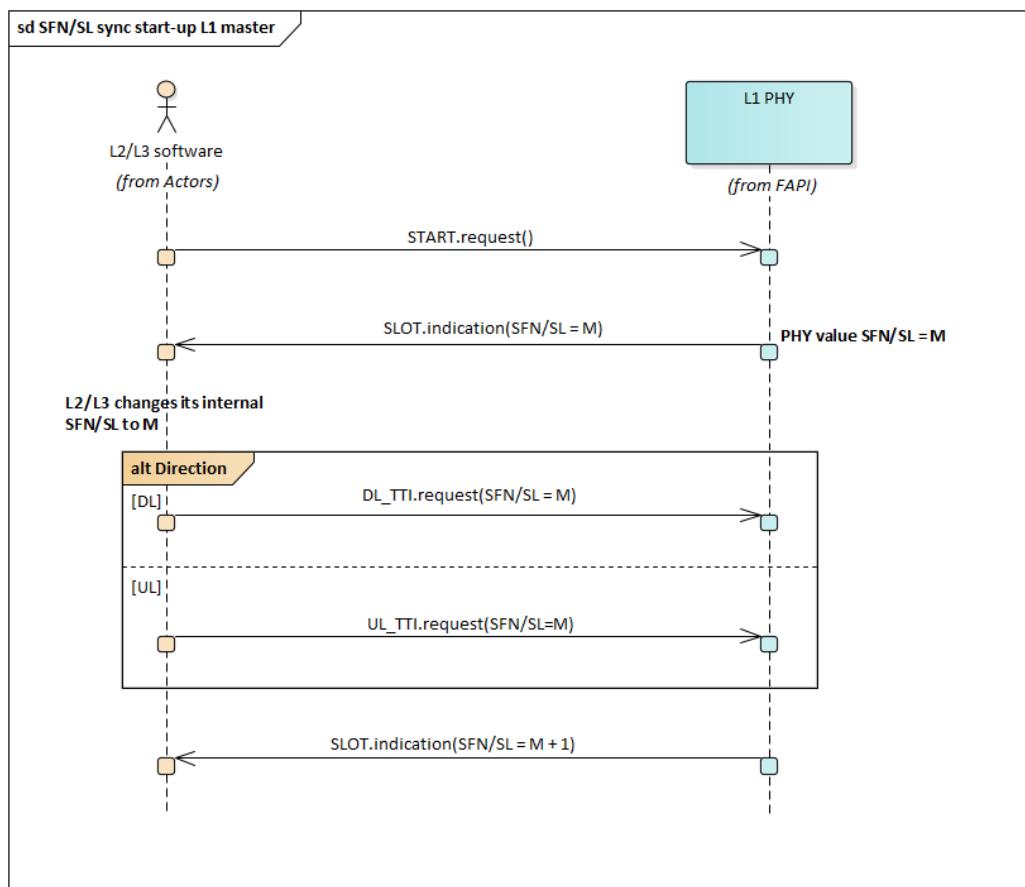


Figure 2-21 SFN/SL synchronization start-up with L1 master

The SFN/SL synchronization maintenance procedure is shown in Figure 2-22. In this example, the L1 PHY is expecting the next `DL_TTI.request` or `UL_TTI.request` to contain information regarding frame M. The procedure followed is:

- The PHY sends a `SLOT.indication` message to the L2/L3 software, with SFN/SL = M
- The L2/L3 software sends a `DL_TTI.request` or `UL_TTI.request` message to the PHY containing SFN/SL = N
- If SFN/SL M = N



- The PHY received the SFN/SL it was expecting. No SFN/SL synchronization is required
- If SFN/SL M ≠ N
 - The PHY received a different SFN/SL from the expected value. SFN/SL synchronization is required
 - The PHY discards the received `DL_TTI.request` or `UL_TTI.request` message
 - The PHY returns an `ERROR.indication` message indicating the mismatch

This SFN/SL synchronization procedure will continue to discard `DL_TTI.request` request or `UL_TTI.request` messages and emit `ERROR.indication` messages until the L2/L3 software corrects its SFN/SL value.

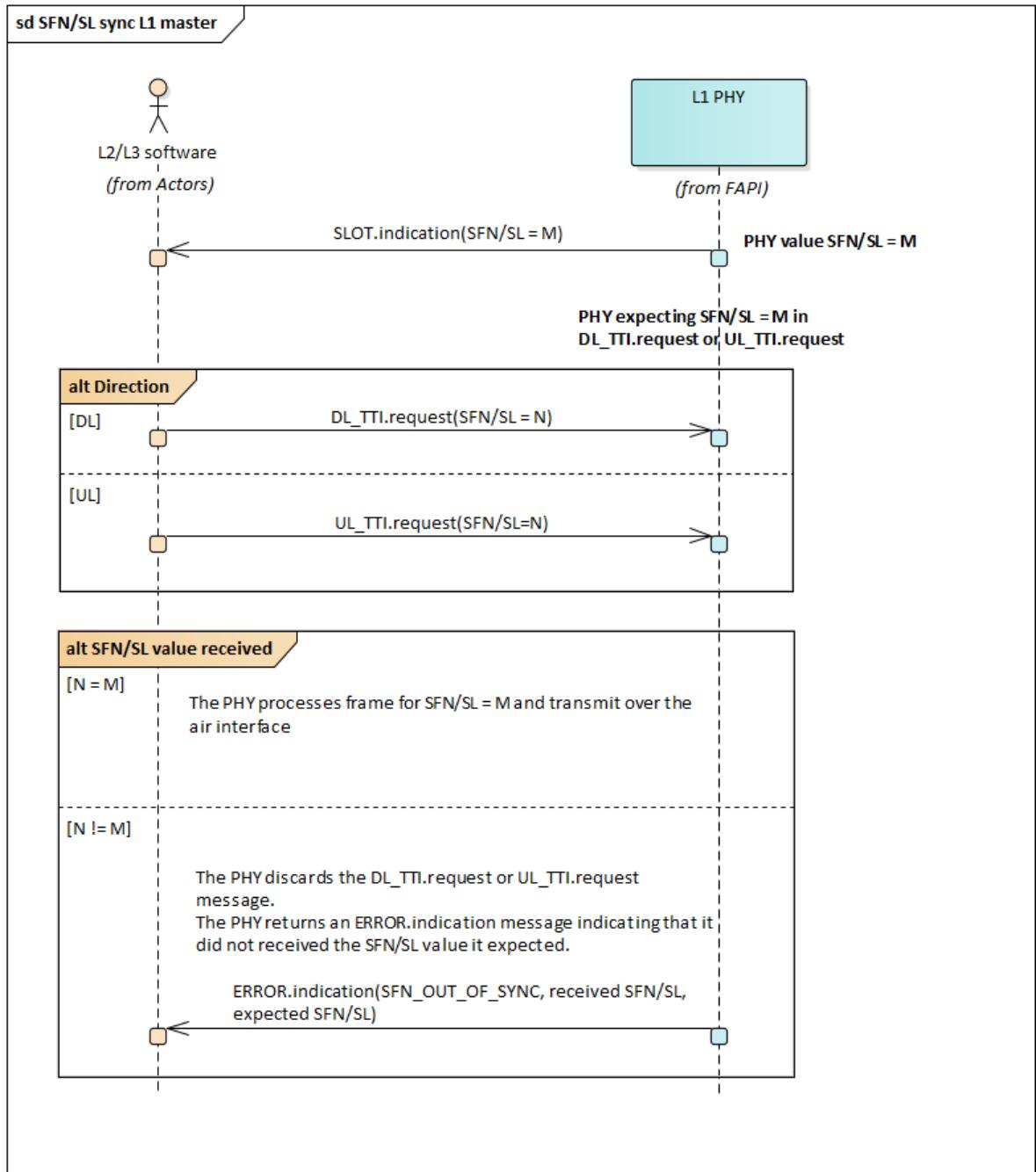


Figure 2-22 SFN/SL synchronization with L1 master

2.2.3a Slot Numerology

L2-initiated Slot-carrying P7 API shall set that slot value according to the highest numerology configured, among all the numerologies, across all uplink and downlink traffic directions and channels.

L2-initiated Channel-specific PDUs shall only be carried in those P7 API whose slot starts (according to the P7 slot numerology) coincide with the start of the channel slot (according to the Channel PDU numerology).



- E.g. If the highest P7 numerology is 60 kHz, 15 kHz PDUs (e.g. PUSCH PDU) are only expected in APIs (UL_TTI.request) where values of 60 kHz slots are divisible by 4

2.2.3 API message order

PHYs supporting Delay Management shall ignore any constraints relating to SLOT.indication or the relative order in which messages arrive for a particular slot. Instead, messages are expected to arrive according to their Receive Windows, as documented in section 2.1.3.5 of SCF-225 [10].

The PHY API has constraints of when certain messages can be sent, or will be received, by the L2/L3 software.

The downlink API message constraints are shown in Figure 2-23:

- The SFN/SL included in the SLOT.indication message is expected in the corresponding DL_TTI.request
- If the PHY is being reconfigured using the CONFIG.request message, this must be the first message for the slot.
- If the *Skip_blank_DL_CONFIG* option is indicated during the PARAM procedure, DL_TTI.request can be optionally skipped when there is nothing to schedule. Otherwise, DL_TTI.request must be sent for every downlink slot and must be the next message.
- If the *Skip_blank_UL_CONFIG* option is indicated during the PARAM procedure, UL_TTI.request can be optionally skipped when there is nothing to schedule. Otherwise, UL_TTI.request must be sent for every uplink slot and must be the next message
- The TX_DATA.request and UL_DCI.request messages are optional. It is not a requirement that they are sent in every downlink slot.
- L1 may indicate whether it can expect more than one instance of DL_TTI.request, UL_TTI.request, UL_DCI.request or TX_DATA for a slot. If supported by L1, L2 may configure L1 to expect only 1 DL_TTI.request, 1 UL_TTI.request and 1 UL_DCI.request for a slot.
- There can be more than one instance of the TX_DATA.request if mini-slot is used. This allows the PHY to minimize latency.

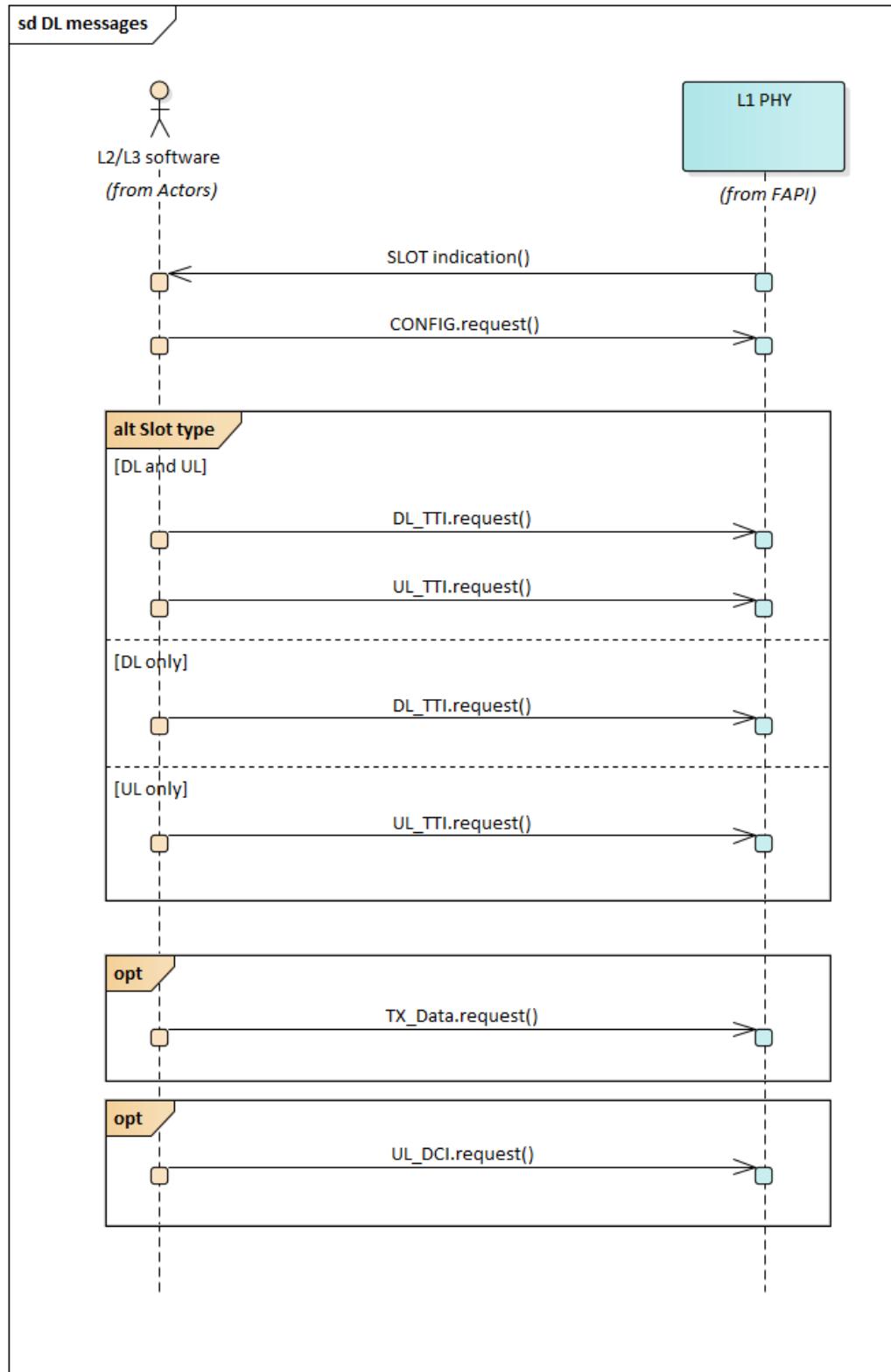


Figure 2-23 DL message order

The uplink API message constraints are shown in Figure 2-24:

- The UL API messages are optional. It is not a requirement that they are sent in every slot.



- If present, the messages can be in any order
 - The CRC.indication message is included if uplink data PDUs were expected in the slot.
 - The RX_DATA.indication message is included if uplink data PDUs were expected in the slot.
 - The UCI.indication message is included if UCI PDUs were expected in the slot.
 - The RACH.indication message is included if any RACH preambles were detected in the slot
 - The SRS.indication message is included if any sounding reference symbol information is expected in the slot.
- L1 may indicate whether it can generate more than one instance of CRC.indication, RX_DATA.indication, UCI.indication, RACH.indication, SRS.indication or DL_TTI.response per slot and subcarrier spacing. If supported by L1, L2 may configure L1 to limit itself to only one instance of the CRC.indication, RX_DATA.indication, UCI.indication, RACH.indication, and SRS.indication messages per slot and subcarrier-spacing.
- There can be more than one instance of the CRC.indication, RX_DATA.indication, UCI.indication, RACH.indication, and SRS.indication messages if more than one subcarrier spacing finishes in the same slot. As an example, from Figure 2-18, slot 7 for u=3 completes at the same time as slot 1 for u=1.
- There can be more than one instance of the CRC.indication and RX_DATA.indication if mini-slot is used. This allows the PHY to minimize latency.

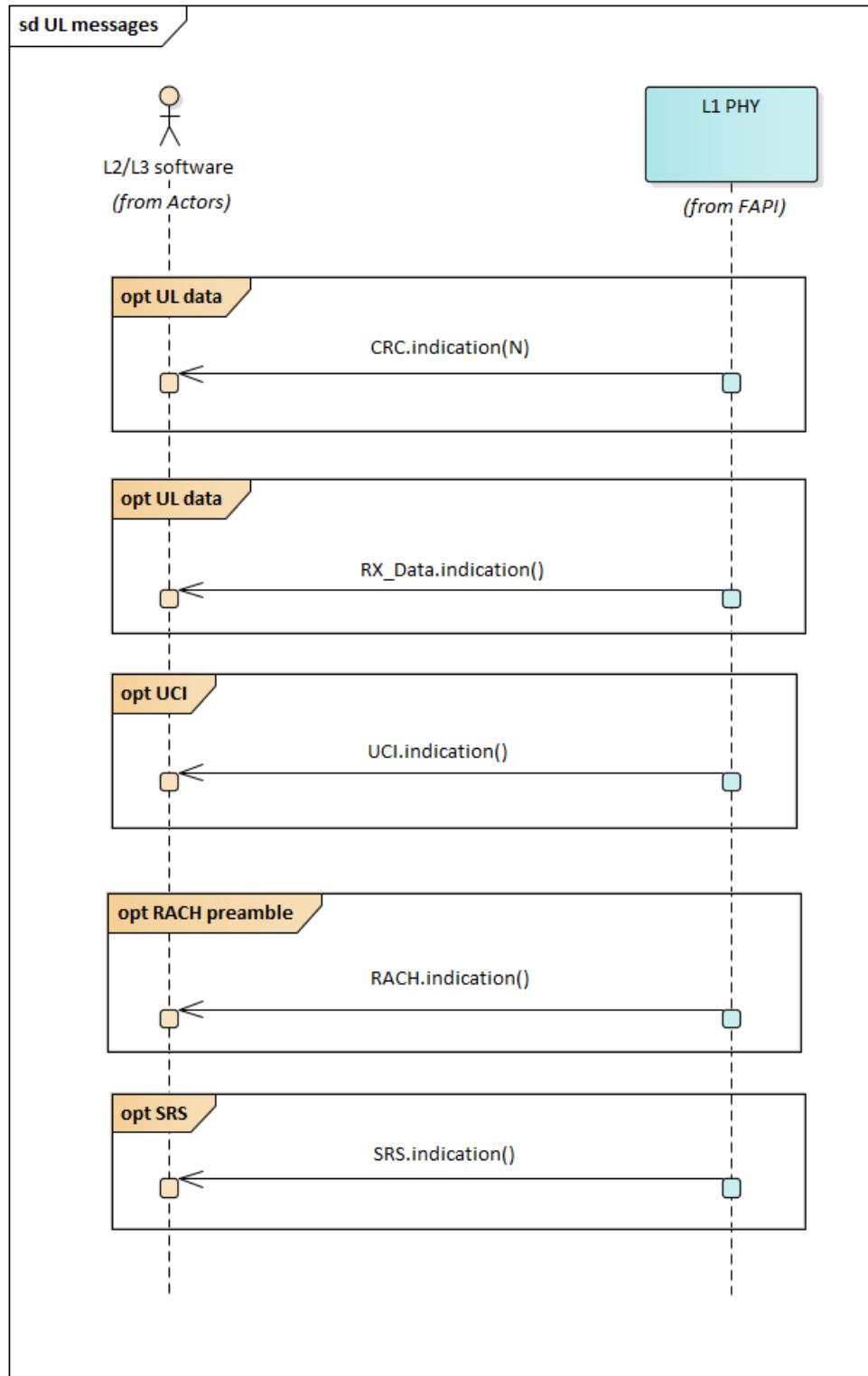


Figure 2-24 UL message order

2.2.4 Downlink

The procedures relating to downlink transmission are described in this Section.



2.2.4.1 BCH

The BCH transport channel is used to transmit the Master Information Block (MIB) information to the UE, per 3GPP [3GPP TS 38.331 [6]]. It has a periodicity of 80ms. BCH payload is carried using the PBCH physical channel. Together with the PSS and the SSS synchronization signals, PBCH forms the Synchronization Signal Block (SSB) as shown in Figure 2-25 in 3GPP TS 38.300 [8].

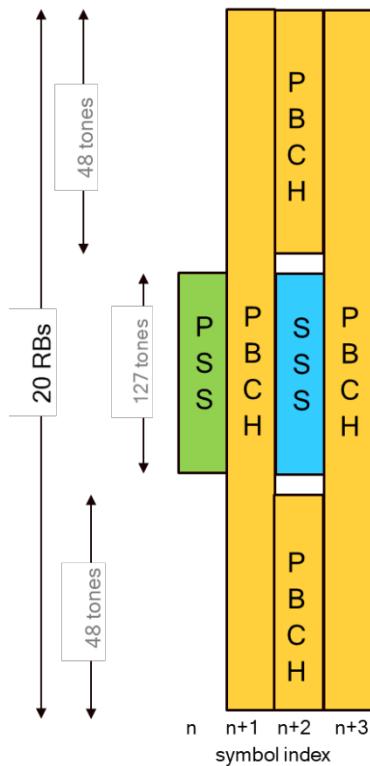


Figure 2-25 Time and frequency structure of the synchronization signal block (SSB)

SS blocks can be transmitted in different beams in a time-multiplexed fashion. The set of SS blocks within a beam-sweep is called an *SS burst*

The periodicity of the SS burst set varies from 5 to 160ms. All SS blocks in an SS burst set needs to be transmitted within 5ms. The SS blocks can be transmitted only in predetermined time slots. Depending on the band of deployment the location of the SSB allowed slots vary as shown in Figure 2-26 and Figure 2-27.



For SS with 15kHz and 30kHz SCS

Max possible values of L shown for every SCS

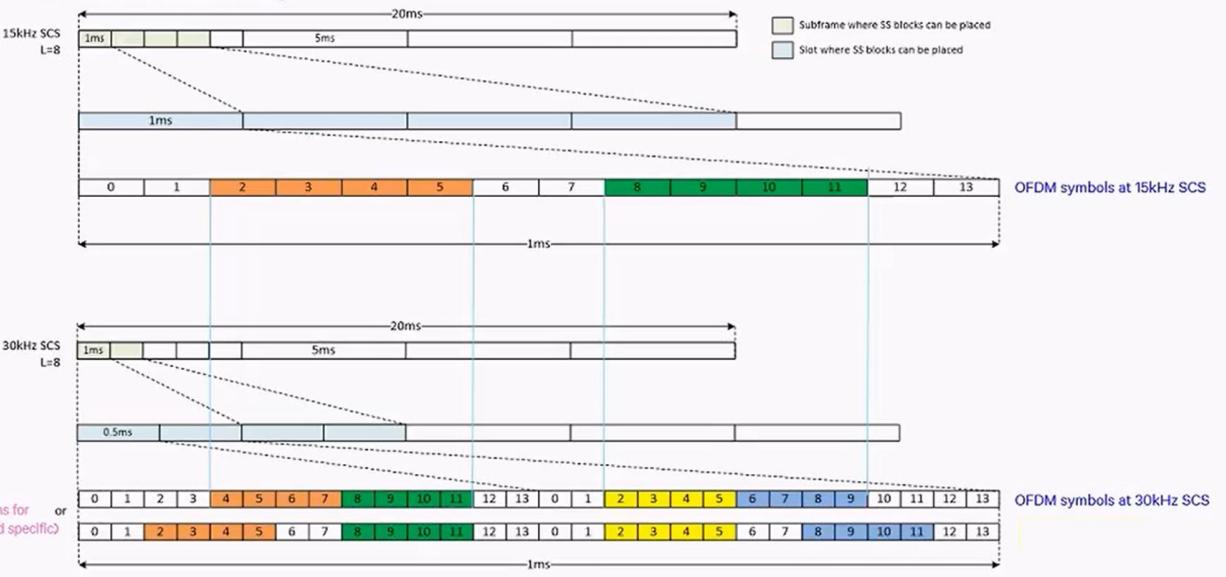


Figure 2-26 Possible SS block locations for FR1 (sub6 GHz) deployments

For SS with 120kHz and 240kHz SCS

Max possible values of L shown for every SCS

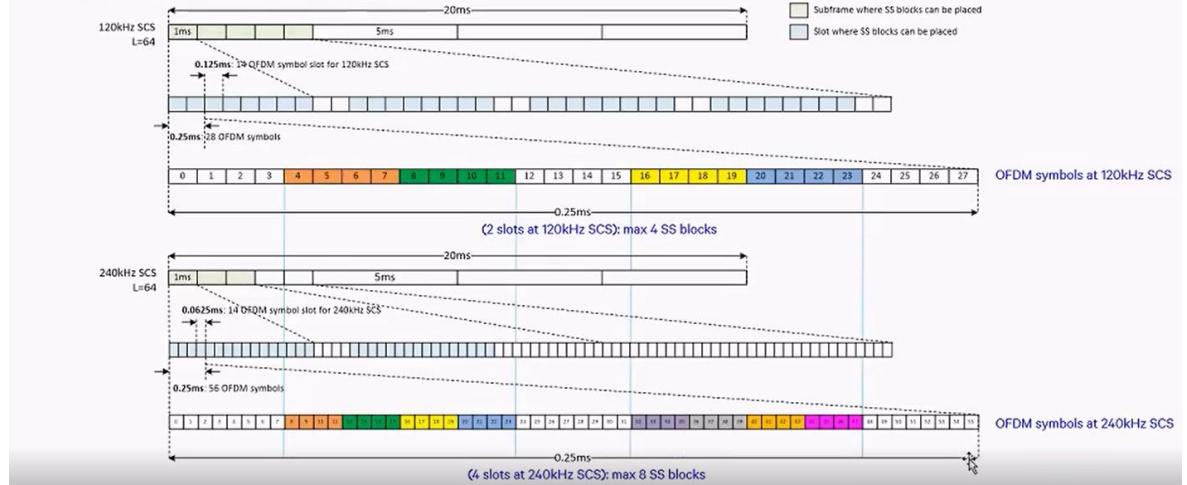


Figure 2-27 Possible SS block locations for FR2 deployments

2.2.4.2 PCH

The PCH transport channel is used to transmit paging messages to the UE. The UE has specific paging occasions where it listens for paging information Figure 2-28. The L2/L3 software is responsible for calculating the correct paging occasion for a UE. The PHY is only responsible for transmitting PCH PDUs when instructed by the DL_TTI.request message.

The PCH procedure is shown in Figure 2-28. To transmit a PCH PDU the L2/L3 software must provide the following information:

- In DL_TTI.request a PDSCH PDU and PDCCH PDU are included.



- In TX_DATA.request a MAC PDU containing the paging message is included.

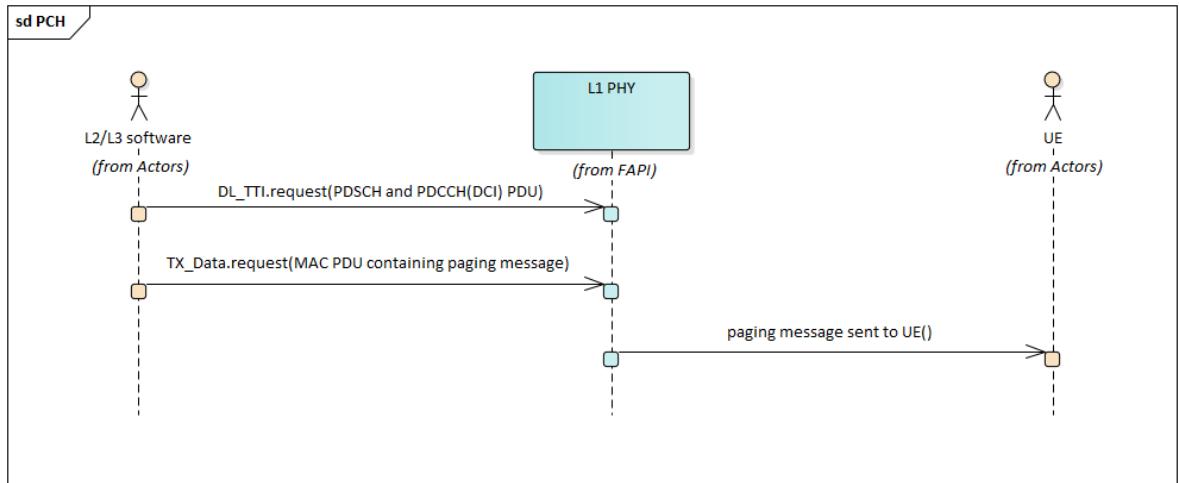


Figure 2-28 PCH procedure

2.2.4.3 DL-SCH

The DL-SCH transport channel is used to send data from the gNB to a single UE. HARQ is always applied on the DL-SCH transport channel at least for unicast PDSCH. Therefore, when scheduling a downlink PDSCH transmission which will require HARQ-ACK feedback from UE, the L2/L3 software has to schedule uplink transmission on PUCCH or PUSCH for the UE to feed back an ACK/NACK response.

The procedure for the DL-SCH transport channel is shown in Figure 2-29. To transmit a DL-SCH PDU, the L2/L3 software must provide the following information:

- In DL_TTI.request a PDSCH PDU and PDCCH PDU are included. The PDCCH PDU contains control regarding the DL frame transmission
- In TX_DATA.request a MAC PDU containing the data is included
- At the expected slot UCI HARQ information is included in a later UL_TTI.request, where the timing of this message is variable. The HARQ can be sent on the PUSCH or PUCCH, therefore, the information of the HARQ response on the uplink is sent in either:
 - PUSCH PDU – is used if the UE is scheduled to transmit data and the ACK/NACK response
 - PUCCH PDU – is used if the UE is just scheduled to transmit the ACK/NACK response
- The PHY will return the ACK/NACK response information in the UCI.indication message

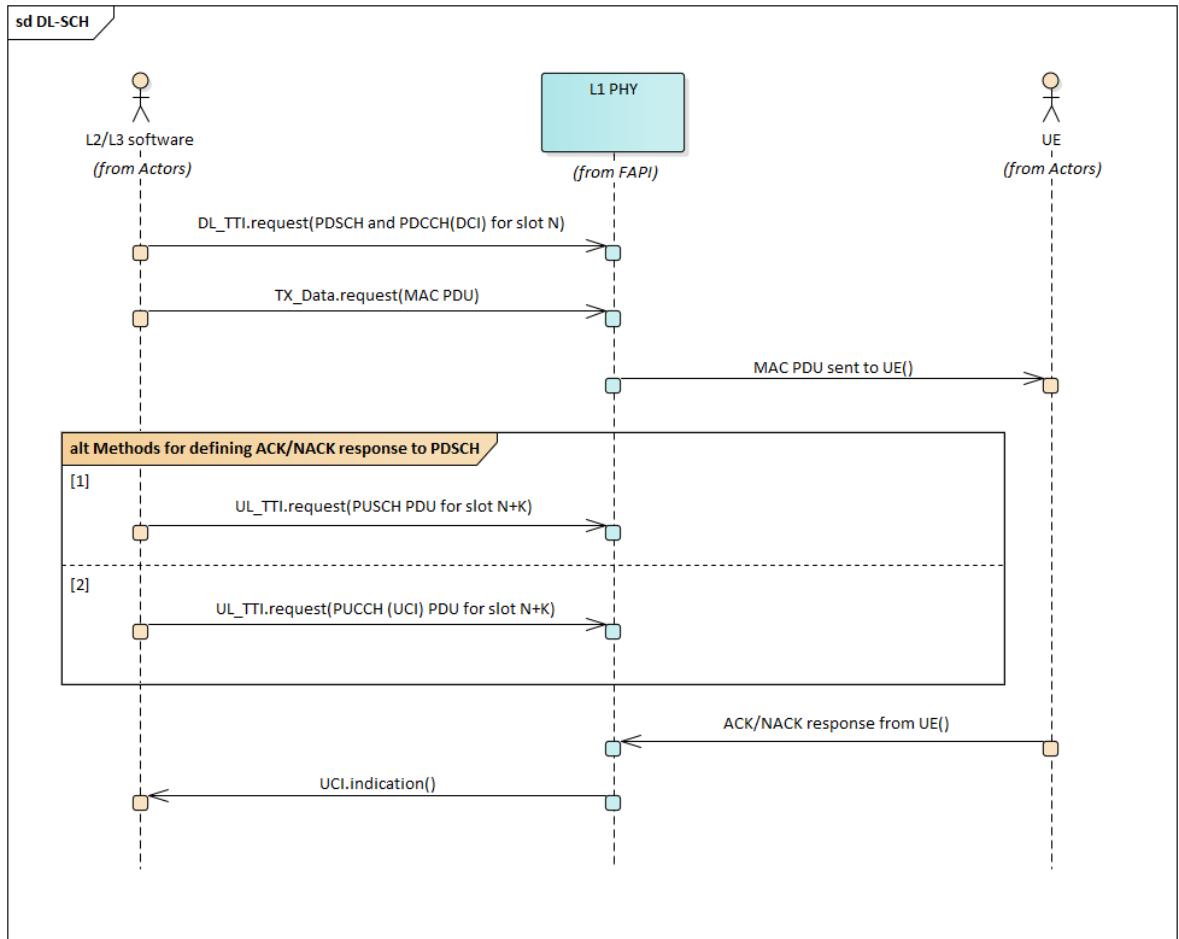


Figure 2-29 DL-SCH procedure

With DCI Format 1-1, the DL SCH channel can send two data transport blocks to a UE. This requires a single PDCCH (DCI) PDU, a single PDSCH PDU and two MAC PDUs. The procedure is shown in Figure 2-30. To initiate a transmission of two data transport blocks, the L2/L3 software must provide the following information:

- In the first transmission slot in the `DL_TTI.request` a PDSCH and PDCCH (DCI) PDU is included. The PDCCH PDU contains control regarding the DL frame transmission. A PDSCH PDU includes two codewords, one for each transport block specified in the DCI PDU.
- In `TX_DATA.request` two MAC PDUs containing the data are included.

(The remaining behaviour is identical to single-layer transmission.)

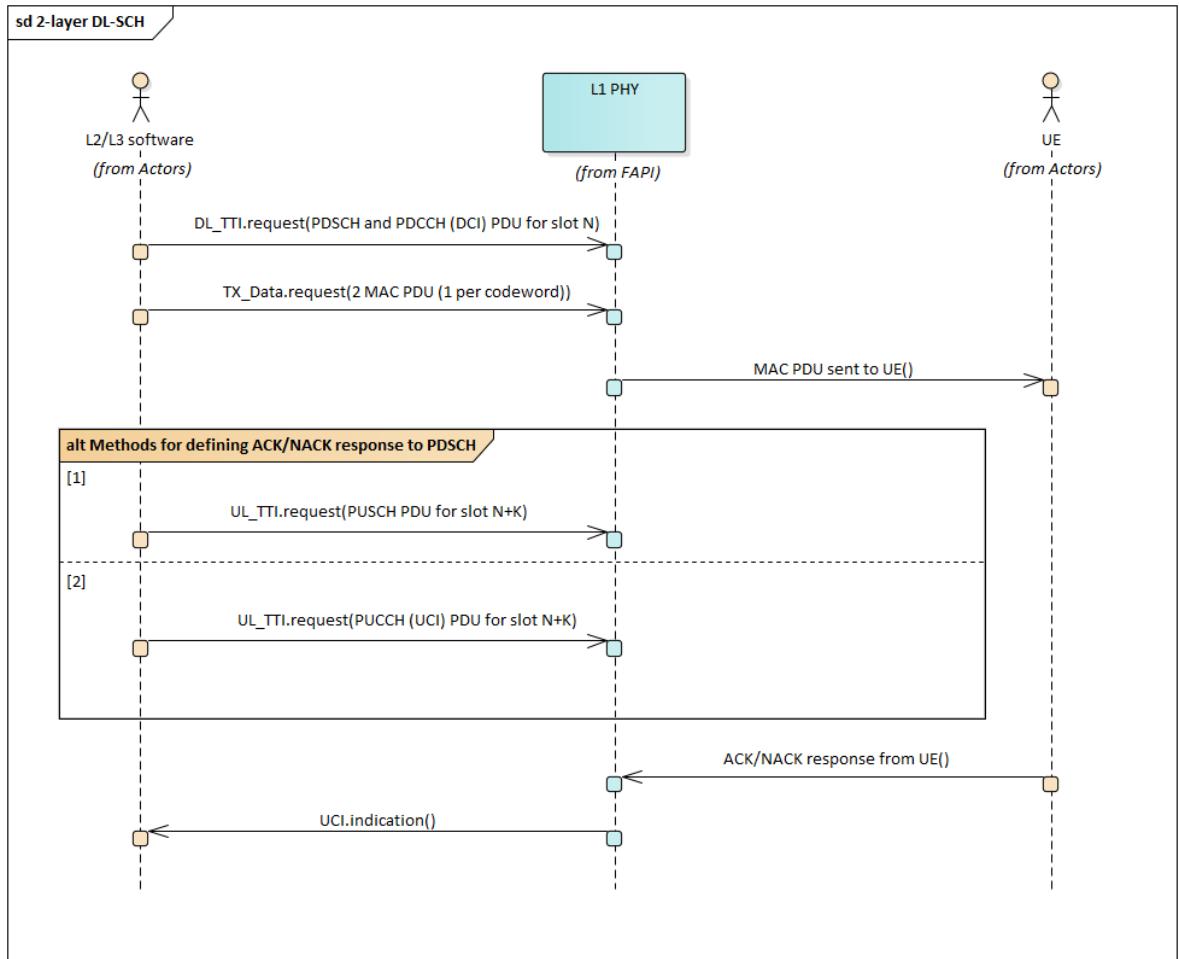


Figure 2-30 2-layer DLSCH procedure

Multi-slot transmission is also an option for the DL-SCH, where the same MAC PDU is transmitted for N slots. The procedure is shown in Figure 2-31, and the L2/L3 must provide the following information:

- In `DL_TTI.request` a PDSCH PDU and PDCCH (DCI) PDU is included. The PDCCH PDU contains control regarding the DL frame transmission and the UE will have previously been configured to be aware that multi-slot transmission is used.
- In `TX_DATA.request` a MAC PDUs containing the data is included.
- The multi-slot transmission can be over 2,4 or 8 slots and for the remaining slots the L2/L3 software must provide the following information:
 - In `DL_TTI.request` a PDSCH PDU is included. The PDSCH PDU contains the information for this slot, for example, in multi-slot transmission the incremental redundancy version changes per slot.
 - In `TX_DATA.request` a MAC PDUs containing the data is included

(The remaining behaviour is identical to single-layer transmission.)

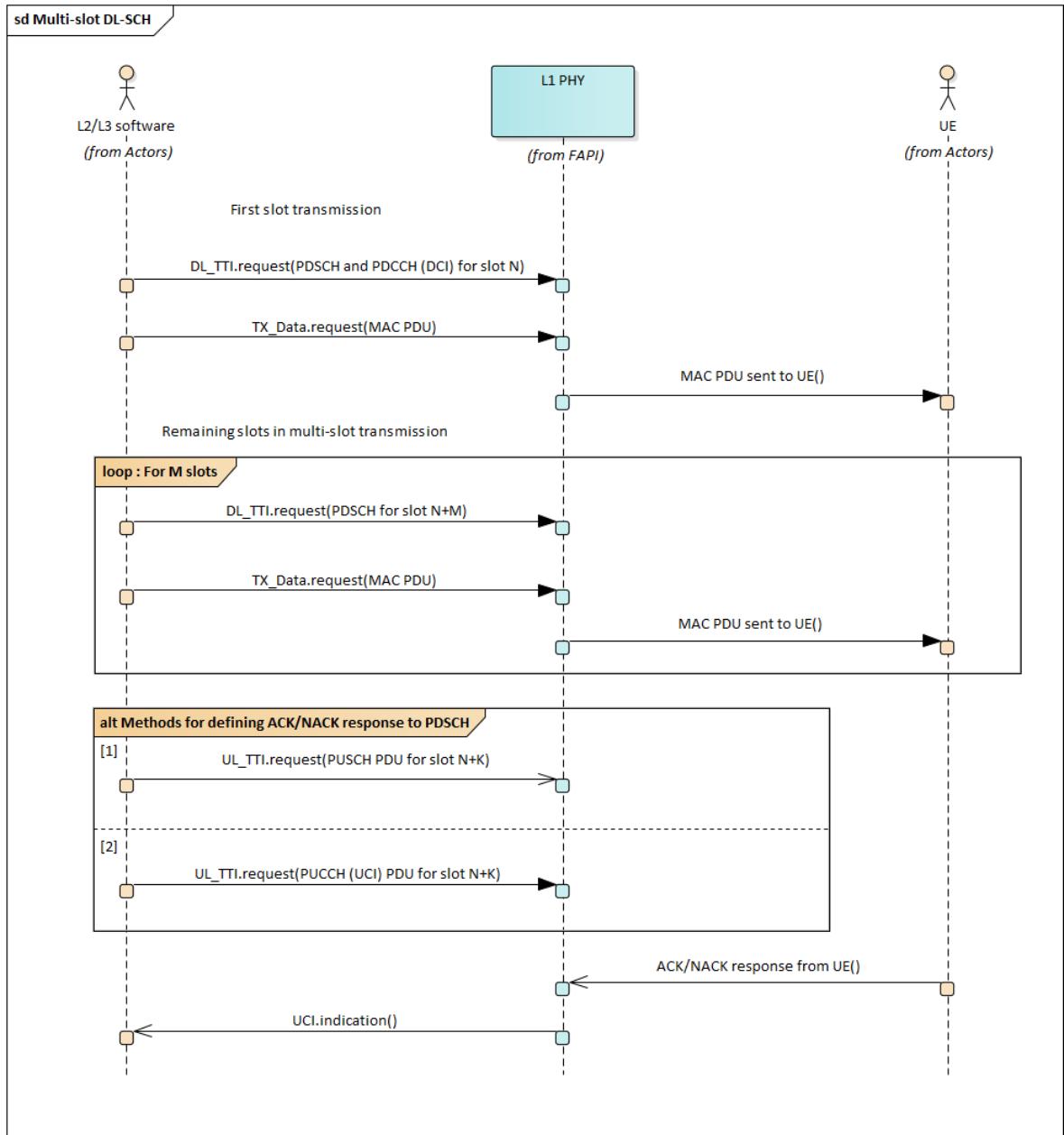


Figure 2-31 Multi-slot DL-SCH procedure

2.2.4.4 Downlink Reference Signals

The downlink includes several reference signals:

- DMRS for PDSCH and PDCCH
- PTRS for PDSCH
- CSI-RS
- PRS

The reference signals transmitted for either PDSCH or PDCCH are included in the `DL_TTI.request` PDSCH or PDCCH PDUs, respectively. However, to transmit a CSI-RS, respectively PRS the `DL_TTI.request` includes a CSI-RS PDU, respectively a PRS PDU. This behaviour is shown below in Figure 2-32, where:



- For a DL data transmission DMRS is included for both PDSCH and PDCCH
 - In `DL_TTI.request` the PDSCH and PDCCH PDUs are included, these PDUs include DMRS information.
 - A `TX_DATA.request` is sent to the PHY including the MAC PDU
 - The MAC PDU is sent to the UE with DMRS in both the PDSCH and PDCCH transmission.
- For a DL data transmission optionally PTRS can be enabled
 - In `DL_TTI.request` the PDSCH and PDCCH PDUs are included. The PDSCH PDUs will indicate that optional PTRS is included and both PDUs include DMRS information.
 - A `TX_DATA.request` is sent to the PHY including the MAC PDU
 - The MAC PDU is sent to the UE with PTRS in the PDSCH transmission and DMRS in both the PDSCH and PDCCH transmission.
- CSI-RS and PRS transmissions are not dependent on PDSCH transmission, or on each other. Instead, CSI-RS or PRS can be sent to a UE in the same slot as a PDSCH transmission, or in a slot with no PDSCH transmission for the UE. There is also no dependency between CSI-RS and PRS transmissions.
 - For CSI-RS, in `DL_TTI.request` the CSI-RS PDU is included. This results in a CSI-RS transmission to the UE.
 - For PRS, in `DL_TTI.request` the PRS PDU is included. This results in a PRS transmission to the UE.

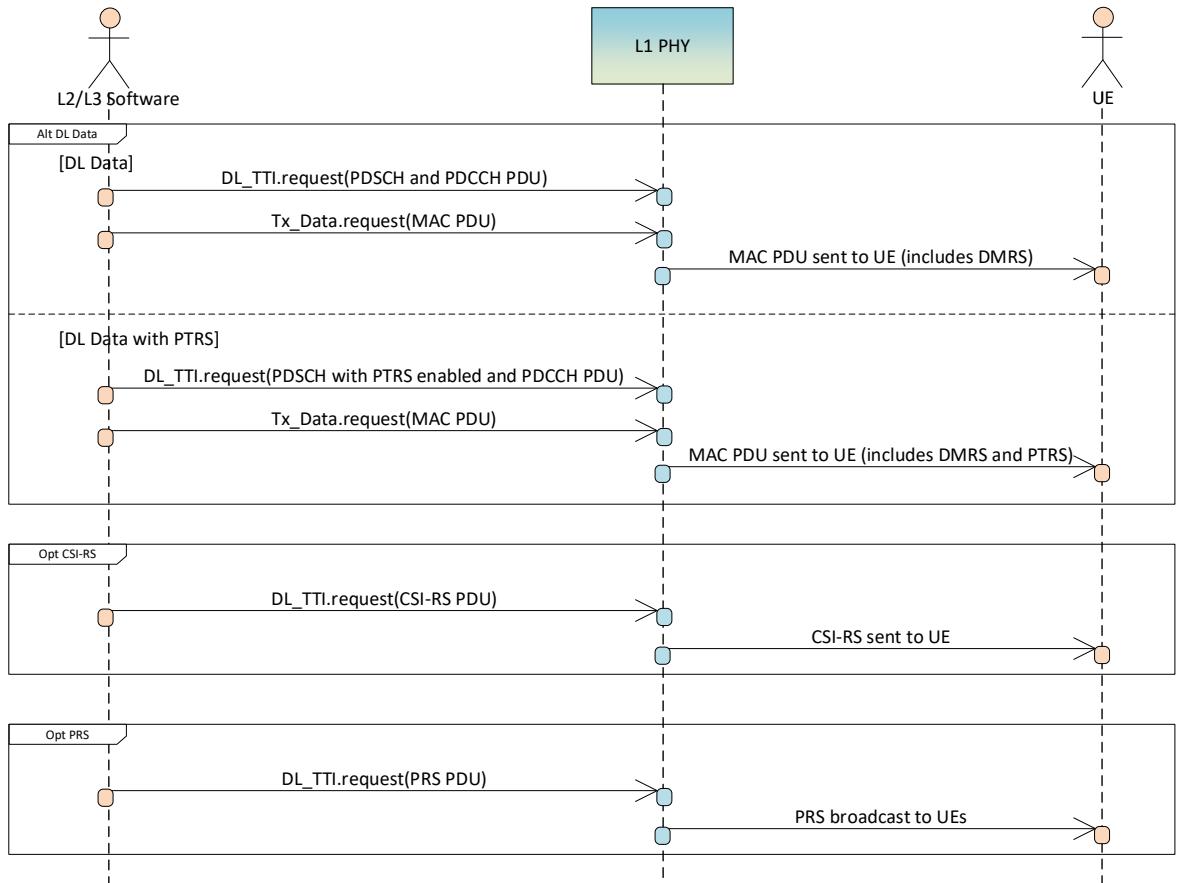


Figure 2-32 Downlink reference signals

2.2.4.5 Transmit Power Levels

In this specification version, two power profiles are supported to signal the transmit power levels in the `DL_TTI.request` message (PDSCH, PDCCH, CSI-RS and SSB) PDUs:

- ProfileNR: relative power levels are signalled based on 3GPP NR specification relationships.
- ProfileSSS: DMRS, CSI-RS and PSS relative power levels are signalled with respect SSS. Data and PT-RS power levels may also be signalled with respect to SSS, or based on 3GPP NR specification relationships.

In all cases:

- L1 may offer the capability to configure baseband scaling of SS/PBCH with respect to the nominal SSS signal generation in 3GPP TS 38.211 [2], section 7.4.2.3.1.

ProfileNR

The power offset parameters used for ProfileNR are shown in Figure 2-33. The PDUs and respective power offset values included are:

- PDCCH PDU



- powerControlOffsetSS[ProfileNR]
- PDSCH PDU
 - powerControlOffsetSS[ProfileNR]
 - powerControlOffset[ProfileNR]
 - β_{DMRS} (numDmrsCdmGrpsNoData and dmrsConfigType)
- CSI-RS PDU
 - powerControlOffsetSS[ProfileNR]
 - powerControlOffset[ProfileNR]
- PRS PDU
 - prsPowerOffset
- SSB PDU
 - β_{PSS} – (betaPSS[ProfileNR])

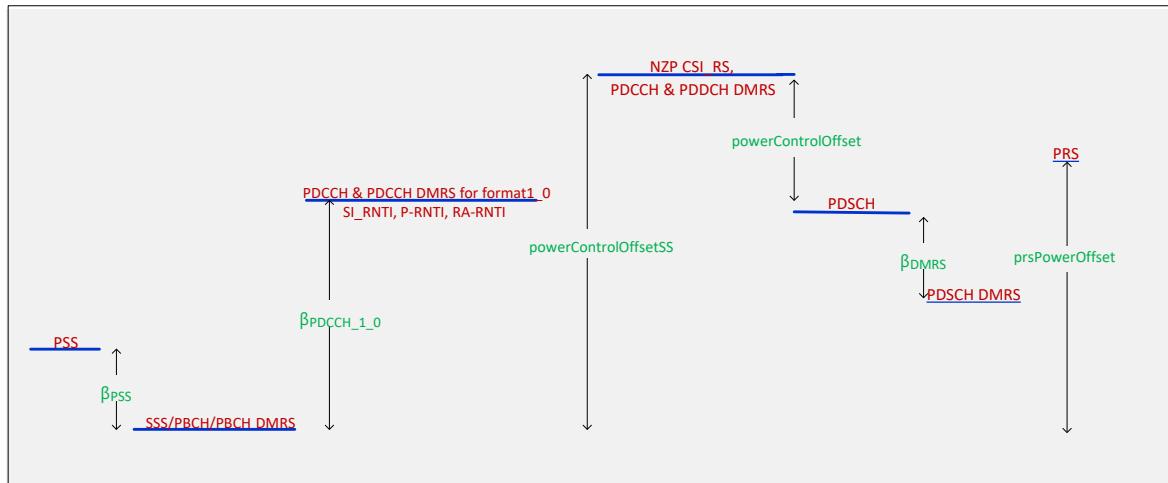


Figure 2-33 ProfileNR: Transmit power levels

ProfileSSS

The power offset parameters used for ProfileSSS are shown in Figure 2-34. The PDUs and respective power offset values included are:

- PDCCH PDU
 - pdcchDmrsPowerOffset[ProfileSSS]
 - pdcchDataPowerOffset[ProfileSSS]
- PDSCH PDU
 - pdschDmrsPowerOffset[ProfileSSS]
 - pdschDataPowerOffset[ProfileSSS]
 - pdschPtrsPowerOffset[ProfileSSS]
- CSI-RS PDU



- csiRsPowerOffset[ProfileSSS]
- PRS PDU
 - prsPowerOffset
- SSB PDU
 - β_{PSS} – (betaPSS[ProfileSSS])

In the cases of some signals, the L1 maybe be signalled – if supported, and configured – to make use of 3GPP-based power offsets, in which case ProfileNR parameters are used for those signals.

1. pdcchDataPowerOffset[ProfileSSS]
2. pdschDataPowerOffset[ProfileSSS]
3. pdschPtrsPowerOffset[ProfileSSS]
4. betaPSS[ProfileSSS]

In Figure 2-34, these parameters are marked in *[bracketed, italicized captions]*.

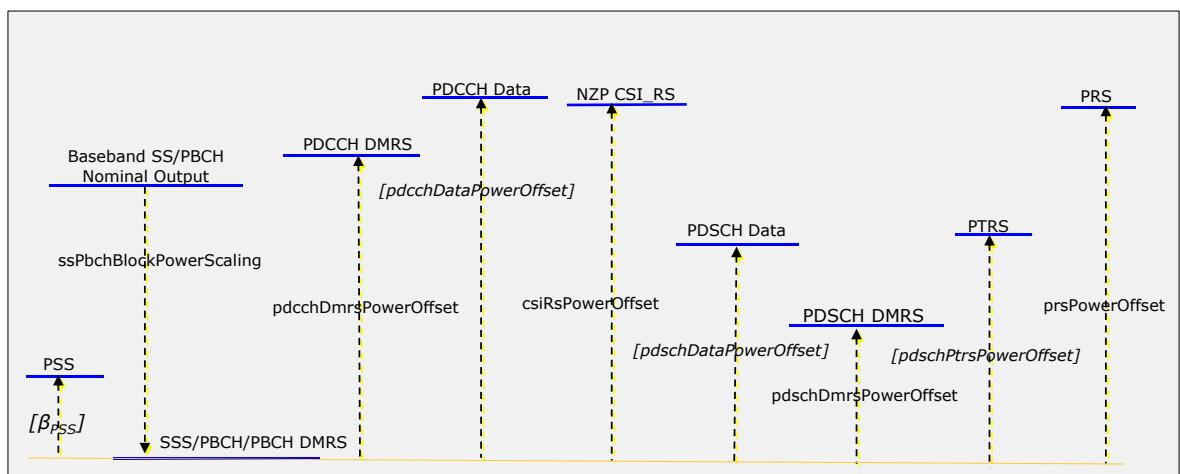


Figure 2-34 ProfileSSS: Transmit power levels

2.2.5 Uplink

The procedures relating to uplink reception are described in this Section.

2.2.5.1 RACH

The RACH transport channel is used by the UE to send data to the gNB when it has no scheduled resources. Also, the L2/L3 software can indicate to the UE that it should initiate a RACH procedure.

In the scope of the PHY API, the RACH procedure begins when the PHY receives an `UL_TTI.request` message indicating the presence of a PRACH slot.

The RACH procedure is shown in Figure 2-35. To configure a RACH procedure the L2/L3 software must provide the following information:

- The `UL_TTI.request` is sent to the PHY including a RACH PDU.



- If a UE decides to RACH, and a preamble is detected by the PHY:
 - The PHY will include 1 RACH PDU for each FD and TD occasion in the `RACH.indication` message. This RACH PDU includes all detected preambles
- If no RACH preamble is detected by the PHY, then no `RACH.indication` message is sent

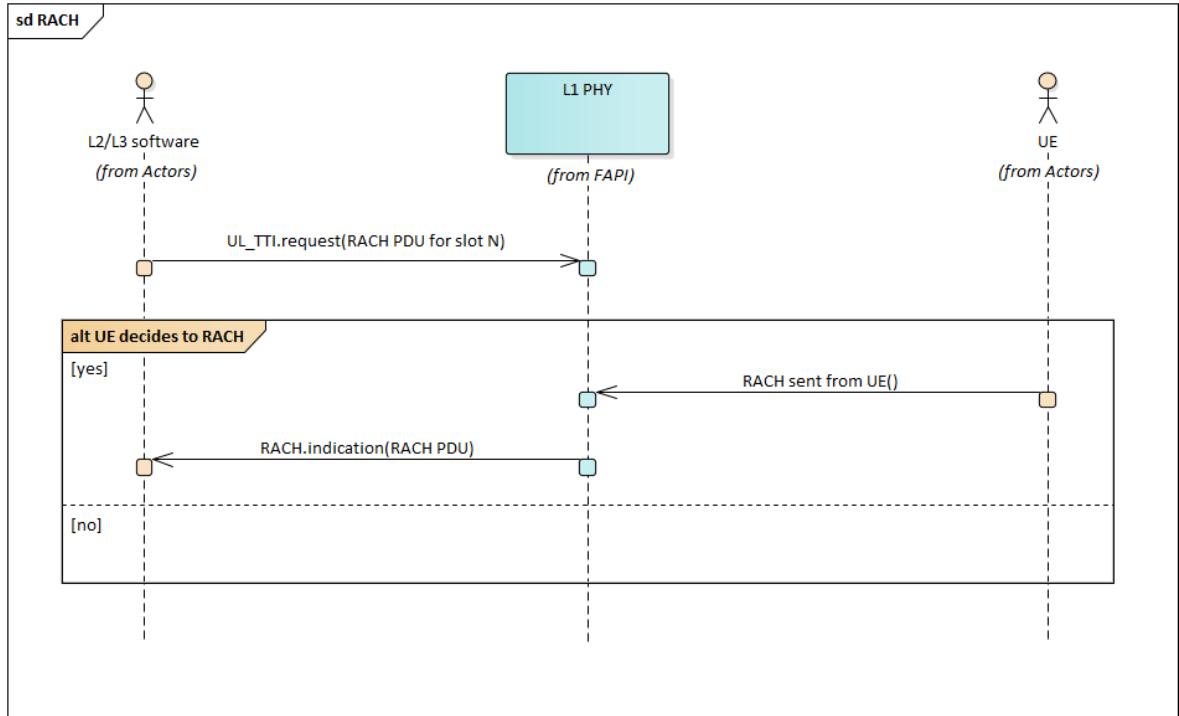


Figure 2-35 RACH procedure

In addition to the Release-15 RACH procedure, this FAPI release supports a 2-step RACH procedure, per 3GPP Rel-16 specifications [24][25], based on the a map linking MsgA-PRACH preambles and MsgA-PUSCH & associated DMRS occasions. The map may be:

- configured semistatically (in P5 MsgA-PUSCH configuration)
- signalled at slot resolution (in PRACH PDU in `UL_TTI.request`)

Conceptually, the 2-step RACH Procedure combines the RACH procedure above, with MsgA-PUSCH PDU signalling similar to UL-SCH section 2.2.5.2, where:

- the `UL_DCI.request` / control info is replaced by the above map configuration/signalling.
- The `UL_TTI.request` → PUSCH PDU is replaced by `UL_TTI.request` → MsgA-PUSCH PDU.

In particular, as illustrated in Figure 2-36

- For PHYs supporting semistatic configuration of a map linking MsgA-PRACH preambles and MsgA-PUSCH & associated DMRS occasions, MAC may choose to configures the map in P5 `CONFIG.request`, via the *Multi MsgA-PUSCH Config TLV*.



- An `UL_TTI.request` is sent to the PHY including a RACH PDU.
 - If and only if a semistatic map MsgA-PRACH preambles and MsgA-PUSCH & associated DMRS occasions was not configured via `P5_CONFIG.request`, a map shall be configured dynamically in the PRACH PDU.
- An `UL_TTI.request` is set to PHY including a MsgA-PUSCH PDU corresponding to the map signalled or configured earlier.
- Upon observing MsgA PRACH Preamble, PHY issues a `RACH.indication`
- Upon observing MsgA PUSCH corresponding to the preamble, PHY issues a `CRC.indication` for the carried MsgA PUSCH payload.
 - PHY determines the correspondence between the MsgA PRACH Preamble and the MsgA PUSCH observation from the earlier configured or signalled map.
 - If the CRC is successful, PHY also issues an `RX_DATA.indication` to signal the MsgA PUSCH payload.

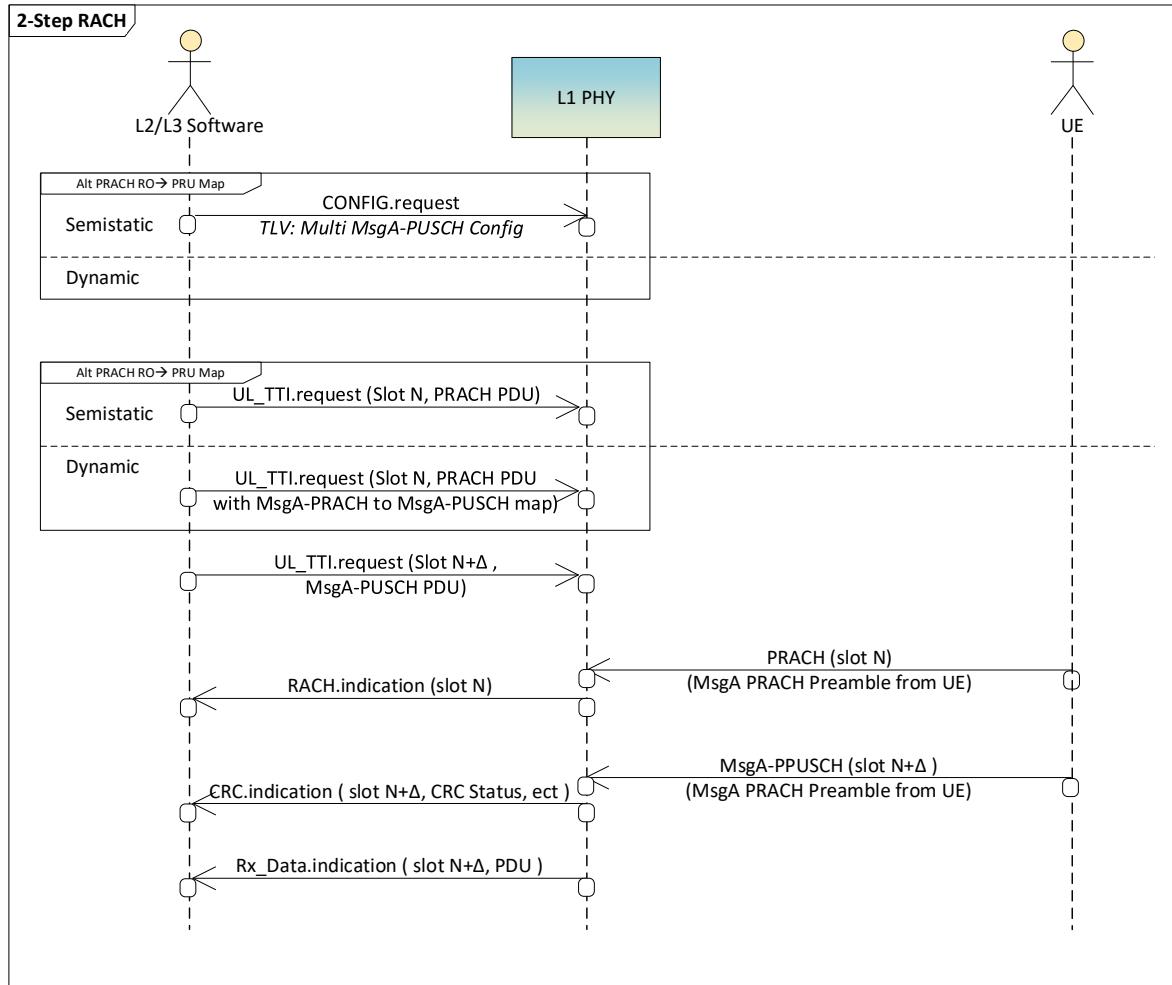


Figure 2-36: 2-Step RACH Procedure



Linking MsgA-PRACH preambles and MsgA-PUSCH & associated DMRS occasions

For P5-based MsgA-RACH → MsgA-PUSCH Maps:

- Each instance of the MsgA-PUSCH resource table (Table 3-43) in TLV 0x1032 applies to a single group (A or B)
- A separate PRACH configuration (pointed to by msgA-prachResConfigIndex) is needed to map to each Msg-A PUSCH resource in Table 3-43, via the map in Table 3-44.

For P7-based MsgA-RACH → MsgA-PUSCH Maps:

- A PRACH PDU may contain maps (Table 3-101) for multiple groups (A, B or dedicated), where multiple group-specific MsgA-PUSCH may be mapped in a single invocation of mapping Table 3-43.

A very simplified mapping illustration is based on the configuration in Figure 2-37:

- **Msg-A PRACH:**
 - consists of two time-domain occasions (rTD_0 and rTD_1) and two frequency-domain occasions (rFR_0 and rFD_1).
 - Accommodates two groups (A and B).
 - The height of the resources represents the number of preambles in each resources (e.g. here group B contains twice as many preambles as group A) – see section 8.1A of 3GPP TS 38.213 [22] for the detailed order of PRACH resources.
- **Msg-A PUSCH:**
 - Group-A consists of two time-domain occasions (uTD_0 and uTD_1) and two DMRS resources (uDR_a , uDR_b)
 - Group-B consists of two time-domain occasions (uTD_2 and uTD_3) and one DMRS resource (uDR_a)
 - The height of the resources represents the number of frequency-domain occasions in each resources.
 - Here, Group-A and Group-B are assigned the same number of frequency-domain occasions.
 - See section 8.1A of 3GPP TS 38.213 [22] for the detailed order of MsgA-PUSCH resources. The ordered indices of *MsgA-A PUSCH Occasions and associated DMRS resources* [22] constitute the space of PRU Preamble Indices, in this document.
 - Here, it can be seen that there are:
 - half as many Msg-A PUSCH frequency domain occasions than there are preambles in Msg-A PRACH Group-A
 - => 2 Msg-A PRACH preambles → 1 MsgA PUSCH resource for Group A
 - a quarter as many Msg-A PUSCH frequency domain occasions than there are preambles in Msg-A PRACH Group-B
 - => 4 Msg-A PRACH preambles → 1 MsgA PUSCH resource for Group B

For this extremely simplified illustration, the Msg-A PRACH → Msg-A PUSCH mapping is shown in Figure 2-38:

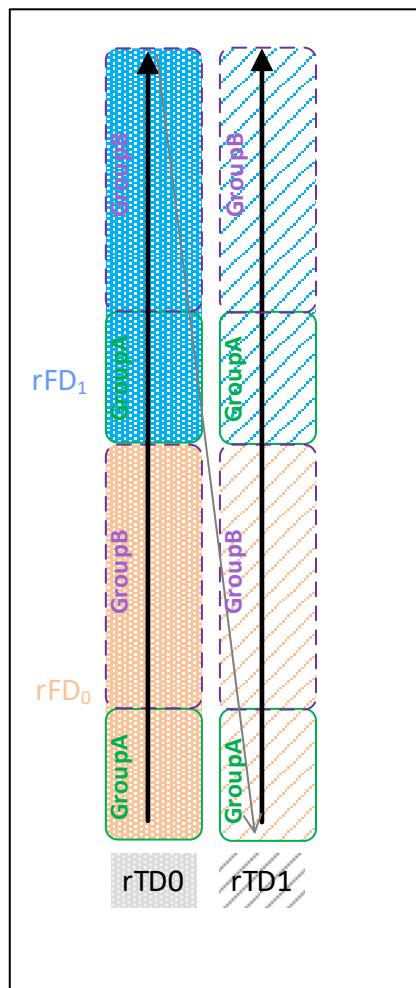
- P5-based mapping requires two invocations of mapping Table 3-44 (one for each Group A or B)
- P7-based mapping requires one invocation of mapping Table 3-44.



Notes:

- this example was specifically chosen to allow highly repetitive mapping for intuitive illustration. In general, P7-based mapping can require multiple invocations of mapping Table 3-44.
- the ordered list of Msg-A PUSCH resources in each Group constitutes *Preamble Space* for the Group.

Msg-A PRACH



Msg-A PUSCH

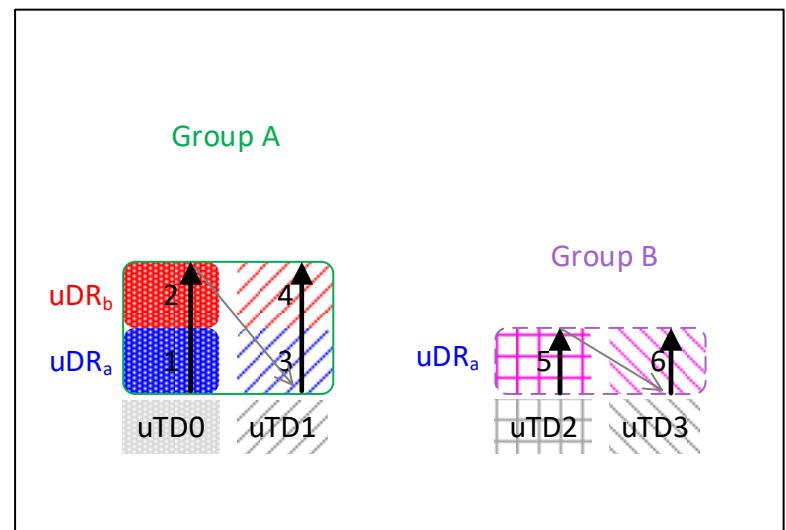


Figure 2-37: 2-Step RACH Msg-A PRACH and MsgA-PUSCH illustration

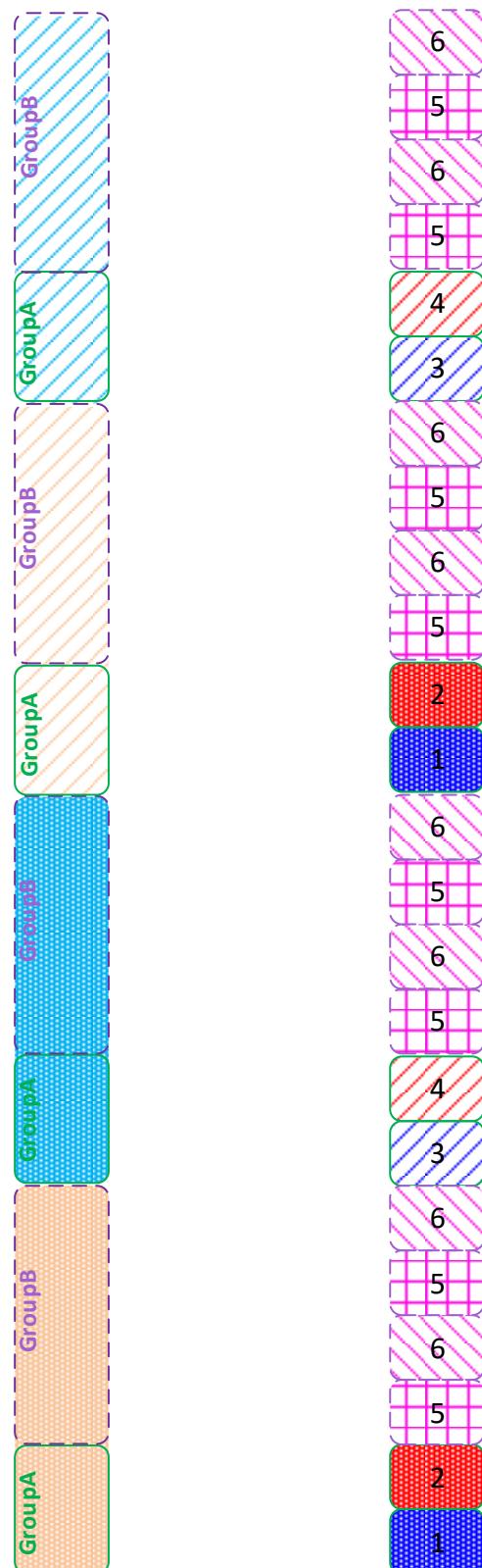


Figure 2-38: MsgA-PRACH → MsgA-PUSCH mapping illustration



2.2.5.2 UL-SCH

The UL-SCH transport channel is used to send data from the UE to the gNB. HARQ is always applied on the UL-SCH transport channel and the ACK/NACK value is indicated when the next transmission for the HARQ process is scheduled via DCI. Therefore, included in the next iteration of this procedure.

The procedure for the UL-SCH transport channel is shown in Figure 2-39. To transmit an UL-SCH PDU, the L2/L3 software must provide the following information:

- Within the `UL_DCI.request` for slot N a DCI PDU is included. The DCI PDU contains control information regarding the UL frame transmission being scheduled and indicates new data or retransmission for a specified HARQ process.
- In `UL_TTI.request` for slot N+K1 an ULSCH PDU is included, where the value of K1 is variable.
- The PHY will return CRC information for the received data in the `CRC.indication` message
- The PHY will return the received uplink data in the `RX_DATA.indication` message.
- If UCI information was expected in the uplink slot, the PHY will return the `UCI.indication` message.

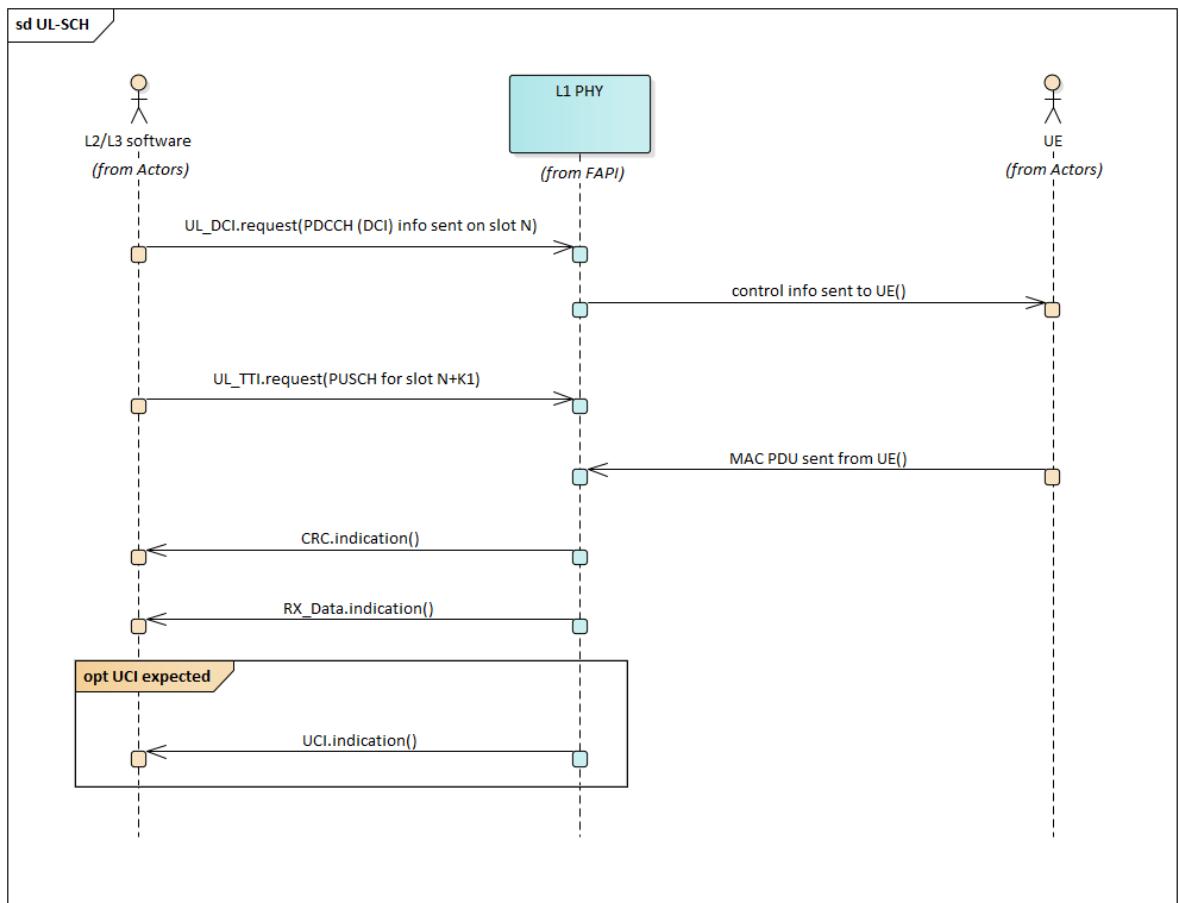


Figure 2-39 ULSCH procedure



Multi-slot transmission is also an option for the UL-SCH with configured grant, where the same MAC PDU is transmitted for N slots. The procedure is shown in Figure 2-40, and the L2/L3 must provide the following information:

- For each slot in the PUSCH transmission
 - In `UL_TTI.request` for slot N+M a PUSCH PDU is included, the UE will have previously been configured to be aware that multi-slot transmission is used.
 - The PHY will return CRC information for the received data in a the `CRC.indication` message
 - The PHY will return the received uplink data in the `RX_DATA.indication` message.

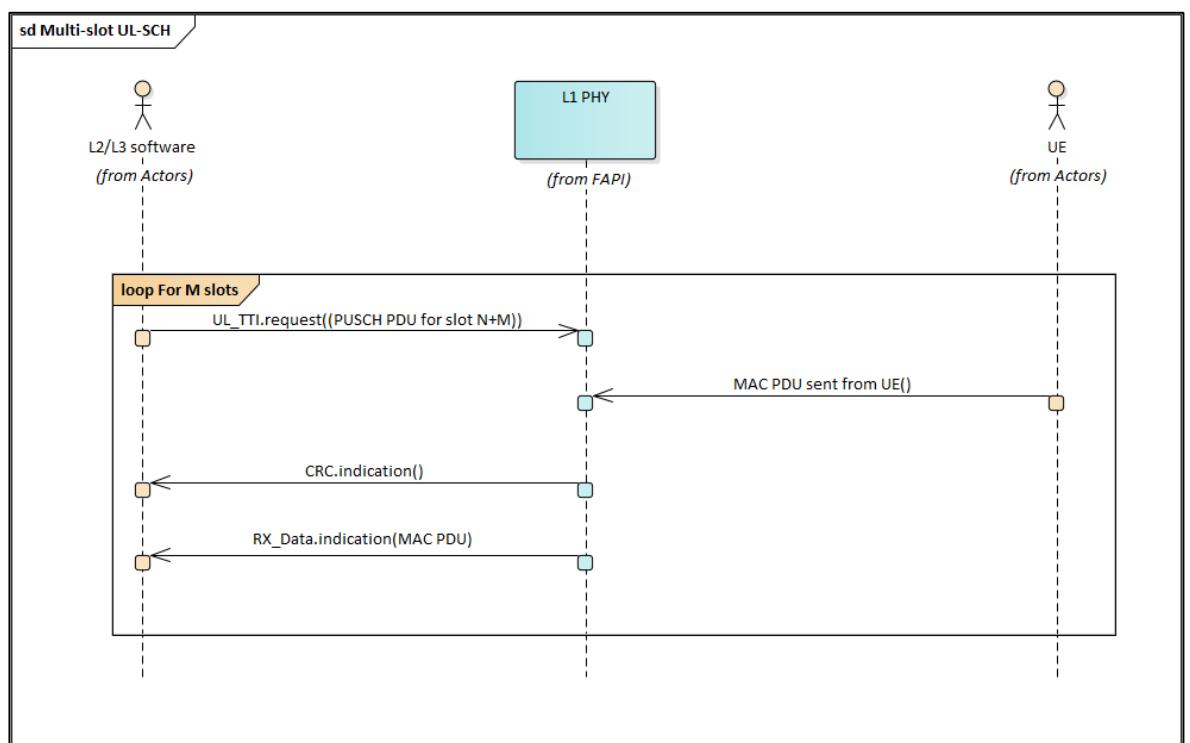


Figure 2-40 Multi-slot UL-SCH procedure

2.2.5.3 SRS

The sounding reference signal (SRS) is used by L2/L3 software to determine the quality of the uplink channel. The SRS procedure is shown in Figure 2-41. To schedule a SRS the L2/L3 software must provide the following information:

- In `UL_TTI.request` a SRS PDU is included.
- The PHY will return the SRS response to the L2/L3 software in the `SRS.indication` message

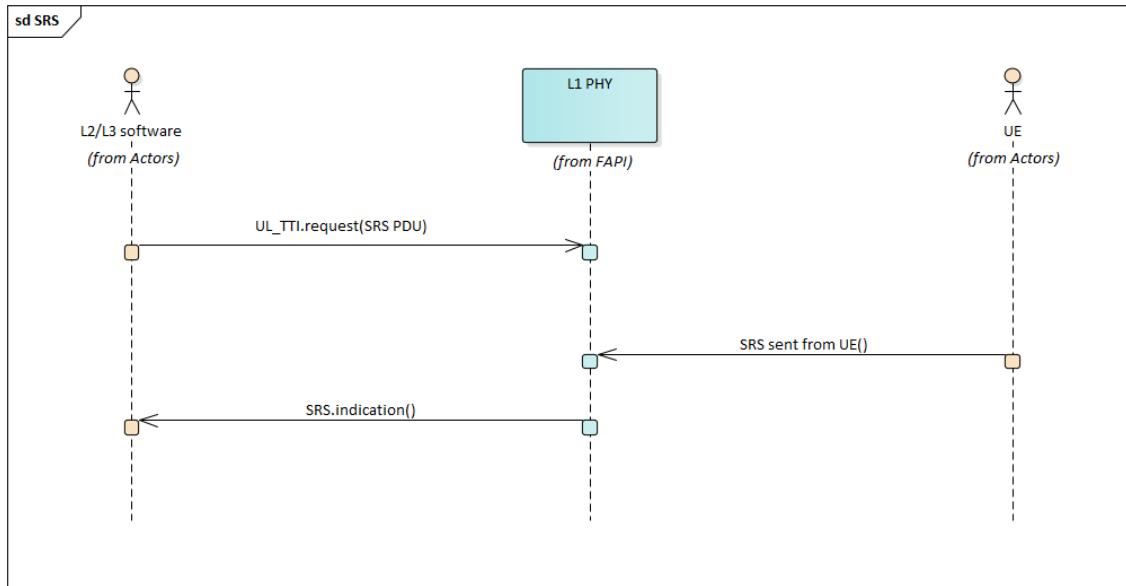


Figure 2-41 SRS procedure

2.2.5.4 CSI

The CSI reporting mechanism is used by the L2/L3 software to determine the quality of the downlink channel. The CSI reporting procedure is shown in Figure 2-42. To schedule a CSI report the L2/L3 software must provide the following information:

- For an aperiodic report the PDCCH PDU is included in the `UL_DCI.request`. This instructs the UE to send a CSI report. For periodic CSI report no explicit DCI information is sent.
- In the `UL_TTI.request`, where the L2/L3 software is expecting a CSI report:
 - If a PUSCH is scheduled for UCI then a PUSCH PDU is included with the CSI configuration information.
 - Otherwise, a PUCCH is scheduled with the CSI configuration information.
- The PHY will return the CSI report to the L2/L3 software in the `UCI.indication` message

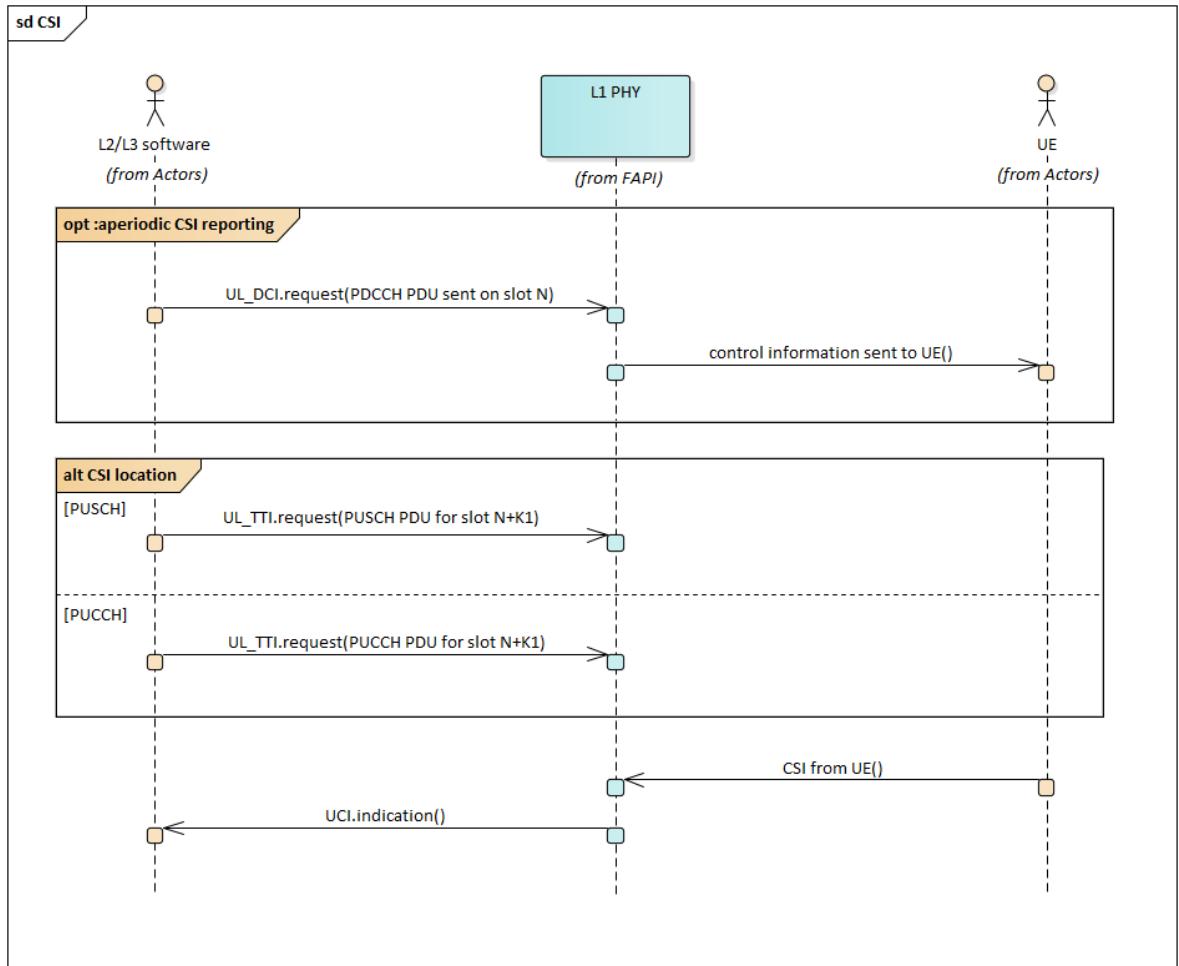


Figure 2-42 CSI procedure

2.2.5.5 SR

The scheduling request (SR) procedure is used by the UE to request additional uplink bandwidth. The L2/L3 software configures the SR procedure during the RRC connection procedure. The SR procedure is shown in Figure 2-43. To schedule a SR the L2/L3 software must provide the following information:

- In `UL_TTI.request` a PUCCH PDU is included.
- The PHY will return the SR to the L2/L3 software in the `UCI.indication` message.

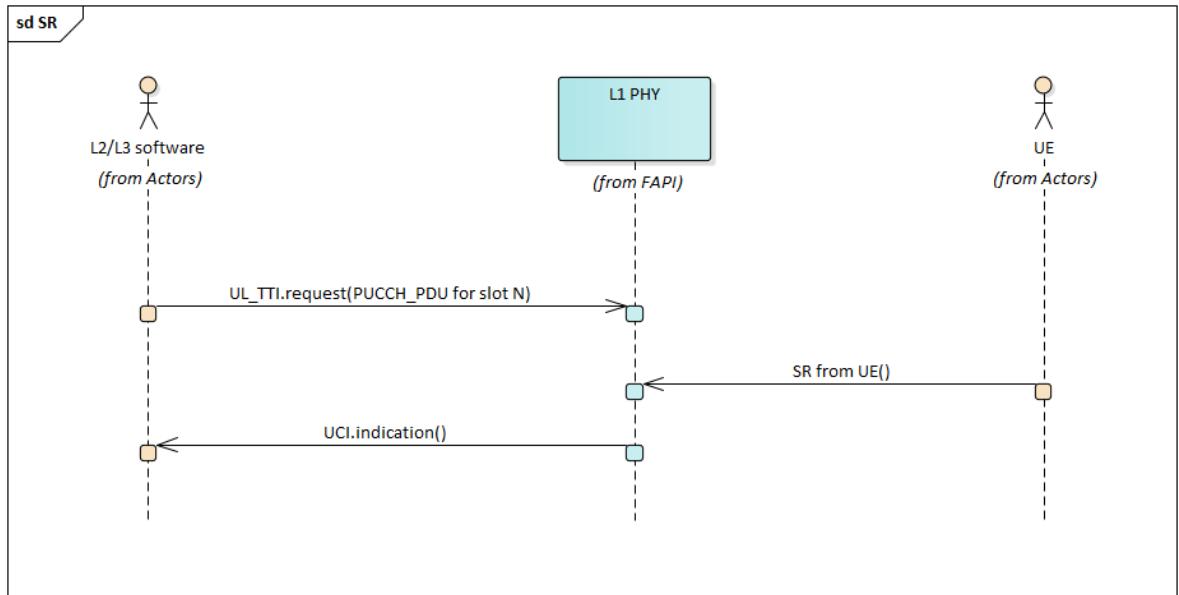


Figure 2-43 SR procedure

2.2.5.6 Uplink Reference Signals

The downlink includes several reference signals:

- DMRS for PUSCH and PUCCH
- PTRS for PUSCH
- SRS

The reference signals transmitted for either PUSCH or PUCCH are included in the `UL_TTI.request` PUSCH or PUCCH PDUs, respectively. However, to transmit a SRS the `UL_TTI.request` includes a SRS PDU. The SRS PDU is described in Section 2.2.5.3, while the other uplink reference signals are described below in Figure 2-44, where:

- For a UL data transmission DMRS is included for PUSCH
 - In `UL_TTI.request` the PUSCH PDU is included, this PDU includes DMRS information.
 - The MAC PDU is sent from the UE with DMRS.
- For a UL data transmission optionally PTRS can be enabled
 - In `UL_TTI.request` the PUSCH PDUs will indicate that optional PTRS is included and DMRS is included.
 - The MAC PDU is sent from the UE with PTRS and DMRS.
- For a UL UCI transmission DMRS is included for PUCCH
 - In `UL_TTI.request` the PUCCH PDU is included, this PDU includes DMRS information.
 - The UCI is sent from the UE with DMRS.

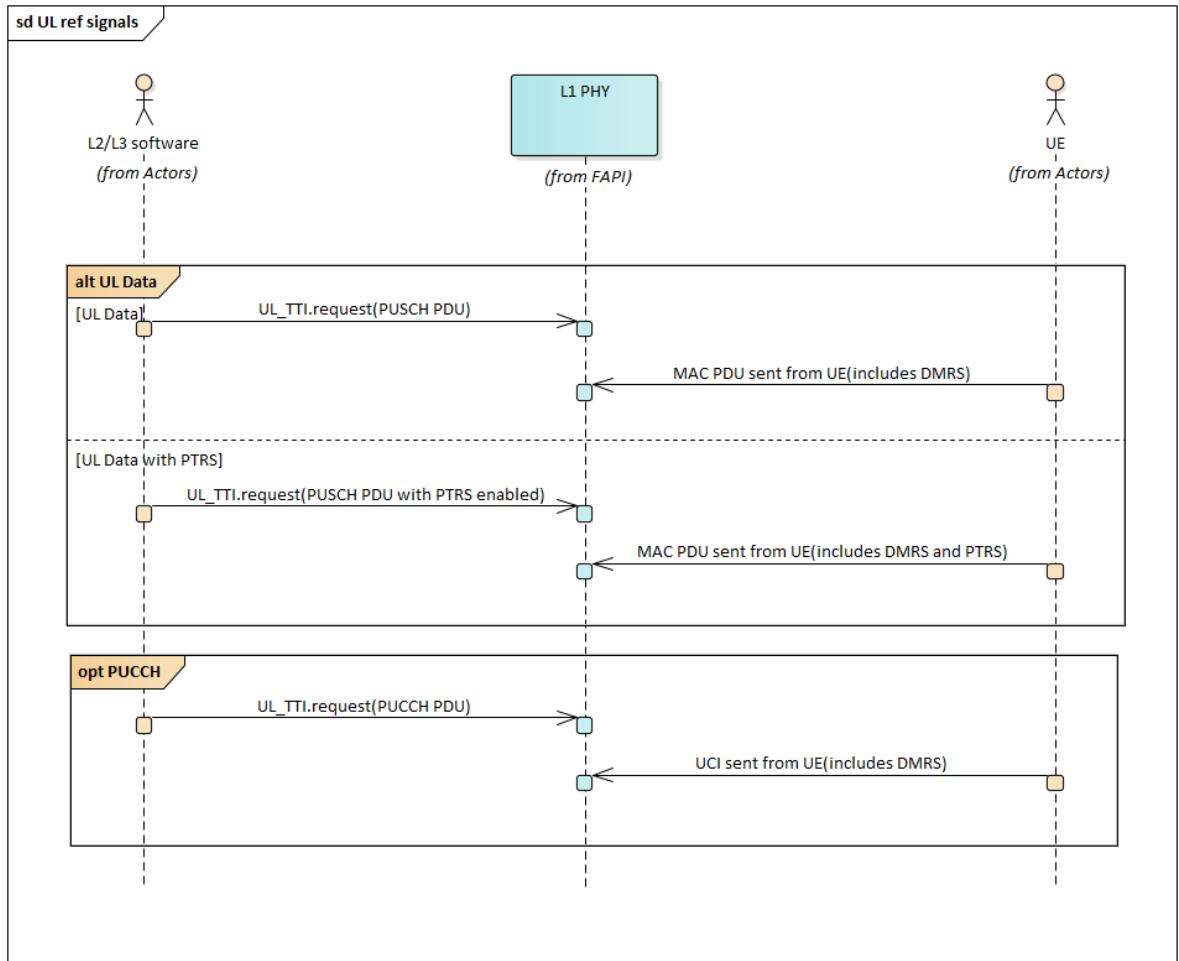


Figure 2-44 Uplink reference signals

2.2.6 Spatial Multiplexing

Spatial multiplexing of channels and layers can be achieved either via precoding and beamforming tables configurable to PHY, or via dynamic precoder computation by PHY.

The spatial multiplexing parameters are defined to be flexible and support varying levels of sophistication.

2.2.6.1 Precoding and Beamforming Based on Semistatic Tables

For semi-static spatial multiplexing, the precoding and beamforming is defined in tables loaded into the PHY at configuration time. In slot messages a precoding index and/or beamforming index is included in each PDU.

Precoding and digital beamforming can be viewed as a cascade of two matrix multiplications PM and DB as depicted in Figure 2-45. For precoding operation an index to a pre-stored precoding matrix PM(idx) is specified in the message. Digital beamforming is represented by a matrix DB. For efficient messaging columns of DB are picked from a pre-stored digital beamforming table (DBT) as illustrated in Figure 2-46. It suffices to specify the index of the beam to be applied for each input port in the message. Baseband ports shown in Figure 2-45 are logical entities that may be further processed by a digital frontend (DFE) unit. Within the context of 5GNR a



baseband port typically corresponds to component carrier that is handled by a specific RF path (i.e. TXRU).

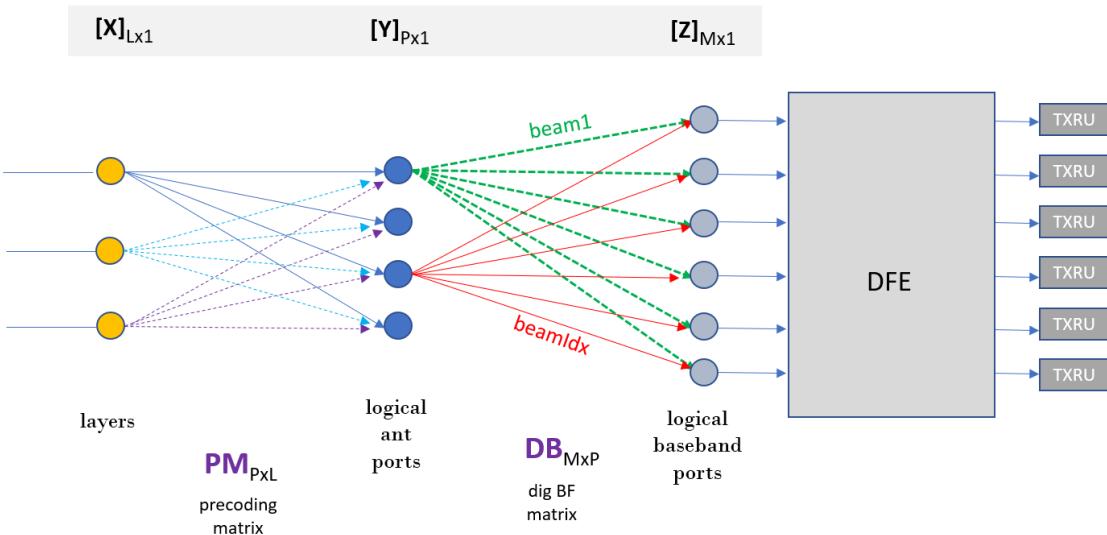


Figure 2-45 Precoding and digital beamforming flow

X : precoder input vector of size ($L \times 1$)
 Y : precoder output vector of size ($P \times 1$)
 Z : digital beamformer output vector of size ($M \times 1$)
 PM : Precoding matrix of size ($P \times L$)
 DB : digital beam forming matrix of size ($M \times P$)

$$Y = \text{PM} \times X$$

$$Z = \text{DB} \times Y$$

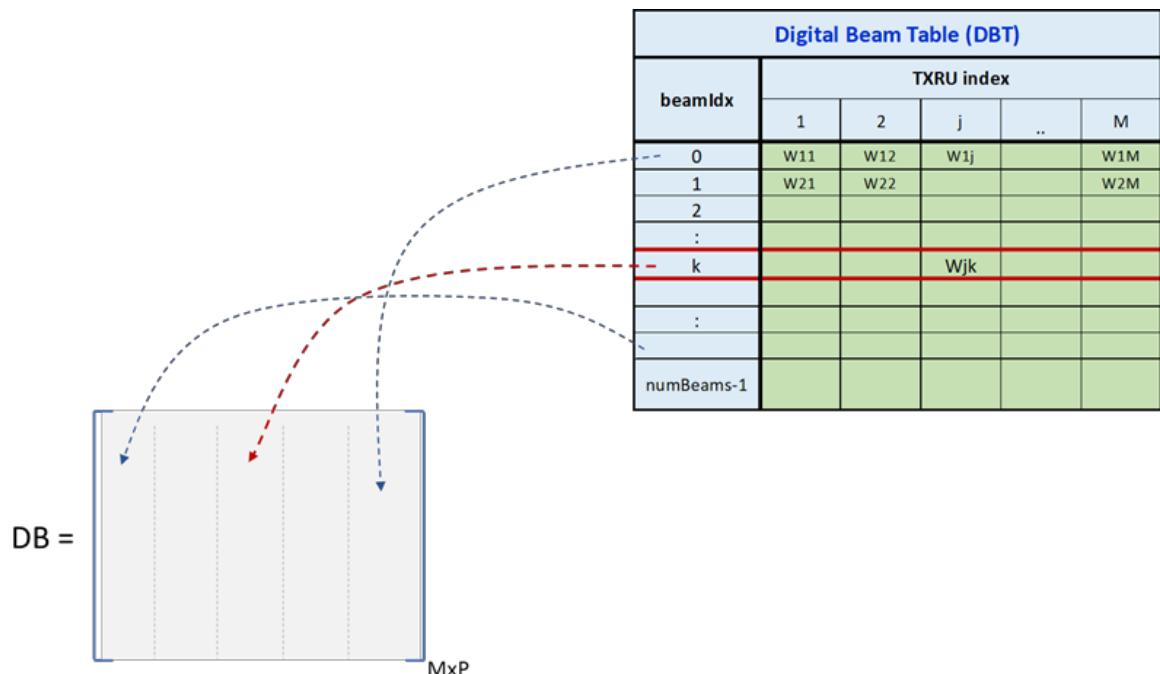


Figure 2-46 Overview of digital beam table (DBT)



2.2.6.2 Precoding based on PHY-determined weights

Alternatively, downlink precoders/uplink combiners can be determined in PHY. To help PHY in such spatial multiplexing, PHY collects SRS samples which can be used to form uplink codebook-based combiners. Under channel reciprocity conditions, SRS samples can also be used to form downlink precoders and non-codebook uplink combiners.

To assist L2 select UEs and layers for spatial multiplexing in a slot, L1 reports a frequency-subsampled set of channel estimates to L2, in the form of e.g. the channel matrix H for the UE, or an SVD decomposition thereof, as described in Table 2-3.

MU-MIMO direction	5G NR SRS type	Type of channel abstraction (in SRS.indication)	Layer signalling for PHY precoder/combiner determination
DL	Antenna-switch	SVD representation: $H = U S V$ (under channel reciprocity assumptions) U: $Rx_{UE} \times Rx_{UE}$ left-singular vector matrix S: $Rx_{UE} \times Rx_{UE}$ diagonal matrix of singular values V: $Rx_{UE} \times Tx_{gNB}$ matrix of right-singular vectors for Rx_{UE} singular values.	Selection, in DL_TTI.request of UEs' Downlink (e.g. PDSCH), PDU to schedule, and of singular values in S diagonal matrix, for each scheduled UE
UL	Codebook-based	H	Selection of UEs to schedule, and of UL-TPMI, for each scheduled UE.
UL	Non-codebook-based	H (under weak channel reciprocity assumptions)	Selection of UEs to schedule, and of Set of SRS tx ports, for each scheduled UE.

Table 2-3 SRS-based reporting & layer selection, per direction & usage type

Digital precoding (for downlink) is depicted in Figure 2-47, and digital combining (for uplink) is depicted in Figure 2-48. Downlink precoder determination and uplink combiner determination are based on a signalling of UEs and layers (see Table 2-3) for joint spatial multiplexing over the same set of RBs and symbols.

The set of all the RBs and symbols in a slot over which the same set of UE channels and layers is scheduled is denoted as a "MU-MIMO Group", as illustrated in Figure 2-49.

Baseband ports shown in Figure 2-47 and Figure 2-48 are logical entities that may be additionally processed by a digital frontend (DFE) unit. Within the context of 5GNR a baseband port typically corresponds to component carrier that is handled by a specific RF path (i.e. TXRU).

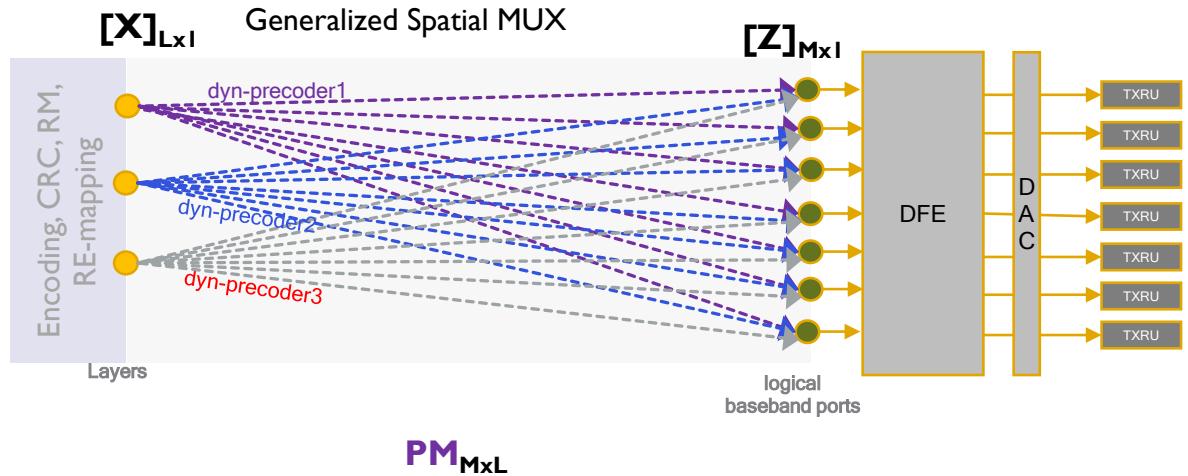


Figure 2-47 Dynamic Precoding flow (Downlink)

X : precoder input vector of size ($L \times 1$)
 Z : digital beamformer output vector of size ($M \times 1$)
 PM : Dynamic Precoding matrix of size ($P \times M$)

$$Z = PM \times X$$

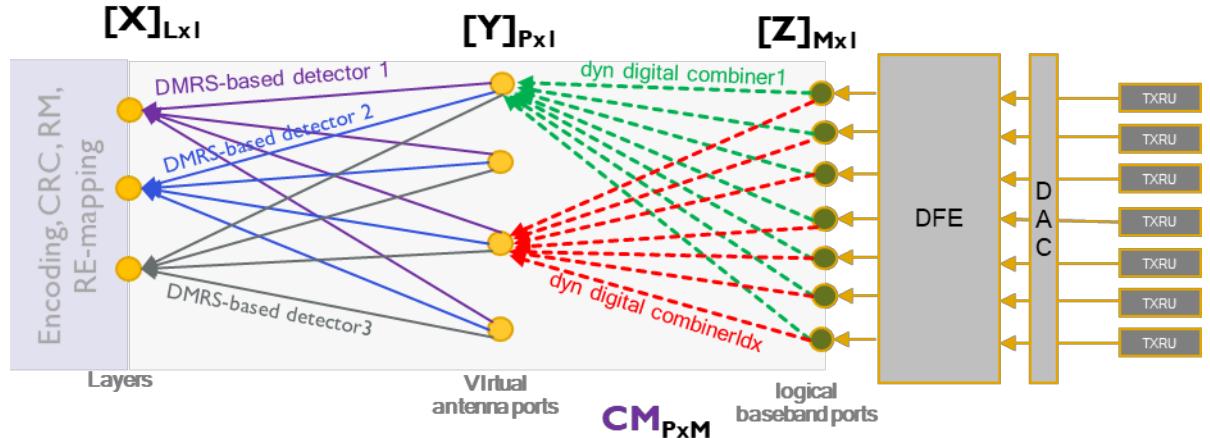


Figure 2-48 Dynamic Combiner flow (Uplink)

X : Layer output vector of size ($L \times 1$)
 Y : Spatial stream digital combiner output vector of size ($P \times 1$)
 Z : Digital combiner input vector of size ($M \times 1$)
 CM : Dynamic Combining matrix of size ($P \times M$)

$$Y = CM \times Z$$

$X = \text{detection}(Y)$ – detection is based on DM-RS detection, not the outcome of combining.

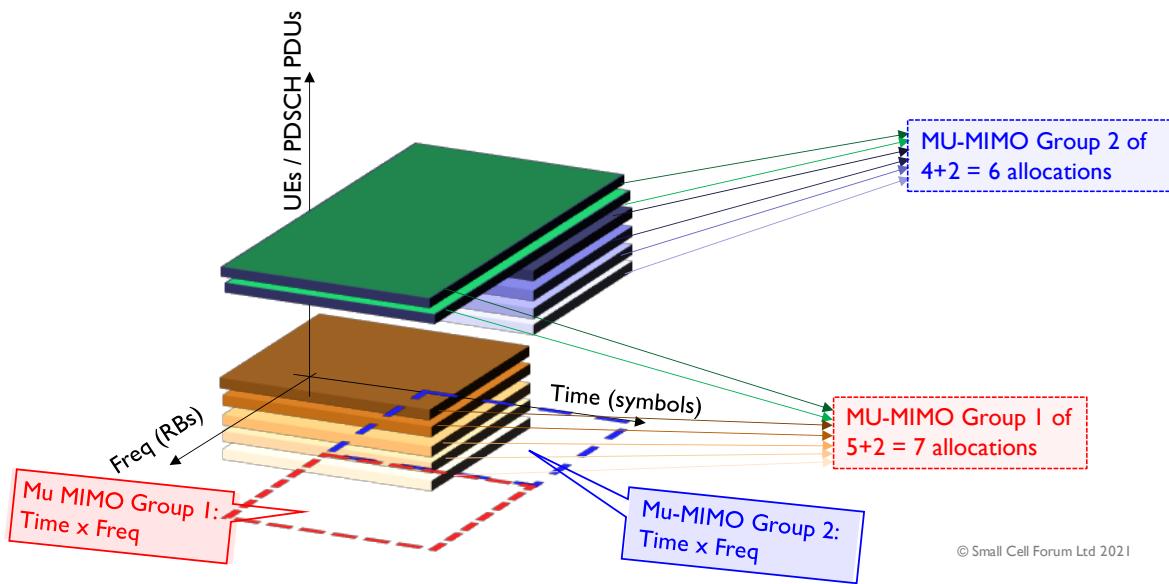


Figure 2-49 MU-MIMO Groups

An example call flow for MU-MIMO scheduling of two Ues is illustrated in Figure 2-50:

1. MAC configures PHY to expect Antenna Switch SRS Resources for each UE
 - o In this example, the SRS arrive in different slots, but they could have arrived in the same slot, depending on higher-layer configuration and triggering.
2. PHY generates SVD reports ($H = U S V$) for each of the Ues.
3. MAC schedules the two Ues, with respective layers picked from the per-PRG singular value matrix S from step 2, for each UE).
4. PHY computes (RE-level) joint precoders, corresponding to the MAC UE and layer selection step 3.
 - o MU-MIMO Group: assists in joint precoder computation
 - o PDSCH PDUs: channel profiles, for each UE

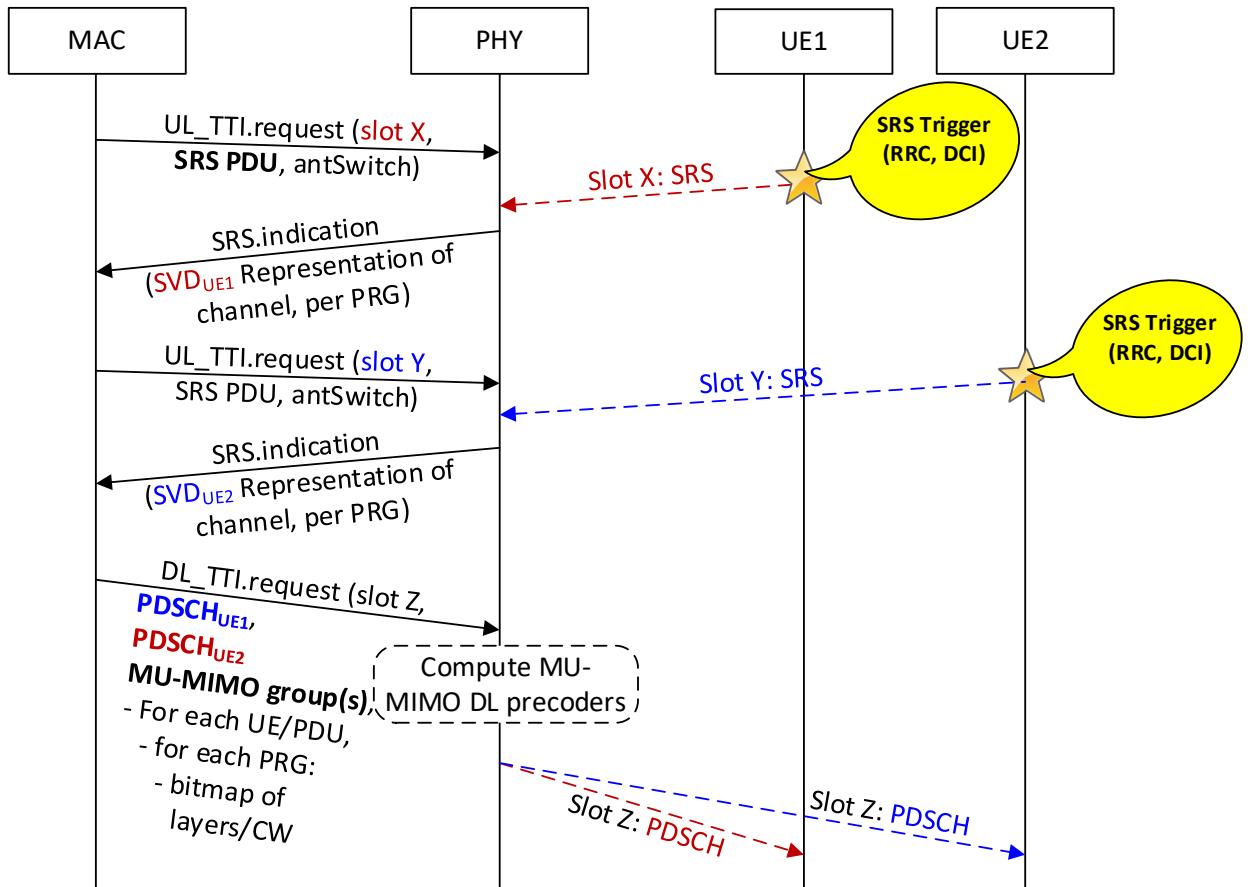


Figure 2-50 Example Call flow for MU-MIMO DL Scheduling

2.2.6.3 Downlink Spatial Stream Limits

In practice, the number of parallel precoders that can be supported by a PHY is limited. Based on PHY capability, MAC assigns DL spatial stream indices between 0 and [(max# downlink streams)-1], so that each RE and symbol output from a precoder is unambiguously associated with a unique DL stream index. The maximum number of DL streams is subject to L1 capability (e.g. maxNumberDLSpatialStreams, maxNumCarriersBWLayersProductDL, maxNumCarriersBWAntennasProductDL).

Downlink spatial streams may – for instance – correspond to downlink data flows, as indexed by “c_eAxC_ID” in section 4.5.4.1 of [26].

To this effect, MAC associates DL stream indices per PRG as follows:

- PDCCH PDU: one stream index per DCI.
- PDSCH PDU: one stream index per layer.
- CSI-RS PDU: one stream index per cdm port in a cdm group.
- SSB PDU: one stream index.

Where L1 capability (e.g. dlSpatialStreamChannelPriority) allows the same index to be associated with the same RE and symbol in multiple PDUs for a slot, PHY drops all but the RE and symbol for the highest priority channel PDU.



2.2.6.4 Uplink Spatial Stream Limits

In practice, the number of parallel combiners that can be supported by a PHY is limited. Based on PHY capability, MAC assigns UL spatial stream indices between 0 and [(max # uplink streams)-1], so that:

- each RE and symbol output from a combiner is unambiguously associated with a at least one UL channel PDU.
- when ul combiners are signaled as UL logical ports, the number of spatial steams associated with a channel is equal to the number of logical ports for the channel.
- the number of spatial streams associated with an UL channel is at least as large as the number of layers for the UL channel.

The maximum number of UL streams is subject to L1 capability (e.g. maxNumberUISpatialStreams, maxNumCarriersBWLayersProductUL, maxNumCarriersBWAntennasProductUL).

Uplink spatial streams may – for instance – correspond to uplink data flows, as indexed by “c_eAxC_ID” in section 4.5.4.1 of [26].

To this effect, MAC associates UL stream indices per PDU as follows:

- PUCCH PDU: (at least) one stream index.
- PUSCH PDU: (at least) one stream index per layer.
- PRACH PDU: as many stream indices as logical channels (at least one).
- SRS PDU: as many stream indices as logical channels, except if SRS PDUs are associated with RAW ports. In the latter case, MAC does not assign UL spatial streams to the SRS PDU

2.2.7 Error sequences

The error sequences used for each slot procedure are shown in Figure 2-51 to Figure 2-54. In all slot procedures errors that are detected by the PHY are reported using the ERROR.indication message. In Section 3, the PHY API message definitions include a list of error codes applicable for each message.

The DL_TTI.request, UL_TTI.request, UL_DCI.request and TX_DATA.request messages include information destined for multiple UEs. An error in information destined for one UE can affect a transmission destined for a different UE. For each message the ERROR.indication sent by the PHY will return the first error it encountered.

If the L2/L3 software receives an ERROR.indication message for DL_TTI.request, UL_TTI.request, UL_DCI.request or TX_DATA.request, it should assume that the UE did not receive data and control sent in this slot. This is similar to the UE experiencing interference on the air-interface and 5G mechanisms, such as, HARQ and ARQ, will enable the L2/L3 software to recover.

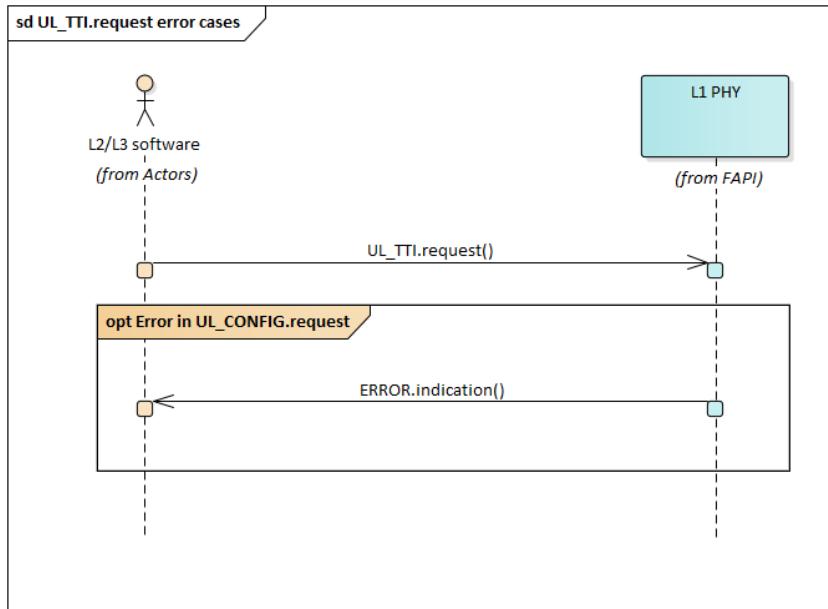


Figure 2-51 `UL_TTI.request` error sequence

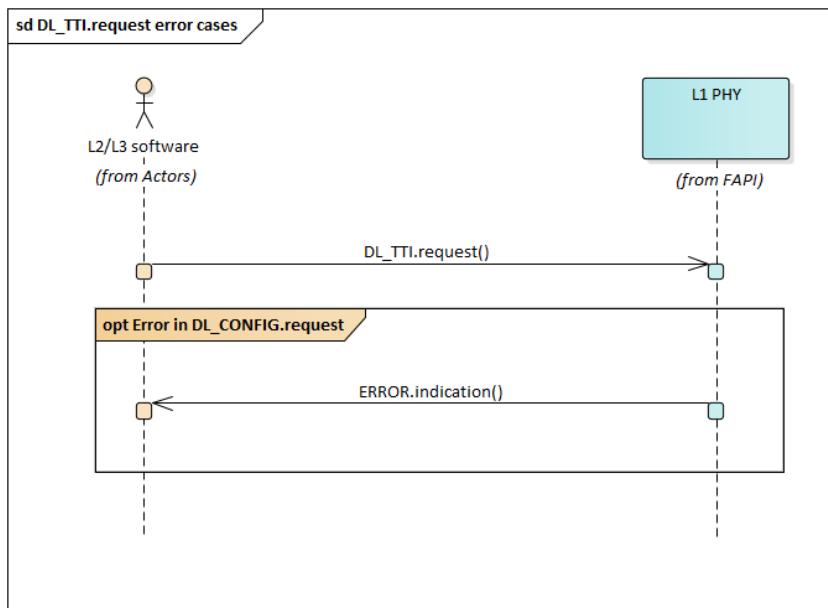


Figure 2-52 `DL_TTI.request` error sequence

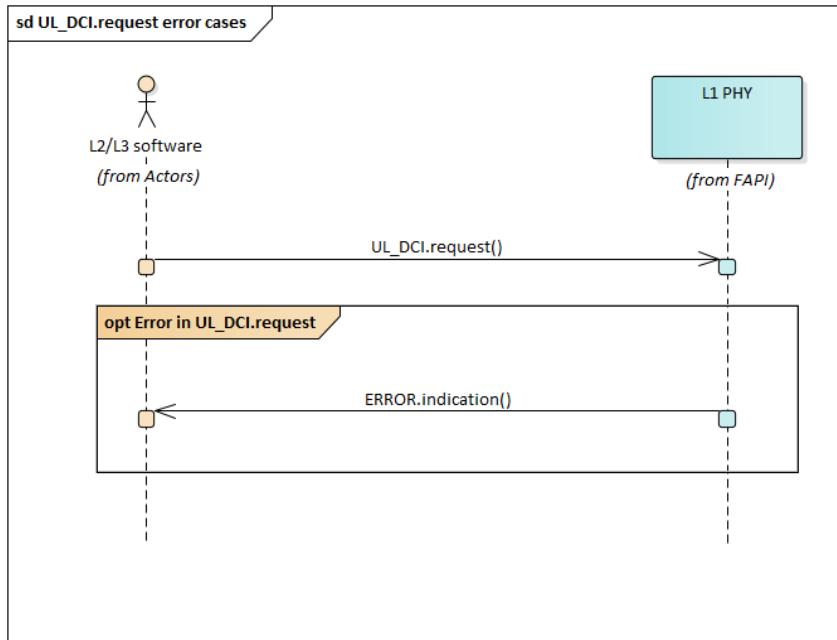


Figure 2-53 `UL_DCI.request` error sequence

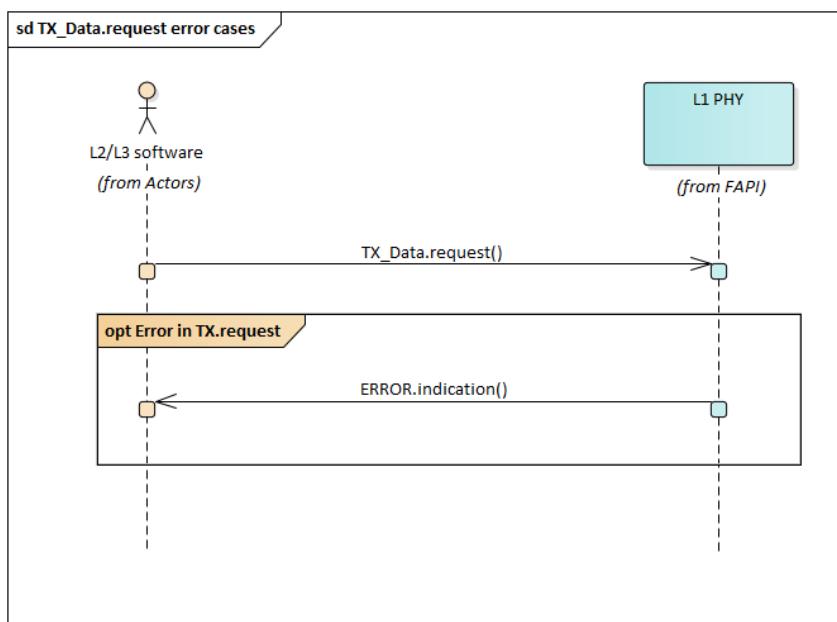


Figure 2-54 `TX_DATA.request` error sequence



3. PHY API Messages

This section provides a description of the PHY API message formats. It defines the PHY API message header, the message bodies and the error codes associated with the PHY API.

3.1 General Message Format

FAPI messages may be defined using the Basic (section 3.1.1) or nFAPI (section 3.1.2) message formats.

3.1.1 Simplified Message Format

The general message format of the PHY API is shown in Table 3-1, where it can be seen that there is a top structure of 1 or more PHY API messages. This permits the optional bundling of messages and the inclusion of vendor specific messages.

Description
PHY API Message Header (see Table 3-2)
PHY API Message1 (see Table 3-3)
PHY API MessageLAST
Vendor-specific message

Table 3-1 General PHY API message format

Type	Description
uint8_t	Number of messages included in PHY API message
uint8_t	An opaque handle (purpose not defined, however, usages may include a PHY ID, Carrier ID or Common Context). Only P5 PARAM and CONFIG messages may be scoped at Common Context, in the current version of the specification.

Table 3-2 PHY API message header

Each PHY API message consists of a generic header followed by a message body, as shown in Table 3-3, and consists of a message type ID, a length field and a message body.

Type	Description
uint16_t	Message type ID
uint32_t	Length of message body (bytes)
Message body	

Table 3-3 General PHY API message structure

The current list of message types is given in Table 3-5. The PHY API messages follow a standard naming convention where:

- All .request messages are sent from the L2/L3 software to the PHY.
- All .response messages are sent from the PHY to the L2/L3 software. These are sent in response to a .request.
- All .indication messages are sent from the PHY to the L2/L3 software. These are sent asynchronously.



The message body is different for each message type; however, each message body obeys the following rules:

- The first field in each response message is an error code. For each message it is indicated which error codes can be returned. A full list of error codes is given in Section 3.3.6.1.

A full description of each message body is given in the remainder of Section 3.

The API mechanism can use either little-endian byte order, or big-endian byte order. The selection of byte ordering is implementation specific. However, the LSB of the bit stream shall be the LSB of the field defined in the API. For example, the location of a 12-bit bitstream within a 16-bit value would be:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msb								bitstream						lsb	

This document assumes that the API messages are transferred using a reliable in-order delivery mechanism.

3.1.2 nFAPI Message Format

The general message format of the PHY API is shown in section 2.3.2 of SCF-225 [10], for transport layers other than defined in section 2.3.1 of SCF-225 [10], with the following changes:

- the API mechanism can use either little-endian byte order, or big-endian byte order, as described in section 3.1.1.
- combined messages, as described in section 2.3.2.2 and defined in section 2.3.2.4 of SCF-225 [10] are not supported.
 - **Note:** The absence of combined messages is addressed by the additional param (Table 3-28) and config (Table 3-57) Delay Management TLVs in section 3.3.
- only P7 messages designated as dedicated in SCF-225 [10] are supported, as documented in sections 3.4.12, 3.4.13, and 3.4.14, for PHYs indicating Delay Management capability.
- a 32-byte value shall precede the nFAPI header from section 2.3.2 of SCF-225 [10], with the following value to indicate the message payloads carried in the nFAPI PDU, as described in Table 3-4:
 - 0: FAPI configuration messages (P5 or P19-C)
 - 1: FAPI slot-based messages (P7 or P19-S)

Description
nFAPI adaptation dword (32-bit)
nFAPI PDU, as in SCF-225

Table 3-4 Adaptation of nFAPI structure



3.1.3 Padding

L1 may indicate support for particular padding mechanisms in *MessageEncodingSupport* TLV. If so supported, the following mechanisms are supported:

1. Legacy padding (*Message Encoding Support* = 0): padding of FAPI fields messages is not specified.
2. No padding (*Message Encoding Support* = 1): FAPI fields and messages are concatenated in their order of appearance, without any padding.
3. Aligned padding (*Message Encoding Support* = 2): FAPI messages following the padding rules below:
 - a. Each scalar parameter is start-aligned to its size (e.g. 16-bit words start on 16-bit boundaries)
 - b. Each parameter that is a scalar array is start-aligned and size-aligned to a 32-bit boundary
 - c. Minimal necessary padding to ensure alignment is added before the parameter.
 - d. Each structure, PDU, message, and non-scalar loop iteration starts and ends on a 32-bit boundary
 - e. Alignment is considered with respect to the start of the FAPI message header; the message headers themselves (or fields thereof) are not subject to padding.

3.2 Message Types

The messages in the 5G PHY API are given in Table 3-5.

Message	Value	Message Body Definition
PARAM.request	0x00	See Section 0
PARAM.response	0x01	See Section 03.3.1.1
CONFIG.request	0x02	See Section 3.3.2.1
CONFIG.response	0x03	See Section 3.3.2.2
START.request	0x04	See Section 3.3.4.1
START.response	See section 2.3.2.7 of SCF-225 [10] Note: 0x108, as of nFAPIv2	See Section 3.3.4.3
STOP.request	0x05	See Section 3.3.5.1
STOP.indication	0x06	See Section 3.3.5.2
ERROR.indication	0x07	See Section 3.3.6.1
RESET.request	0x08	See Section 3.3.8.13.3.8.2
RESET.indication	0x09	See Section 3.3.8.2
CONNECTIVITY.indication	0x0a	See Section 3.3.6.3
RESERVED	0x0b-0x7f	
DL_TTI.request	0x80	See Section 3.4.2
DL_TTI.response	0x8a	See Section 3.4.2b
UL_TTI.request	0x81	See Section 3.4.3
SLOT.indication	0x82	See Section 3.4.1
UL_DCI.request	0x83	See Section 3.4.4
TX_DATA.request	0x84	See Section 3.4.6
RX_Data.indication	0x85	See Section 3.4.7



Message	Value	Message Body Definition
CRC.indication	0x86	See Section 3.4.8
UCI.indication	0x87	See Section 3.4.9
SRS.indication	0x88	See Section 3.4.10
RACH.indication	0x89	See Section 3.4.11
TIMING.indication	0x8b	See Section 3.4.1a Note: 0x8a is taken by DL_TTI.response
RESERVED	0x8c-0xff	
DL Node Sync	See section 2.3.2.7 of SCF-225 [10] Note: 0x180, as of nFAPIv2	See Section 3.4.12
UL Node Sync	See section 2.3.2.7 of SCF-225 [10] Note: 0x181, as of nFAPIv2	See Section 3.4.13
Timing Info	See section 2.3.2.7 of SCF-225 [10] Note: 0x182, as of nFAPIv2	See Section 3.4.14
RESERVED for Vendor Extension Messages	2.3.2.7 of SCF-225 [10] Note: 0x0300-0x03FF, as of nFAPIv2	See Section 2.3.2.10 of SCF-225 [10]

Table 3–5 PHY API message types

3.3 Configuration Messages

The configuration messages are used by the L2/L3 software to control and configure the PHY.

3.3.1 PARAM

The PARAM message exchange was described in section 2.1.1.1.PARAM.request

This message can be sent by the L2/L3 when the PHY is in the IDLE state and, optionally, the CONFIGURED state. If it is sent when the PHY is in the RUNNING state, a MSG_INVALID_STATE error is returned in PARAM.response. If L2/L3 does not signal any TLVs, the message length in the generic header = 0.

The PARAM.request message is given in Table 3–6. From this table it can be seen that PARAM.request contains a list of optional TLVs with L2/L3 signalling.

Field	Type	Description
TLVs	Variable	See Table 3–10.

Table 3–6 PARAM.request message body

3.3.1.1 PARAM.response

The PARAM.response message is given in Table 3–7. From this table it can be seen that PARAM.response contains a list of TLVs providing information about the PHY. When the PHY is in the IDLE state, this information relates to the PHY's overall capability. When the PHY is in the CONFIGURED state this information relates to the current configuration.

The full list of PARAM TLVs is given in Section 3.3.1.3, with the following requirements on the PHY in terms of reporting:



- There is no requirement for every TLV to be reported
- There is no requirement for L1 to provide the TLVs in the order specified in the Tables

Field	Type	Description
Error Code	uint8_t	See Table 3-8
Number of TLVs	uint8_t	Number of TLVs contained in the message body.
TLVs	Variable	See Table 3-11.

Table 3-7 PARAM.response message body

3.3.1.2 PARAM errors

The error codes that can be returned in PARAM.response are given in Table 3-8.

Error code	Description
MSG_OK	Message is OK.
MSG_INVALID_STATE	The PARAM.request was received when the PHY was in the RUNNING state.
MSG_INVALID_PHY_ID	The PHY ID is not defined
MSG_UNINSTANTIATED_PHY	The PHY ID is not instantiated
MSG_INVALID_DFE_Profile	No valid DFE Profile exists for the PHY ID
FEU_NOT_INITIALIZED	DFE or RF chains associated with the PHY are not initialized
DFE PROFILE NOT SELECTED	No DFE profile was selected for associationg DFE and RF chains with the PHY

Table 3-8 Error codes for PARAM.response

3.3.1.3 PARAM TLVs

PARAM and CONFIG TLVs are used in the PARAM and CONFIG message exchanges, respectively. For both the PARAM and CONFIG TLVs the TLV format is given in Table 3-9. Each TLV consists of:

- Tag parameter of 2 bytes
- Length parameter of 2 bytes
- Value parameter

The length of the Value parameter ensures the complete TLV is a multiple of 4-bytes (32-bits).

The TLVs for the PARAM message exchange are given in this Section 3.3.1.3 and for the CONFIG message exchange in Section 3.3.2.4.

Type	Description
uint16_t	Tag.
uint16_t	Length (in bytes)
variable	Value

Table 3-9 TLV format



To aid additions of future TLVs in later versions of this API, the highest value tag in the Tables below is 0x0162. In addition, the tag value range 0x0100-0x014F only be used for alignment with nFAPI tag value ranges.

PARAM TLVs used in the `PARAM.request` message are given in Table 3-11.

Tag	Field	Type	Description
0x0037	protocolVersion	uint8_t	<p>Indicates if the highest FAPI Protocol version supported by L2/L3</p> <ul style="list-style-type: none">• 0-2: reserved values• 3 = FAPIv3• 4 = FAPIv4• 5 = FAPIv5 <p>5-255: reserved values. Future FAPI release will be signaled from this range.</p>

Table 3-10 PARAM.request TLVs

PARAM TLVs used in the `PARAM.response` message are given in Table 3-11.

Field	Type	Description
Cell Param	struct	See Table 3-13
Carrier Param	struct	See Table 3-14
PDCCH Param	struct	See Table 3-15
PUCCH Param	struct	See Table 3-16
PDSCH Param	struct	See Table 3-17
PRS Param	struct	See Table 3-18
PUSCH Param	struct	See Table 3-19
MsgA-PUSCH Param	struct	See Table 3-20
PRACH Param	struct	See Table 3-21
MsgA-PRACH Param	struct	See Table 3-22
Measurement Param	struct	See Table 3-23
UCI Param	struct	See Table 3-24
Capability Validity	struct	See Table 3-25
PHY Support	struct	See Table 3-26
PHY/DFE Validity Map	struct	See Table 3-27
Delay Management	struct	See Table 3-28
Rel-16 mTRP support	struct	See Table 3-29
Protocol, User Plane and Encapsulation parameters	struct	See Table 3-30
SRS Capabilities	struct	See Table 3-31
Spatial Multiplexing & MIMO Capabilities	struct	See Table 3-32

Table 3-11 PARAM.response TLVs

Tag	Field	Type	Description
This table contains the top-level configuration parameters relating to the cell.			



Tag	Field	Type	Description
0x0001	Release capability	uint16_t	<p>Bitmap indicating which release the PHY supports</p> <p>For each bit:</p> <p>0 = not supported 1 = supported</p> <p>Bits:</p> <p>0 = Release15 1 = Release16</p>
0x0002	PHY state	uint16_t	<p>Indicates the current operational state of the PHY.</p> <p>Value:</p> <p>0 = IDLE 1 = CONFIGURED 2 = RUNNING</p>
0x0003	<i>Skip_blank_DL_CONFIG</i>	uint8_t	<p>Indicates if the PHY requires a <code>DL_TTI.request</code> message every DL slot or if the message can be skipped if there is nothing to schedule</p> <p>Value:</p> <p>0 = not supported 1 = supported</p>
0x0004	<i>Skip_blank_UL_CONFIG</i>	uint8_t	<p>Indicates if the PHY requires a <code>UL_TTI.request</code> message every UL slot or if the message can be skipped if there is nothing to schedule</p> <p>Value:</p> <p>0 = not supported 1 = supported</p>
0x0005	Num Config TLVs To Report	uint16_t	<p>Indicates the number of CONFIG TLVs which will be reported. For each CONFIG TLV it is indicated if they are optional or mandatory in each PHY state</p>
For Num Config TLVs To Report {			
Value	Tag	uint16_t	Tag value of CONFIG TLV, see Table 3-36 for possible values
	Length	uint8_t	Length of value field Value: 1
		uint8_t	<p>Provides information of optional and mandatory status for each TLV</p> <p>Value:</p> <p>0 = Idle state only, optional 1 = Idle state only, mandatory 2 = Idle and Configured states, optional 3 = Idle state mandatory, Configured state optional 4 = Idle, Configured and Running states optional 5 = Idle state mandatory, Configured and Running states optional</p>



Tag	Field	Type	Description																														
	}																																
0x0038	Power Profiles Supported	uint8_t	<p>Supported Power Profiles</p> <p>Bit Location:</p> <ul style="list-style-type: none"> • 0[LSB] = ProfileNR (same as in FAPIv2) – signals offset are based on standard 3GPP NR parameters. • 1 = ProfileSSS (All RS, except possibly PTRS, are offset from SSS) • 2-7: reserved, set to 0 <p>Bit values:</p> <ul style="list-style-type: none"> • 0 = Not supported • 1 = Supported <p>Default: 0000 0001b</p>																														
	numSignals	uint8_t	<p>Number of signals</p> <p>Value: 9 (in FAPIv5)</p>																														
	For signalIndex = 0 to numSignals - 1		<p>[Signal]'s power range with respect to main RS, when supported, each index corresponds to a signal and its reference RS option(s), per the table below.</p> <p>Except for the SSS and PRS signals, the ReferenceRS entries are only valid for ProfileSSS, shall be set to all-zero if ProfileSSS is not supported.</p> <table border="1"> <thead> <tr> <th>Idx</th> <th>Signal</th> <th>RS Index^{Notes:} Reference RS</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>PDCCH-DMRS</td> <td>0: SSS</td> </tr> <tr> <td>1</td> <td>PDCCH-Data</td> <td>0: SSS 1: PDCCH-DMRS</td> </tr> <tr> <td>2</td> <td>PDSCH-DMRS</td> <td>0: SSS</td> </tr> <tr> <td>3</td> <td>PDSCH-Data</td> <td>0: SSS or 1: PDSCH-DMRS</td> </tr> <tr> <td>4</td> <td>PDSCH-PTRS</td> <td>0: SSS or 1: PDSCH-DMRS</td> </tr> <tr> <td>5</td> <td>CSI-RS</td> <td>0: SSS 1: SSS</td> </tr> <tr> <td>6</td> <td>PSS</td> <td>0: SSS 1: SSS</td> </tr> <tr> <td>7</td> <td>SSS</td> <td>0: unscaled SSS 1: SCF-222 v2.0</td> </tr> <tr> <td>8</td> <td>PRS</td> <td>0: SSS</td> </tr> </tbody> </table>	Idx	Signal	RS Index ^{Notes:} Reference RS	0	PDCCH-DMRS	0: SSS	1	PDCCH-Data	0: SSS 1: PDCCH-DMRS	2	PDSCH-DMRS	0: SSS	3	PDSCH-Data	0: SSS or 1: PDSCH-DMRS	4	PDSCH-PTRS	0: SSS or 1: PDSCH-DMRS	5	CSI-RS	0: SSS 1: SSS	6	PSS	0: SSS 1: SSS	7	SSS	0: unscaled SSS 1: SCF-222 v2.0	8	PRS	0: SSS
Idx	Signal	RS Index ^{Notes:} Reference RS																															
0	PDCCH-DMRS	0: SSS																															
1	PDCCH-Data	0: SSS 1: PDCCH-DMRS																															
2	PDSCH-DMRS	0: SSS																															
3	PDSCH-Data	0: SSS or 1: PDSCH-DMRS																															
4	PDSCH-PTRS	0: SSS or 1: PDSCH-DMRS																															
5	CSI-RS	0: SSS 1: SSS																															
6	PSS	0: SSS 1: SSS																															
7	SSS	0: unscaled SSS 1: SCF-222 v2.0																															
8	PRS	0: SSS																															

Table 3-12 Signal and Reference RS

Notes:

1. Power offset w.r.t. RS Index 0 is signaled as a value in dB. Power offset w.r.t. RS Index 1 – if allowed and supported – is signaled as a set of look-up table indices



Tag	Field	Type	Description
			<p>2. e.g. for CSI-RS, the same reference signal (SSS) is indexed twice, the difference being whether the power offset is an explicit value (<i>csiRsPowerOffsetProfileSSS</i>) in dB or index (<i>powerControl/OffsetSSProfileNR</i>) representing one of the values in 3GPP TS 38.214 [5], sec 5.2.2.3.1</p>
	ReferenceRS	uint16	<p>Bit indexes defined per the Reference RS index in the Table 3-12, all other bits shall be set to 0.</p> <p>Valid range:</p> <ul style="list-style-type: none">• 0: not supported• 1: supported. <p>If ProfileSSS is supported, at least one bit is set to 1.</p> <p>If RS index 0 is supported, the supported power offset range signaled via [PowerOffsetMin, PowerOffsetMax] with respect to SSS.</p> <p>If RS index 1 is supported, L1 sets the signal[signalIndex] via internal look-up tables, from 3GPP specifications, as described in section 2.2.4.5.</p> <p>As an exception, for index 7 (SSS), RS index 1 refers to the SSS baseband power configuration from SCF-222 v2.0</p>
	Power Offset Min	int16	<p>Value Range for signalIndex ≠ 7:</p> <ul style="list-style-type: none">• -32767 -> -32767 representing -32.767dB to 32.767dB in 0.001dB steps• -32768: Reference RS 0 not supported. <p>Value Range for signalIndex=7 (SSS):</p> <ul style="list-style-type: none">• -11000 -> 12000 representing -110.0 dB to 120.0 dB in 0.01dB, if ssPbchBlockPowerScaling-Unit-Choice = dB• -11000 -> 0 representing -110.0 dBFS to 0.0 dBFS in 0.01dB, if ssPbchBlockPowerScaling-Unit-Choice = dBFS• -32768: Reference RS 0 not supported (i.e. fallback to FAPIv2 SSS baseband scaling)
	Power Offset Max	int16	Same as PowerOffsetMin, with PowerOffsetMin ≤ PowerOffsetMax
0x0058	ssPbch Block Power Scaling-Unit-Choice	uint8_t	Signals unit PHY assumes for ssPbchBlockPowerScaling:



Tag	Field	Type	Description
			Value: 0 = 'dB' (default from FAPIv3) 1 = 'dBFS'
0x0039	maxNum PDUs In DL_TTI	uint16_t	Maximum number of PDU that L1 can process in <code>DL_TTI.request</code> .
0x003A	maxNum PDUs In UL_TTI	uint16_t	Maximum number of PDU that L1 can process in <code>UL_TTI.request</code> .
0x003B	maxNum PDUs In UL_DCI	uint16_t	Maximum number of PDU that L1 can process in <code>UL_DCI.request</code> .

Table 3-13 Cell and PHY parameters

Tag	Field	Type	Description
This table contains the parameters relating to carrier configuration.			
0x0006	cyclicPrefix	uint8_t	Bitmap indicating support. For each bit: 0= not supported 1 = supported Bits: 0 = Normal 1 = Extended
0x0007	supported Subcarrier Spacings DI	uint8_t	Bitmap indicating support. For each bit: 0= not supported 1 = supported Bits: 0 = 15KHz 1 = 30KHz 2 = 60KHz 3 = 120KHz 4 = 240KHz
0x0059	supported Subcarrier Spacings SSB	uint8_t	Bitmap indicating SSB numerology support. If this capability not signaled, SSB numerology capability is signaled by <code>supportedSubcarrierSpacingsDI</code> : For each bit: 0= not supported 1 = supported Bits: 0 = 15KHz 1 = 30KHz 3 = 120KHz 4 = 240KHz All other bit positions (#2, #6, #7) should be set to 0, in this release
0x0008	supported Bandwidth DI	uint16_t	Bitmap indicating support format. For each bit: 0= not supported



Tag	Field	Type	Description
			1 = supported Bits: 0 = 5MHz 1 = 10MHz 2 = 15MHz 3 = 20MHz 4 = 25MHz 5 = 40MHz 6 = 50MHz 7 = 60MHz 8 = 70MHz 9 = 80MHz 10 = 90MHz 11 = 100MHz 12 = 200MHz 13 = 400MHz
0x0009	supported Subcarrier Spacings UI	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = 15KHz 1 = 30Khz 2 = 60Khz 3 = 120Khz
0x000A	supported Bandwidth UI	uint16_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = 5MHz 1 = 10MHz 2 = 15MHz 3 = 20MHz 4 = 25MHz 5 = 40MHz 6 = 50MHz 7 = 60MHz 8 = 70MHz 9 = 80MHz 10 = 90MHz 11 = 100MHz 12 = 200MHz 13 = 400MHz
0x003C	ssPbch Multiple Carriers In A Band	uint8_t	0 = disabled 1 = enabled
0x003D	multiple Cells SsPbch In A Carrier	uint8_t	Indicates that multiple cells will be supported in a single carrier 0 = disabled



Tag	Field	Type	Description
			1 = enabled
0x005A	meaning Of Carrier Frequency	uint8_t	Indicates the PHY interpretation of dlFrequency/ulFrequency parameters: 0 = Point-A [default] 1 = RF center frequency f0.

Table 3-14 Carrier parameters

Tag	Field	Type	Description
This table contains the configuration parameters relating PDCCH configuration.			
0x000B	cce Mapping Type	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Interleaved 1 = Non-interleaved
0x000C	coreset Outside First 3 Ofdm Syms Of Slot	uint8_t	0 = not supported 1 = supported
0x000D	precoder Granularity Coreset	uint8_t	0 = not supported 1 = supported
0x000E	pdcch MuMimo	uint8_t	0 = not supported 1 = supported
0x000F	pdcch Precoder Cycling	uint8_t	0 = not supported 1 = supported
0x0010	max Pdcchs Per Slot	uint8_t	Value 1->255

Table 3-15 PDCCH parameters

Tag	Field	Type	Description
This table contains the configuration parameters relating PUCCH configuration.			
0x0011	pucch Formats	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Format 0 1 = Format 1 2 = Format 2 3 = Format 3 4 = Format 4
0x0012	max Pucchs Per Slot	uint8_t	Value 1->255
0x003E	pucch Group And Sequence Hopping	uint8_t	Support for signaling mechanism of PUCCH Group & Sequence hopping parameters, per UL BWP ID 0 = dynamic only (explicit parameters in P7) 1 = semi-static only (per linked in P5 to UL BWP ID, with only UL BWP ID signaled in P7 PUCCH PDU) 2 = both dynamic and semi-static



Tag	Field	Type	Description
0x003F	max Num Ul Bwp Ids	uint8_t	Maximum number of UL BWP IDs supported by PHY, for linking with semi-static PUCCH parameters (see tag 0x003E). 0 = corresponds to value 0 in tag 0x003E > 0: maximum number of semi-static UL BWP IDs configurable to PHY (note that this only relates to semi-static signaling of PUCCH parameters)
0x0040	pucch Aggregation	uint8_t	Bitmap indicating support for PUCCH aggregation (as signaled by the multiSlotTxIndicator parameter). For each bit: 0= not supported 1 = supported Bits: 0 = pucch-Repetition over multiple slots of PUCCH format 1 or 3 or 4

Table 3-16 PUCCH parameters

This table contains the configuration parameters relating PDSCH configuration.			
Tag	Field	Type	Description
0x0013	pdsch Mapping Type	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Type A 1 = Type B
0x0014	pdsch Allocation Types	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Type 0 1 = Type 1
0x0015	pdsch Vrb To Prb Mapping	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Non-interleaved 1 = Interleaved
0x0016	pdsch Cbg	uint8_t	0 = not supported 1 = supported with segmentation in L2 (if so, then L1 shall be capable of reporting CRC via DL_TTI.response) 2 = supported, with segmentation in L1 3 = support with either type of segmentation
0x0017	pdsch Dmrs Config Types	uint8_t	Bitmap indicating support format. For each bit: 0= not supported



Tag	Field	Type	Description
			1 = supported Bits: 0 = Type 1 1 = Type 2
0x0018	pdsch Dmrs Max Length	uint8_t	0 = Length 1 (i.e. only single symbol DMRS is supported) 1 = Length 2 (this includes support for double- and single-symbol DMRS)
0x0019	pdsch Dmrs Additional Pos	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: For single-symbol DMRS: 0 = pos0 (1+0) 1 = pos1 (1+1) 2 = pos2 (1+1+1) 3 = pos3 (1+1+1+1) For double-symbol DMRS: 4 = pos0 (1+0) 5 = pos1 (1+1)
0x001A	max Pdschs TBs Per Slot	uint8_t	Value: 1->255
0x001B	max Number Mimo Layers Pdsch	uint8_t	Value: 1->8
0x001C	supported Max Modulation Order DI	uint8_t	Value: 0 = QPSK 1 = 16 QAM 2 = 64 QAM 3 = 256 QAM
0x001D	max MuMimo Users DI	uint8_t	Value: 1 ->255 Note: 1 indicates no DL MU-MIMO support.
0x001E	pdsch Data In Dmrs Symbols	uint8_t	0 = not supported 1 = supported
0x001F	preemption Support	uint8_t	0 = not supported 1 = supported
0x0020	pdsch Non Slot Support	uint8_t	0 = not supported 1 = supported
0x0041	ssb Rate Match	uint8_t	Rate match via SSB configuration or PDU index reference in PDSCH PDU 0 = not supported 1 = supported via SSB configuration reference 2 = supported via SSB PDU index reference 3 = supported via either configuration or PDU index reference
0x0042	supported Rate Match Pattern Type And Method	uint16_t	Bitmap indicating support for rate match pattern types and methods:



Tag	Field	Type	Description
			<p>For each bit position, value is 0: not supported 1: supported</p> <p>Bit positions 0: scs 15 kHz PRB-symbol bitmap 1: scs 30 kHz PRB-symbol bitmap 2: scs NCP 60 kHz PRB-symb bitmap 3: scs ECP 60 kHz PRB-symb bitmap 4: scs 120 kHz PRB-symbol bitmap 5: by reference in PDSCH PDU, to P5-configured PRB-symbol configuration 6: by value, direct signaling of PRB-symbol patterns in PDSCH PDU. Shall be 0 in this release. 7: CORESET-based @ scs 15 kHz 8: CORESET-based @ scs 30 kHz 9: CORESET-based @ scs NCP 60 kHz 10: CORESET-based @ scs ECP 60 kHz 11: CORESET-based @ scs 120 kHz 12: by reference in PDSCH PDU, to PRB-symbol patterns signaled at the <code>DL_TTI.request_top_level</code> 13: by reference in PDSCH PDU, to P5-configured CORESET-based configuration 14: by reference in PDSCH PDU, to CORESET-based patterns signaled at the <code>DL_TTI.request_top_level</code> 15: <reserved> set to 0 in this release.</p> <p>No other method is supported in this release.</p>
0x0043	pdcch Rate Match	uint8_t	Refers to ability to signal PDCCH rate match resources in PDSCH PDU. 0 = not supported 1 = supported
0x0044	num Of Rate Match Pattern LTECrs Per Slot	uint8_t	0: no support for LTE-CRS rate matching > 0: max # patterns supported per slot
0x0045	num Of Rate Match Pattern LTECrs In Phy	uint8_t	0: no support for LTE-CRS rate matching > 0: max # patterns supported in this PHY
0x005B	IteCrs Rate Match Mbsfn Derivation	uint8_t	PHY capability to determine MBSFN slots. For each bit position: <ul style="list-style-type: none">- 0: method not supported- 1: method supported Bit positions: <ul style="list-style-type: none">- [0]: slot classified as MBSFN or non-MBSFN based on MBSFN pattern signaling in MbsfnSubframeConfigList



Tag	Field	Type	Description
			<ul style="list-style-type: none">- [1]: slot classified as MBSFN or non-MBSFN based on the P7 IteCrsMbsfnPattern
0x005C	supported LTECrs Rate Match Method	uint16_t	<p>Bitmap indicating support for LTE-CRS rate match pattern methods:</p> <p>For each bit position, value is</p> <p>0: not supported 1: supported</p> <p>Bit positions</p> <p>0: by reference in PDSCH PDU, to P5-configured LTE-CRS patterns 1: by reference in PDSCH PDU, to LTE-CRS patterns signaled at the <code>DL_TTI.request</code> top level 2-15: <reserved> set to 0 in this release.</p> <p>No other method is supported in this release.</p>
0x0046	csiRs Rate Match	uint8_t	Refers to ability to signal CSI-RS rate match PDU indices in PDSCH PDU. 0 = not supported 1 = supported
0x0047	pdsch Trans Type Support	uint8_t	0 = cannot interpret pdschTransType in PDSCH PDU 1 = requires a valid pdschTransType in PDSCH PDU 2 = can interpret PDSCH PDU resource allocation both with and without pdschTransType
0x0048	pdsch Mac Pdu Bit Alignment	uint8_t	Alignment of TLV data L1 expects in <code>TX_DATA.request</code> MAC PDU. 1 = 8 bit 2 = 16-bit 3 = 32-bit (default) 4 = 64-bit 5 = 128-bit 6 = 256-bit
0x005D	max Number Prb Sym Bitmaps Per Slot	uint16_t [5]	<p>maximum number of PRB Symbol Bitmaps that can be referenced (in any PDSCH) per slot, for each numerology.</p> <p>[0]: 15KHz [1] = 30KHz [2] = 60KHz [3] = 120KHz [4] = undefined (set to "not applicable", in this release)</p> <p>Value range for each entry:</p>



Tag	Field	Type	Description
			<ul style="list-style-type: none">• 0-1000: number of different rate match patterns per slot, for the numerology.• 65535: not applicable (default)
0x005E	max Number CsiRs Rate Match Per Slot	uint16_t [5]	<p>maximum number of CSI-RS that can be referenced (in any PDSCH) per slot, for each numerology.</p> <p>[0]: 15KHz [1]: 30KHz [2]: 60KHz [3]: 120KHz [4]: undefined (set to "not applicable", in this release)</p> <p>Value range for each entry:</p> <ul style="list-style-type: none">• 0-1000: number of different rate match patterns per slot, for the numerology. <p>65535: not applicable (default)</p>
0x0155	max Number Prs Puncturing Per Slot	uint16_t [5]	<p>maximum number of PRS that can be referenced (in any PDSCH) per slot, for each numerology.</p> <p>[0]: 15KHz [1]: 30KHz [2]: 60KHz [3]: 120KHz [4]: undefined (set to "not applicable", in this release)</p> <p>Value range for each entry:</p> <ul style="list-style-type: none">• 0-1000: number of different rate puncturing patterns per slot, for the numerology. <p>65535: not applicable (default)</p>
0x005F	max Num Ssb Per Pdsch Slot	uint8_t	<p>Maximum number of SS/PBCH blocks that can be transmitted in a PDSCH slot.</p> <p>Value:</p> <ul style="list-style-type: none">• 1 – 16;• 255: not applicable (default)
0x0060	universal Rate Match	uint8_t	<p>Signals whether L1 expects all rate match mechanism to apply universally to all PDSCH allocations, that is:</p> <ul style="list-style-type: none">• If time/frequency resources for one PDSCH PDU A overlap with a time/frequency resource X that is unavailable to another PDSCH PDU B, then resource X shall also be unavailable to PDSCH PDU A• MAC ensures that rate match patterns are associated with PDSCH PDUs according to this universality principle.



Tag	Field	Type	Description
			<ul style="list-style-type: none"> • MAC signals all LTE_CRS, PrbSymRM and CoresetRM patterns applicable to a slot, <code>DL_TTI.request</code> for the slot. <p>Value: 0: universal rate match not required. 1: universal rate match required. 2: universal rate match and puncturing required.</p>
0x0061	require Pdsch Signaling Associations	uint8_t	<p>Signals whether L1 expects L2 to signal in a PDSCH PDU all rate match patterns and signals relevant to a particular PDSCH PDU.</p> <p>Note: this is particularly relevant when some pattern or structure X (e.g. SSB) is fully obscured by another pattern or structure Y (e.g. PrbSymb). When required by L1, L2 shall include both structures X and Y, when PDSCH PDU allocation is mutually exclusive with at least one RE and symbol of X.</p> <p>Value: 0: required 1: not required.</p>

Table 3-17 PDSCH parameters

Tag	Field	Type	Description
This table contains the configuration parameters relating PRS configuration.			
0x0156	puncturing of PRS symbols	uint8_t	<p>Indicates whether puncturing of PRS symbols is supported (this is needed to allow SSB to puncture PRS)</p> <p>Values:</p> <ul style="list-style-type: none"> • 0: not supported • 1: supported
0x01576	PRS PDUs per slot	uint16_t	<p>The absolute number of PRS PDUs per slot.</p> <p>Value: 0 ... 65534</p>
0x0158	Proportion of PRS RBs per slot	uint8_t	<p>Maximum number of PRS RBs per slot.</p> <p>Value: 0 ... 100</p>

Table 3-18 PRS parameters

Tag	Field	Type	Description
This table contains the configuration parameters relating PUSCH configuration.			
0x0021	uci Mux Ulsch In Pusch	uint8_t	<p>0 = not supported 1 = supported</p>
0x0022	uci Only Pusch	uint8_t	0 = not supported



Tag	Field	Type	Description
			1 = supported 0 = not supported
0x0023	pusch Frequency Hopping	uint8_t	0 = not supported 1 = supported
0x0024	pusch Dmrs Config Types	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Type 1 1 = Type 2
0x0025	pusch Dmrs Max Len	uint8_t	0 = Length 1 (i.e. only single symbol DMRS is supported) 1 = Length 2 (this includes support for double- and single-symbol DMRS)
0x0026	pusch Dmrs Additional Pos	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: For single-symbol DMRS: 0 = pos0 (1+0) 1 = pos1 (1+1) 2 = pos2 (1+1+1) 3 = pos3 (1+1+1+1) For double-symbol DMRS: 4 = pos0 (1+0) 5 = pos1 (1+1)
0x0027	pusch Cbg	uint8_t	0 = not supported 1 = supported
0x0028	pusch Mapping Type	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Type A 1 = Type B
0x0029	pusch Allocation Types	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Type 0 1 = Type 1
0x002A	pusch Vrb To Prb Mapping	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Non-interleaved



Tag	Field	Type	Description
			1 = Interleaved In this FAPI release, the puschVrbToPrbMapping[1] entry shall be set to 0, as interleaved VRB-PRB mapping is not supported in 3GPP NR.
0x002B	pusch Max Ptrs Ports	uint8_t	Value = 0,1,2
0x002C	max Puschs TBs Per Slot	uint8_t	Value 1->255
0x002D	max Number Mimo Layers Non Cb Pusch	uint8_t	Value: 0 -> 4 L1 should ensure that maxNumberMimoLayers NonCbPusch + maxNumberMimoLayers CbPusch > 0
0x0049	max Number Mimo Layers Cb Pusch	uint8_t	Value: 0 -> 4 L1 should ensure that maxNumberMimoLayers NonCbPusch + maxNumberMimoLayers CbPusch > 0
0x002E	supported Max Modulation Order UI	uint8_t	Value: 0 = QPSK 1 = 16 QAM 2 = 64 QAM 3 = 256 QAM
0x002F	max MuMimo Users UI	uint8_t	Value: 1 ->255 Note: 1 indicates no UL MU-MIMO support.
0x0030	dfts Ofdm Support	uint8_t	0 = not supported 1 = supported, with pi/2-PBSK support 2 = supported, without pi/2-BPSK support
0x0031	pusch Aggregation Factor	uint8_t	Bitmap indicating support for different aggregation factors. For each bit: 0= not supported 1 = supported Bits: 0 = aggregation factor 1 1 = aggregation factor 2 2 = aggregation factor 4 3 = aggregation factor 8
0x004A	pusch Lbrm Support	uint8_t	0 = not supported 1 = supported
0x004B	pusch Trans Type Support	uint8_t	0 = cannot interpret puschTransType in PUSCH PDU 1 = requires a valid puschTransType in PUSCH PDU 2 = can interpret PUSCH PDU resource allocation both with and without puschTransType
0x004C	pusch Mac Pdu Bit Alignment	uint8_t	Alignment of TLV data L1 uses for the MAC PDU in RX_DATA.indication. 1 = 8 bit (default) 2 = 16-bit 3 = 32-bit



Tag	Field	Type	Description
			4 = 64-bit 5 = 128-bit 6 = 256-bit

Table 3-19 PUSCH parameters

Tag	Field	Type	Description
This table contains the configuration parameters relating to MsgA-PUSCH configuration.			
0x0062	msgA-nrOf pusch Res Config	uint16_t	Maximum number of supported MsgA PUSCH resource configuration index per cell. Value 0-65534 Value 65535 is reserved and shall not be used in this release.
0x0063	msgA-pusch Trans Precoding	uint8_t	Indicating if PHY support transform precoding for MsgA PUSCH transmission Value 0: not supported Value 1: supported
0x0064	msgA-Intra Slot Pusch Freq HopH	uint16_t	Indicating if PHY support intra-slot frequency hopping per PUSCH occasion for MsgA PUSCH payload transmission Value 0: not supported Value 1: supported
0x0065	msgA-max Pusch Fd Occasions	uint8_t	Indicating supported maximum number of FDMed PUSCH occasions for MsgA PUSCH payload transmission Value: 0 = 1 1 = 2 2 = 4 3 = 8
0x0066	msgA-guard Band	uint8_t	Indicating whether PHY support MsgA PUSCH transmission with configured guard band or not Value 0: support MsgA PUSCH with no zero RB guard band Value 1: support MsgA PUSCH with one RB guard band
0x0067	msgA-guard Period	uint8_t	Bitmap Indicating whether PHY support MsgA PUSCH transmission with configured guard period Bit 0 represents zero symbol guard period Bit 1 represents one symbol guard period Bit 2 represents two symbol guard period Bit 3 represents three symbol guard period



Tag	Field	Type	Description
			Bit set to 0 means not supported, set to 1 means supported
0x0068	msgA-pusch Mapping Type	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Type A 1 = Type B
0x0069	msgA-pusch Dmrs Max Len	uint8_t	Value: 0 = Length 1 1 = Length 1 and Length 2
0x006A	msgA-pusch Dmrs Cdm Group	uint8_t	Bitmap indicating support CDM group for MsgA PUSCH Each bit: 0 means not supported, 1 means supported Bit 0 = CDM group 0 Bit 1 = CDM group 1
0x006B	msgA-pusch Dmrs Additional Pos	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: For single-symbol DMRS: Bit 0 = pos0 (1+0) Bit 1 = pos1 (1+1) Bit 2 = pos2 (1+1+1) Bit 3 = pos3 (1+1+1+1) For double-symbol DMRS: Bit 4 = pos0 (1+0) Bit 5 = pos1 (1+1)Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = pos0 (1+0) 1 = pos1 (1+1) 2 = pos2 (1+1+1) 3 = pos3 (1+1+1+1) Note: MsgA PUSCH only support Type 1 DM-RS
0x006C	msgA-supported Max Modulation Order UI	uint8_t	Value: 0 = QPSK 1 = 16 QAM
0x006D	msgA-max Pusch Per Pru	uint8_t	Maximum number of PUSCH decodable per PUSCH occasion (including associated DMRS). Value: 1 -> 255
0x006E	msgA-max Nr of Cdm Group	uint8_t	Number of MsgA PUSCH DM-RS CDM group supported Value: 0 -> one CDM group



Tag	Field	Type	Description
			Value: 1 -> two CDM group
0x006F	msgA-max Nr of Dmrs Port	uint8_t	Number of PUSCH DM-RS ports supported per CDM group Value: 0 -> first DM-RS port per CDM group Value: 1 -> first two DM-RS ports per CDM group Value: 2 -> four DM-RS ports per CDM group
0x0070	msgA-Prach-To-Pusch Mapping Scheme	uint8_t	Indicate which scheme PHY supported to receive MsgA PUSCH configuration information. Value: 1 -> semi-static configuration via P5 Value: 2 -> dynamic configuration via P7 Value: 3 -> both schemes supported

Table 3-20 MsgA-PUSCH parameters

Tag	Field	Type	Description
This table contains the parameters relating PRACH configuration			
0x0032	prach Long Formats	uint8_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Format 0 1 = Format 1 2 = Format 2 3 = Format 3
0x0033	prach Short Formats	uint16_t	Bitmap indicating support format. For each bit: 0= not supported 1 = supported Bits: 0 = Format A1 1 = Format A2 2 = Format A3 3 = Format B1 4 = Format B2 5 = Format B3 6 = Format B4 7 = Format C0 8 = Format C2
0x0159	prach Short Format Numerologies	uint16_t	Bitmap indicating numerology support. For each bit: • 0= not supported • 1 = supported Bits positions:



Tag	Field	Type	Description
			[0]: 15KHz [1]: 30KHz [2]: 60KHz [3]: 120KHz
0x0034	prach Restricted Sets	uint8_t	Value: 0 = not supported 1 = Type A and B supported 2 = Type A supported, only 3 = Type B supported, only
0x0035	max Prach Fd Occasions In A Slot	uint8_t	Value: 0 = 1 1 = 2 2 = 4 3 = 8
0x004D	max Num Prach Configurations	Uint16_t	Value 0 = no Prach configurations supported (can only be signaled if PHY supports both dedicated and shared configurations. PHY must signaled must signal non-zero value for at least one of dedicated and shared capability corresponding to this TLV) 1 = no support for multi-PRACH configuration > 1 = maximum number of PRACH configurations supported by PHY
0x004E	prachMultipleCarriers InABand	uint8_t	0 = disabled 1 = enabled
0x0071	prach Frequency Offset Meaning	uint8_t	1 = Offset between Point-A and a PRB overlapping with the lowest RE of the PRACH signal 2 = k1 definition in [3GPP TS 38.211 [2], sec 5.3.2]
0x0072	max Num Td Fd Prach Occasions Per Slot	uint16_t [5]	Maximum number of time-frequency PRACH occasions per PRACH slot. Entries refer to PRACH slot SCS: <ul style="list-style-type: none">[0]: 15 kHz (FR1)[1]: 30 kHz (FR1)[2]: N/A (set to 65535)[3]: 60 kHz (FR2)[4]: 120 kHz (FR2) Value: <ul style="list-style-type: none">1-1024;65535: not applicable (default)
0x0073	number Formats	uint16_t	Number of formats included for this TLV Value = 14 in this release



Tag	Field	Type	Description
	prach Prach Root Processing Rate	uint16_t [n]	<p>n = numberFormats</p> <p>Z-C root processing rate, per RO format type. Word positions:</p> <ul style="list-style-type: none">- [0] – undefined (set to 0)- [1] – Format 0- [2] – Format 1- [3] – Format 2- [4] – Format 3- [5] – Format A1- [6] – Format A2- [7] – Format A3- [8] – Format B1- [9] – Format B2- [10] – Format B3- [11] – Format B4- [12] – Format C0- [13] – Format C1- [>13] Undefined in this release, but may be supported in future releases <p>For the specified format, this capability describes the maximum number of Zadoff-Chu roots that can be processed in a frame, after accounting for receive diversity (e.g. 1 root over N virtual ports is counted as N roots).</p> <p>There is no default value for this capability.</p>
0x0074	number Formats	uint16_t	<p>Number of formats included for this TLV</p> <p>Value = 14 in this release</p>
	prach Occasion QueueSize	uint32_t [n]	<p>n = numberFormats</p> <p>The sum of weighted RACH Occasions (RO) in the queue size shall not exceed the Cumulative Weight Threshold, with the weights and threshold signaled, at the following word positions:</p> <ul style="list-style-type: none">- [0] – Cumulative Weight Threshold- [1] – Weight for Format 0- [2] – Weight for Format 1- [3] – Weight for Format 2- [4] – Weight for Format 3- [5] – Weight for Format A1- [6] – Weight for Format A2- [7] – Weight for Format A3- [8] – Weight for Format B1- [9] – Weight for Format B2- [10] – Weight for Format B3- [11] – Weight for Format B4- [12] – Weight for Format C0- [13] – Weight for Format C1- [>13] Undefined in this release, but may be supported in future releases



Tag	Field	Type	Description
			<p>For the specified format, this capability describes the weight to assign to each RO in the queue, after accounting for receive diversity (e.g. 1 RO over N virtual ports is counted as N ROs). The sum of all weighed ROs is compared against the Cumulative Weight Threshold to determine the queue size.</p> <p>There is no default value for this capability.</p>

Table 3-21 PRACH parameters

Tag	Field	Type	Description
This table contains the parameters relating to MsgA-PRACH configuration			
0x0075	msgA-max Num Prach Configurations	Uint16_t	<p>Value</p> <p>0 = no support in this PHY</p> <p>> 0 = maximum number of PRACH configurations supported by PHY</p>
0x0076	msgA-prach Long Formats	uint8_t	<p>Bitmap indicating support format.</p> <p>For each bit:</p> <p>0= not supported</p> <p>1 = supported</p> <p>Bits:</p> <p>0 = Format 0</p> <p>1 = Format 1</p> <p>2 = Format 2</p> <p>3 = Format 3</p>
0x0077	msgA-prach Short Formats	uint16_t	<p>Bitmap indicating support format.</p> <p>For each bit:</p> <p>0= not supported</p> <p>1 = supported</p> <p>Bits:</p> <p>0 = Format A1</p> <p>1 = Format A2</p> <p>2 = Format A3</p> <p>3 = Format B1</p> <p>4 = Format B2</p> <p>5 = Format B3</p> <p>6 = Format B4</p> <p>7 = Format C0</p> <p>8 = Format C2</p>
0x0078	msgA-prach Restricted Sets	uint8_t	<p>Value:</p> <p>0 = not supported</p> <p>1 = Type A and B supported</p> <p>2 = Type A supported, only</p> <p>3 = Type B supported, only</p>



Tag	Field	Type	Description
0x0079	msgA-max Prach Fd Occasions In A Slot	uint8_t	Value: 0 = 1 1 = 2 2 = 4 3 = 8
0x007A	msgA-prach Multiple Carriers In A Band	uint8_t	0 = disabled 1 = enabled

Table 3-22 MsgA-PRACH parameters



Tag	Field	Type	Description
This table contains the parameters relating Measurement reporting			
0x0036	RssiMeasurementSupport	uint8_t	<p>Indicates if RSSI measurement is supported and what units can be used for reporting.</p> <p>The RSSI reported will be total received power summed across all antennas. RSSI includes all noise and interference contributions.</p> <p>Bitmap indicating support. For each bit: 0 = not supported 1 = supported Bits: 0 = RSSI reporting in dBm 1 = RSSI reporting in dBFS</p>



Tag	Field	Type	Description
0x007B	Rsrp Measurement Support	uint8_t	<p>Indicates if RSRP measurement is supported.</p> <p>The RSRP reported will be <i>the linear average over the power contributions (in [W]) of the resource elements of the antenna port(s) that carry reference signals configured for RSRP measurements within the considered measurement frequency bandwidth in the configured reference signal occasions</i>. [3GPP TS 38.215 [18], section 5.1.2]</p> <p>For dBm unit measurements:</p> <ul style="list-style-type: none">• <i>For frequency range 1, the reference point for the RSRP shall be the antenna connector of the FEU.</i> [3GPP TS 38.215 [19], section 5.1.2]• <i>For frequency range 2, RSRP shall be measured based on the combined signal from antenna elements corresponding to a given receiver branch.</i> [3GPP TS 38.215 [19], section 5.1.2] <p>Bitmap indicating support. For each bit: 0 = not supported 1 = supported Bits: 0 = RSRP reporting in dBm 1 = RSRP reporting in dBFS</p>



Tag	Field	Type	Description
0x015A	preamble Power Unit	uint8_t	<p>Indicates the unit for the preamblePwr in RACH.indication.</p> <p>For dBm unit measurements:</p> <ul style="list-style-type: none">- For frequency range 1, the reference point for the measurement shall be the antenna connector of the FEU.- For frequency range 2, the report shall be measured based on the combined signal from antenna elements corresponding to a given receiver branch. <p>Similar to the regular RSRP report, the preamblePwr report will be <i>the linear average over the power contributions (in [W]) of the resource elements of the antenna port(s) that carry reference signals configured for RSRP measurements within the considered measurement frequency bandwidth in the configured reference signal occasions</i>. [3GPP TS 38.215 [18], section 5.1.2]</p> <p>Unlike the regular RSRP report, the preamblePwr report represents the preamble power observation averaged over all PRACH REs, at the PRACH numerology</p> <p>Value:</p> <ul style="list-style-type: none">- 0 = preamblePwr reporting in dBm- 1 = preamblePwr reporting in dBFS- 2 = preamblePwr reporting in either dBm or dBFS



Tag	Field	Type	Description
0x015B	srsRsrpPowerUnit	uint8_t	<p>Indicates the unit for the srsRsrp in SRS.indication.</p> <p>For dBm unit measurements:</p> <ul style="list-style-type: none">- For frequency range 1, the reference point for the measurement shall be the antenna connector of the FEU.- For frequency range 2, the report shall be measured based on the combined signal from antenna elements corresponding to a given receiver branch. <p>Value:</p> <ul style="list-style-type: none">- 0 = preamblePwr reporting in dBm- 1 = preamblePwr reporting in dBFS- 2 = preamblePwr reporting in either dBm or dBFS

Table 3–23 Measurement parameters

This table contains the parameters relating to UCI reporting			
Tag	Field	Type	Description
0x004F	maxNumUciMaps	uint16	<p>Indicates the maximum number of UCI Part1 to Part 2 maps supported in the UCI Configuration Table 3–51.</p> <p>Value: 0 -> 65535</p> <p>0 (default): no support for FAPIv3 UCI reporting (i.e.no support for CSI part2 reporting)</p>
0x007C	uciReportFormatPucchFormat234	uint8	<p>Indicates how UCI for PUCCH Formats 2, 3, 4 is reported</p> <p>For each both position: 0 = PHY supports; 1 = PHY does not support</p> <ul style="list-style-type: none">• Bit #0: UCI report includes separate parameters to for SR, HARQ, CSI Part 1 and CSI Part2 (i.e. PHY will only set bits 0-3 in pduBitmap of Table 3-98)• Bit #1: UCI report includes parameters for uciPart 1 and uciPart2 (i.e. PHY will only set bits 2-3 in pduBitmap of Table 3-98) <p>Note: In this release exactly one of bit positions 0 or 1 is set.</p>

Table 3–24 UCI parameters



This table describes the validity scopes of capability TLVs in Table 3-11			
Tag	Field	Type	Description
0x0050	numCapabilities	uint16	<p>Indicates the number of capabilities for which validity scope is declared. Capabilities are listed by tag. Each tag appears at most once.</p> <p>Value: 0 -> 65535</p>
For numCapabilites			
	CapabilityTag	uint16	A tag for any TLV in Table 3-11, except for tag 0x0050. Tag 0x0050 shall be Valid ANYTIME
	ValidityScope	uint8	<p>0: Valid ANYTIME 1: Valid after PHY Profile Selection 2: Valid after PHY and DFE Profile Selection</p> <p>Default: Tag 0x0050 shall be 0, regardless whether it is listed or not. All other tags default to 2, if not listed.</p>

Table 3-25 Capability validity scope



Tag	Field	Type	Description
This table define PHY and PHY Profile support parameters			
0x0051	This tag describes the supported PHY Profiles. This TLV is only supported at Common Context. Terminology: <ol style="list-style-type: none">3. interface BB port: a baseband (BB) port label known to both PHY and FEU components. BB ports may be either in the UL or DL direction.4. local BB port: an index used locally in the PHY to refer to BB ports. If a PHY ID is assigned a set of interface BB ports, part of the PHY Profile Selection step of the PHY Instantiation procedure described in section 2.1.9, the local BB port indices are labelled starting from 0, per direction in increasing number of the BB ports in the same direction: <ol style="list-style-type: none">5. the n-th local DL BB port for the PHY corresponds to the n-th largest interface DL BB port assigned to the PHY.6. the n-th local UL BB port for the PHY corresponds to the n-th largest interface UL BB port assigned to the PHY.		
	num Phy Profiles	uint16	Indicates the number of supportable PHY Profiles. Value: 0 -> PHY Profiles feature not supported (this value is disallowed when FAPI is carried over nFAPI, as specified in SCF-225 [10]) 1 -> 65535: number of supported PHY Profiles
	num DI Bb Ports	uint16	Indicates the number of DL interface BB (baseband) ports. These labels are common to all FEU, per SCF-223 [9], and PHYs. DL interface BB ports are indexed 0 ... (numDIBbPorts-1) Value: 0 -> 65535
	num UI Bb Ports	uint16	Indicates the number of UL interface BB ports. These labels are common to all FEU, per SCF-223 [9], and PHYs. UL interface BB ports are indexed numDIBbPorts ... (numDIBbPorts + numUIBbPorts - 1) Value: 0 -> 65535
	For numPhyProfiles (<i>PHY profiles IDs defined implicitly, as ordered indices of this loop, starting from 1</i>)		
	max Num Phys	Uint8	Indicates the maximum number of PHY IDs supported in this PHY profile
	For maxNumPhys (<i>PHY IDs for this Profile defined implicitly, as ordered indices of this loop, starting from 1</i>)		



Tag	Field	Type	Description
			<p>DL BB Ports: the set of DL interface ports for this PHY ID is a union of non-overlapping ranges of DL BB interface ports:</p> $\{dlPortRangeStart[0], \dots, dlPortRangeStart[0] + dlPortRangeLen[0] - 1\} \cup \{dlPortRangeStart[1], \dots, dlPortRangeStart[1] + dlPortRangeLen[1] - 1\} \cup \dots$ <p>The start/length of the above ranges fall in the range of DL BB interface ports.</p>
	num DI Port Ranges	uint8	nD: Number of DL port ranges for this PHYs
	dl Port Range Start	uint16 [nD]	Start of each range of DL ports.
	dl Port Range Len	uint16 [nD]	Length of each range of DL ports.
	num UL Port Ranges	uint8	nU: Number of UL port ranges for this PHYs
	ul Port Range Start	uint16 [nU]	Start of each range of UL ports.
	ul Port Range Len	uint16 [nU]	Length of each range of UL ports.
0x0052	Time Management	uint8_t	<p>Type of time management mechanism: Values:</p> <ul style="list-style-type: none"> 0 – SFN/SL synchronization 1- Delay Management with Timestamps 2- Delay Management without Timestamps
0x015C	DL_TTI RxWindow for SSB	uint8_t	<p>Lists the allowed anchoring of the RxWindow for SSB PDUs received in DL_TTI.request (if PHY supports Delay Management)</p> <p>Value:</p> <ul style="list-style-type: none"> 1 – SSB Numerology 2 – Highest PDSCH numerology, if lower than the SSB numerology 3 – either.
0x015D	UL_TTI RxWindow for PRACH	uint8_t	<p>Lists the allowed anchoring of the RxWindow for short PRACH PDUs received in UL_TTI.request (if PHY supports Delay Management)</p> <p>Value:</p> <ul style="list-style-type: none"> 1 – short PRACH Numerology 2 – Highest PUSCH numerology, if lower than the short PRACH numerology 3 – either.
0x0053	This TLV is mandatory in this specification version, whenever L2 signals its ProtocolVersion in PARAM.request.		
	phy Fapi Protocol Version	uint8_t	<p>Indicates if the highest FAPI Protocol version supported by PHY</p> <ul style="list-style-type: none"> • 0-2: reserved values • 3 = FAPIv3



Tag	Field	Type	Description
			<ul style="list-style-type: none">• 4 = FAPIv4• 5 = FAPIv5• 6-255: reserved values. Future FAPI release will be signaled from this range.
	phy Fapi Negotiated Protocol Version	uint8_t	<p>Indicates the version FAPI protocol version that PHY uses, for pairing with the MAC protocol version, if any was provisioned. If addressed to PHY-ID #0, the assumption applies to all PHYs under the SoC/PNF control.</p> <ul style="list-style-type: none">• 0 = No MAC Protocol provisioned PHY will use the same version as ProtocolVersion.• 1 = FAPIv1• 2 = FAPIv2• 3 = FAPIv3• 4 = FAPIv4• 5 = FAPIv5• 6-254: reserved values. Future FAPI release will be signaled from this range.• 255 = No support for MAC's FAPI version. PHY will use the same version as ProtocolVersion.



Tag	Field	Type	Description
0x0055	num Entries	uint8_t	Number of entries in this TLV Value: 7 in this release
	more Than One Indication Per Slot	unit8_t [num Entries]	<p>Support flag for transmission of more than one indication per slot.</p> <p>This TLV is only supported at Common Context.</p> <p>Value, for each entry:</p> <ul style="list-style-type: none">• 1: at most one message instance per slot & numerology• 2: may generate more than one message instance per slot & numerology• 3: configurable by L2, via the flag indicationInstancesPerSlot – Defaults to at most one message instance per slot & numerology.• other values are reserved <p>Scope of each entry:</p> <ul style="list-style-type: none">[0]: RX_DATA.indication[1]: CRC.indication[2]: UCI.indication[3]: RACH.indication[4]: SRS.indication[5]: DL_TTI.response[6]: TIMING.indication
0x0056	num Entries	uint8_t	Number of entries in this TLV Value: 4 in this release



Tag	Field	Type	Description
	more Than One Request Per Slot	uint8_t [num Entries]	<p>Support flag for reception of more than one request per slot.</p> <p>This TLV is only supported at Common Context.</p> <p>Value, for each entry:</p> <ul style="list-style-type: none">• 1: at most one message instance per slot• 2: may receive more than one message instance per slot• 3: configurable by L2, via the flag requestInstancesPerSlot – Defaults to at most one message instance per slot & numerology.• other values are reserved <p>Scope of each entry:</p> <ul style="list-style-type: none">[0]: DL_TTI.request[1]: UL_TTI.request[2]: UL_DCI.request[3]: TX_DATA.request
0x015E	Connectivity Observation	uint32_t	<p>Support bitmap for observations and scope for Connectivity Indication.</p> <p>Value for each bit entry:</p> <ul style="list-style-type: none">• 0: not supported• 1: supported <p>Bit positions:</p> <ul style="list-style-type: none">[0]: PHY synchronization observable[1]: FEU synchronization observable[2]: PHY/FEU connectivity state observable.[3]: Observations possible for Common Context[4]: Observations possible for dedicated PHYs
0x015F	P5 Timeout	uint16_t	<p>Timeout for P5 procedures, in milliseconds.</p> <p>Value: 1 ... 10,000 ms</p>

Table 3–26 PHY Support



Tag	Field	Type	Description
This table describes the supported pairings of PHY and DFE Profiles. This TLV is only supported at Common Context.			
0x0054	numPhyProfiles	uint16	Shall be the same as numPhyProfiles in Table 3-26.
	numDfeProfiles	uint16	Number of DFE profiles. The n-th DFE profile corresponds to the n-th largest DFEprofileIdx, as defined in SCF-223 [9], starting from 1.
	profileValidityMap	uint8[x]	$x = \text{ceil floor}(\text{numPhyProfiles} * \text{numDfeProfiles}/8)$. (numPhyProfiles*D + P)-th bit of this map determines the validity of the following pair: <ul style="list-style-type: none"> [Phy Profile#P, DFE Profile#D] Where bits #n is the (n mod 8)-th LSB bit of profileValidityMap(floor(n/8)).

Table 3-27 PHY / DFE Profile Validity Map

Tag	Field	Type	Description
This table contains parameters pertaining to delay management, or other reserved parameter tags. These parameters shall only be defined by PHYs supporting Delay Management (see the Timing Management PHY capability)			
0x0101-0x0105	<reserved>	N/A	See section 3.2.9 of SCF-225 [10]
0x0106	DL_TTI Timing offset	uint32_t	See section 3.2.9 of SCF-225 [10]
0x0107	UL_TTI Timing offset	uint32_t	See section 3.2.9 of SCF-225 [10]
0x0108	UL_DCI Timing offset	uint32_t	See section 3.2.9 of SCF-225 [10]
0x0109	TX_DATA Timing offset	uint32_t	See section 3.2.9 of SCF-225 [10]
0x010A-0x011F	<reserved>	N/A	See section 3.2.9 of SCF-225 [10]
0x011E	Timing window	Uint16_t	See section 3.2.9 of SCF-225 [10]
0x011F	Timing info or Timing Indication mode	uint8_t	See section 3.2.9 of SCF-225 [10] Applies to Timing Info or Timing Indication APIs, depending on whether Delay Management with or without Timestamps is configured
0x0120	Timing info or Timing Indication period	uint8_t	See section 3.2.9 of SCF-225 [10] Applies to Timing Info or Timing Indication APIs, depending on whether Delay Management with or without Timestamps is configured

Table 3-28 Delay management parameters



Tag	Field	Type	Description
This table contains parameters pertaining to support of the 3GPP Rel-16 mTRP Feature, per 3GPP TS 38.300 [11], section 6.12			
0x0057	mTRP Support	uint32_t	<p>Support for the 3GPP mTRP feature. Each bit is set as follows:</p> <p>0 – not supported 1 – supported</p> <p>Bit positions:</p> <p>0 (LSB): mTRP SDM, single DCI 1 : mTRP FDM, single DCI scheme 2a 2 : mTRP FDM, single DCI scheme 2b 3 : mTRP TDM, single DCI scheme 3 4 : mTRP TDM, single DCI scheme 4 5 : mTRP multi-DCI 6 : one of two mTRPs (i.e. PHY can represent single TRP in an mTRP scheme) 7 : both mTRPs (i.e. PHY can represent both TRPs, in an mTRP scheme) 8-31: reserved, shall be set to 0 in this release.</p> <p>Note: Support for mTRP requires at least one of bits 6 and 7 to be set to 1. Also, at least one of the bits 0-5 should also be set to 1 if mTRP is supported.</p>

Table 3-29 Rel-16 mTRP parameters

Tag	Field	Type	Description
This table contains parameters pertaining to user plane			
0x007D	User Plane Encapsulation / Control Plane Separation	uint32_t	<p>Support for encapsulation of user plane payloads. Each bit is set as follows:</p> <p>0 – not supported 1 – supported</p> <p>Bit positions:</p> <p>[TX_DATA.request]</p> <ul style="list-style-type: none">• 0 (LSB): support for TBS-TLV tag value 0 in TX_DATA.request [payload embedded in message]• 1: support for TBS-TLV tag value 1 in TX_DATA.request [32-bit pointer to payload]• 2: support for TBS-TLV tag value 2 in TX_DATA.request [32-bit offset to payload]• 3: support for TBS-TLV tag value 3 in TX_DATA.request [UP/CP separation & and for non-zero Control Length signaling]• 4: support for TBS-TLV tag value 1 in TX_DATA.request [64-bit pointer to payload]• 5: support for TBS-TLV tag value 2 in TX_DATA.request [64-bit offset to payload]



Tag	Field	Type	Description
			<ul style="list-style-type: none">• 6: support for multiple TLVs in TBS-TLVs with tags 1 and 2• 7: Aggregation of TX_DATA.request messages with tag value 3 with other kinds (other than TX_DATA.request with tag value 3) messages:<ul style="list-style-type: none">◦ If this bit is set to 1, there is limit on aggregation, otherwise (if bit is set to 0) TX_DATA.request messages with tag value 3 may only be aggregated with other TX_DATA.request messages with the same tag value. <p><i>[RX_DATA.indication]</i></p> <ul style="list-style-type: none">• 8: support for TBS-TLV tag value 0 in RX_DATA.indication [payload embedded in message]• 9: support for TBS-TLV tag value 1 in RX_DATA.indication [32-bit pointer to payload]• 10: support for TBS-TLV tag value 2 in RX_DATA.indication [32-bit offset to payload]• 11: support for TBS-TLV tag value 3 in RX_DATA.indication [UP/CP separation & and for non-zero Control Length signaling]• 9: support for TBS-TLV tag value 1 in RX_DATA.indication [64-bit pointer to payload]• 10: support for TBS-TLV tag value 2 in RX_DATA.indication [64-bit offset to payload]• 14: support for multiple TLVs in TBS-TLVs with tags 1 and 2] <p><i>[RX_DATA/UCI/SRS.indication]</i></p> <ul style="list-style-type: none">• 15: Aggregation of uplink messages with tag value 3 with other kinds (other than with tag value 3) messages:<ul style="list-style-type: none">◦ If this bit is set to 1, there is limit on aggregation, otherwise (if bit is set to 0) uplink messages with tag value 3 may only be aggregated with other uplink messages with the same tag value. <p><i>[UCI.indication]</i></p> <ul style="list-style-type: none">• 16: support for TBS-TLV tag value 0 in UCI.indication [payload embedded in message]• 17: support for TBS-TLV tag value 1 in UCI.indication [32-bit pointer to payload]• 18: support for TBS-TLV tag value 2 in UCI.indication [32-bit offset to payload]• 19: support for TBS-TLV tag value 3 in UCI.indication [UP/CP separation & and for non-zero Control Length signaling]



Tag	Field	Type	Description
			<ul style="list-style-type: none">• 20: support for TBS-TLV tag value 1 in UCI.indication [64-bit pointer to payload]• 21: support for TBS-TLV tag value 2 in UCI.indication [64-bit offset to payload]• [22-23]: reserved <p><i>[SRS.indication]</i></p> <ul style="list-style-type: none">• 24: support for TBS-TLV tag value 0 in SRS.indication [payload embedded in message]• 25: support for TBS-TLV tag value 1 in SRS.indication [32-bit pointer to payload]• 26: support for TBS-TLV tag value 2 in SRS.indication [32-bit offset to payload]• 27: support for TBS-TLV tag value 3 in SRS.indication [UP/CP separation & and for non-zero Control Length signaling]• 28: support for TBS-TLV tag value 1 in SRS.indication [64-bit pointer to payload]• 29: support for TBS-TLV tag value 2 in SRS.indication [64-bit offset to payload]• [30-31]: reserved
0x007E	Message Encoding Support	uint8_t	<p>Describes padding for P7 messages.</p> <p>Values:</p> <ul style="list-style-type: none">- 0 (default): Legacy (implementation choice)- 1: P7 message parameters encoded without padding (unless otherwise specified for each individual FAPI parameter)- 2: P7 message parameters encoded with padding for alignment (as described in section 3.1.3)- 3-127: reserved128-255: vendor range

Table 3-30 User Plane and Encapsulation parameters

Tag	Field	Type	Description
This table contains the parameters relating to SRS PDU and Reports			
0x0080	supported Srs Usage	uint32_t	<p>Bitmap indicating support different types of SRS usage; Bit positions relating the allowable values in the Usage field of SRS PDU:</p> <ul style="list-style-type: none">- [0] = beamManagement- [1] = codebook- [2] = nonCodebook- [3] = antennaSwitching- [4] = positioning- [5-32] – reserved <p>For each bit position:</p> <ul style="list-style-type: none">- Value 0: not supported- Value 1: supported



Tag	Field	Type	Description
0x0160	numEntries	uint8_t	<p>Number of entries in this TLV Value: 5 in this release.</p>
	allowed Srs Reports Per Usage	uint16_t [n]	<p>n = numEntries.</p> <p>Bitmap indicating the type of reports supported per SRS usage; Bit positions relating the allowable reports in the Report field of SRS PDU.</p> <ul style="list-style-type: none">- [0] = Per-PRG and Symbol SNR- [1] = Normalized Channel I/Q Matrix- [2] = Channel SVD Representation- [3] = [Positioning]- [4-15] – reserved <p>Multiple bits may be set in the bitmap.</p> <p>Array positions correspond to the Usage of the SRS PDU where the report type is signaled:</p> <ul style="list-style-type: none">- [0] = beamManagement- [1] = codebook- [2] = nonCodebook- [3] = antennaSwitching- [4] = positioning <p>Note: the number of array entries in this TLV may increase in future releases</p>
0x0081	Rb Subsampling Resolution For Srs Derived Report	uint16_t [10]	<p>Minimum number of RB subsampling for any usage:</p> <ul style="list-style-type: none">- [0] = beamManagement- [1] = codebook- [2] = nonCodebook- [3] = antennaSwitching- [5-32] – reserved <p>In this release the size is 10, corresponding to the following usages:</p> <ul style="list-style-type: none">- [0] = beamManagement- [1] = codebook- [2] = nonCodebook- [3] = antennaSwitching- [4-9] = reserved <p>Value: 1 ... 275.</p>
0x0082	max Srs Channel Reports PerSlot	uint16_t [10]	<p>Up to as many PDUs send reports corresponding to those PDUs that have requested the report:</p> <p>In this release the size is 10, corresponding to the following usages:</p> <ul style="list-style-type: none">- [0] = beamManagement- [1] = codebook- [2] = nonCodebook- [3] = antennaSwitching



Tag	Field	Type	Description
			<ul style="list-style-type: none">- [4-9] = reserved <p>Value: 1 ... 65535</p>
0x0083	max Num Srs PDUs Per Slot	uint16_t	Maximum number of SRS PDUs per slot Value: 1 ... 65535
0x0084	iq Sample Formats	uint32_t	<p>Each bit indicates the supported IQ format for SRS reports:</p> <p>Bit positions:</p> <ul style="list-style-type: none">- [0]: 16-bit- [1]: 32-bit- [2-31]: reserved <p>Value for each bit position:</p> <ul style="list-style-type: none">- 0: not supported- 1: supported
0x0085	singular Value Formats	uint32_t	<p>Each bit indicates the supported singular value format:</p> <p>Bit positions:</p> <ul style="list-style-type: none">- [0]: 8-bit linear- [1]: 16-bit linear <p>Value for each bit position:</p> <ul style="list-style-type: none">- 0: not supported- 1: supported
0x0086	num Entries	uint8_t	value = 4 in this release
	max Srs Tx Ant Ports Per UE	uint8_t [n]	<p>n= numEntries</p> <p>Max number of UE ports that can be sounded, per srs usage and UE.</p> <p>In this release the entries corresponding to the following usages:</p> <ul style="list-style-type: none">- [0] = beamManagement- [1] = codebook- [2] = nonCodebook- [3] = antennaSwitching <p>Note: This TLV may accommodate more entries in the future.</p> <p>Value (per entry): 1 ... 16</p> <p>Notes:</p> <ul style="list-style-type: none">- This capability limits the number of SRS ports that L1 can be assumed to store for each UE, across all SRS Resources, in all Resource Sets, for a particular usage.- In this release, the usage is reflected by the Usage field in SRS PDU.
0x0087	max Num Consecutive Srs Slots	uint8_t	max number of slots in an SRS bursts Value: 1 .. 255



Tag	Field	Type	Description
0x0088	max Duty Cycle In Percentage	uint16_t	#slots in burst/(#slots in SRS burst + #non-SRS slots after burst) Expressed as value/100 (i.e. from 0.01 ... 100.00%, in steps of 0.01) Value: 0.001 .. 100.00, represented by 1 ... 10,000
0x0089	max Srs Comb Size	uint8_t	All SRS comb sizes up to this value are supported Value: 2, 4, 8
0x008A	max Num Srs Cyclic Shifts	uint8_t	Maximum number of SRS cyclic shifts. All numbers up to this value are supported. Value: 1 ... 12
0x008B	srs Symbols Bitmap	uint16_t [2]	14-bit bitmaps, indicating which symbols can be used for SRS. Array element positions are mapped as follows: <ul style="list-style-type: none">- [0]: bitmap for normal CP- [1]: bitmap for extended CP Bit position 0 corresponds to symbol position 0. Bit positions 14-15 are reserved, for normal CP. Bit positions 12-15 are reserved, for extended CP. Bit values: <ul style="list-style-type: none">- 0: symbol cannot be used for SRS- 1: symbol may be used for SRS
0x008C	max Num Srs Symb Per Slot	uint8_t	maximum number of symbols used by SRS per slot; Value: 1 ... 14
0x0161	SRS-based positioning measurements	uint32_t	Measurements reportable for SRS positioning. Value for each of the bit positions: <ul style="list-style-type: none">• 0: not supported• 1: supported Bit positions: <ul style="list-style-type: none">[0]: UL-RTOA[1]: Rx-Tx time difference[2]: UL AoA[3]: UL SRS-RSRP[4-31]: reserved
0x0162	puncturing of SRS symbols	uint8_t	Indicates whether puncturing of SRS symbols is supported (this is needed to allow PUSCH to puncture Rel-16 positioning SRS) Values: <ul style="list-style-type: none">• 0: not supported• 1: supported

Table 3-31 SRS parameters



This table contains MIMO capabilities introduced since FAPIv4			
Tag	Field	Type	Description
0x008E	max Num Carriers BW Layers Product DL	uint16_t [5]	<p>uint16_t[5] Maximum Num CC * BW in MHz * Layers product for DL across all CC that L1 can process for each of 5 numerologies. In order: 15, 30, 60, 120, 240 kHz.</p> <p>In this release, last entry shall be 0. Value 0xFFFF: not applicable; all other values are supported.</p>
0x008F	max Num Carriers BW Layers Product UL	uint16_t [5]	<p>uint16_t[5] Maximum Num CC * BW in MHz * Layers product for UL across all CC that L1 can process for each of 5 numerologies. In order: 15, 30, 60, 120, 240 kHz.</p> <p>In this release, last entry shall be 0. Value 0xFFFF: not applicable; all other values are supported.</p>
0x0090	max Num Carriers BW Antennas Product DL	uint16_t [5]	<p>uint16_t[5] Maximum Num CC * BW in MHz * Antennas product for DL across all CC that L1 can process for each of 5 numerologies. In order: 15, 30, 60, 120, 240 kHz.</p> <p>In this release, last entry shall be 0. Value 0xFFFF: not applicable; all other values are supported.</p>
0x0091	max Num Carriers BW Antennas Product UL	uint16_t [5]	<p>uint16_t[5] Maximum Num CC * BW in MHz * Antennas product for UL across all CC that L1 can process for each of 5 numerologies. In order: 15, 30, 60, 120, 240 kHz.</p> <p>In this release, last entry shall be 0. Value 0xFFFF: not applicable; all other values are supported.</p>



Tag	Field	Type	Description
0x0092	muMimoDI	uint32_t	<p>Indicates support for MU-MIMO spatial multiplexing, per DL channel type.</p> <p>Bit positions:</p> <ul style="list-style-type: none">- [0]: PDSCH- [1]: PDCCH- [2]: CSI-RS- [3-31] reserved <p>Value for each bit position:</p> <ul style="list-style-type: none">- 0: not supported- 1: supported
0x0093	supported Spatial Mux Per DL Channel Type	uint32_t [3]	<p>Bitmap indicating support different types of precoding + beamforming.</p> <p>Bit position:</p> <ul style="list-style-type: none">- [0]: L2-signaled beams and precoders, using pre-stored PM and DBT- [1]: L1 derived SRS reciprocity based joint Tx precoding <p>Value for each bit position:</p> <ul style="list-style-type: none">- 0: not supported- 1: supported <p>Array entries correspond to DL channels types as follows:</p> <ul style="list-style-type: none">- [0]: PDSCH- [1]: PDCCH- [2]: CSI-RS
0x0094	max Num MuMimo Layers Per DI Channel Type	uint8_t [3]	<p>Maximum number of MuMIMO layers per DL channel type</p> <p>Value: 1-255; 1 = no Mu-MIMO</p> <p>Array entries correspond to DL channels types as follows:</p> <ul style="list-style-type: none">- [0]: PDSCH- [1]: PDCCH- [2]: CSI-RS <p>In this release, non-precoded CSI-RS cannot be spatial multiplexed with other channels</p>



Tag	Field	Type	Description
0x0095	cross Channel Spatial Multiplexing For DL	uint16_t	<p>Signals support for spatial multiplexing of different DL channel types</p> <p>Bit positions:</p> <ul style="list-style-type: none">- [0]: PDSCH & PDCCH- [1]: PDSCH & CSI-RS- [2]: PDCCH & CSI-RS <p>Value for each bit position:</p> <ul style="list-style-type: none">- 0: not supported- 1: supported <p>All other positions are reserved and should be set to 0 in this release</p>
0x0096	muMimoUl	uint32_t	<p>Indicates support for MU-MIMO spatial multiplexing, per UL channel type.</p> <p>Bit positions:</p> <ul style="list-style-type: none">- [0]: PUSCH- [1]: PUCCH- [2-31] reserved <p>Value for each bit position:</p> <ul style="list-style-type: none">- 0: not supported- 1: supported
0x0097	supported Spatial Mux Per UL Channel Type	uint32_t [2]	<p>Bitmap indicating support different types of combining.</p> <p>Bit position:</p> <ul style="list-style-type: none">- [0]: L2-signaled beams, using pre-stored DBT- [1]: L1 derived SRS based joint Rx combining <p>Value for each bit position:</p> <ul style="list-style-type: none">- 0: not supported- 1: supported <p>Array entries correspond to DL channels types as follows:</p> <ul style="list-style-type: none">- [0]: PUSCH- [1]: PUCCH
0x0098	max Num MuMimo Layers Per UL Channe lType	uint8_t [2]	<p>Maximum number of MuMIMO layers per UL channel type</p> <p>Value: 1-255; 1 = no Mu-MIMO</p> <p>Array entries correspond to UL channels types as follows:</p> <ul style="list-style-type: none">- [0]: PUSCH- [1]: PUCCH



Tag	Field	Type	Description
0x0099	cross Channel Spatial Multiplexing For UL	uint16_t	<p>Signals support for spatial multiplexing of different UL channel types</p> <p>Bit positions:</p> <ul style="list-style-type: none">- [0]: PUSCH & PUCCH <p>Value for each bit position:</p> <ul style="list-style-type: none">- 0: not supported- 1: supported <p>All other positions are reserved and should be set to 0 in this release</p>
0x009A	max MuMimo Groups In Fd Per DI Channel	uint16_t [3]	<p>Max number of MuMimo groups over frequency domain RBs for any one time domain symbol, per DL channel type</p> <p>Array entries correspond to DL channels types as follows:</p> <ul style="list-style-type: none">- [0]: PDSCH- [1]: PDCCH- [2]: CSI-RS <p>Value: 1->273</p>
0x009B	max MuMimo Groups In Td Per DI Channel	uint8_t [3]	<p>Max number of MuMimo groups over time domain symbols on any frequency domain RB, per DL channel type</p> <p>Array entries correspond to DL channels types as follows:</p> <ul style="list-style-type: none">- [0]: PDSCH- [1]: PDCCH- [2]: CSI-RS <p>Value: 1->14</p>
0x009C	max MuMimo New Precoders Per Slot	uint16_t	<p>Max number of precoders that can be refreshed per slot. A refreshed joint precoder corresponds to a PRG over which at least one layer is changed or updated from the previous slot.</p> <p>Value: 1-> 65535</p>



Tag	Field	Type	Description
0x009D	all Prbs Groups Prgs	Uint8_t	<p>Indicates whether all PRBs are contained in MU-MIMO PRGs (except possibly BWP edge PRBs).</p> <p>If this capability is set to 1, a single PRG can only be referenced once in prbBitmaps for any one UE's MU-MIMO groups. Applies to both UL and DL MU-MIMO groups.</p> <p>Value:</p> <ul style="list-style-type: none">- 0: no need to ensure all PRBs are contained in the PRGs- 1: all PRG PRBs are contained in PRGs.
0x009E	max MuMimo Groups In Fd Per Ul Channel	uint16_t [2]	<p>Max number of [Channel] MuMimo groups over frequency domain RBs for any one time domain symbol, per UL chanel type</p> <p>Array entries correspond to UL channels types as follows:</p> <ul style="list-style-type: none">- [0]: PUSCH- [1]: PUCCH <p>Value: 1->273</p>
0x009F	max MuMimo Groups In Td Per Ul Channel	uint8_t [2]	<p>Max number of MuMimo groups over time domain symbols on any frequency domain RB, per UL channel type</p> <p>Array entries correspond to UL channels types as follows:</p> <ul style="list-style-type: none">- [0]: PUSCH- [1]: PUCCH <p>Value: 1->14</p>
0x0150	min Rb Resolution For Srs Derived Reports	uint16_t	<p>smallest number of RBs for which L1 reports an SVD channel abstraction.</p> <p>Value: 1 .. 275</p> <p>Note: In this release, SVD channel abstractions are reported as indicated in TLV 0x0160</p>



Tag	Field	Type	Description
0x0151	max Number DI Spatial Streams	uint16_t	<p>Maximum number of spatial streams for transmitting layers per slot, across all channel types. When PHY declares this capability, MAC signals a DL spatial stream ID to each DL layer, ensuring that each RE in a symbol is associated with a single spatial stream, unless otherwise allowed by dlSpatialStreamChannelPriority. If non-unique assignment is allowed, the higher priority layer punctures the lower priority layer.</p> <p>Range: 1-256: the number of spatial streams (MAC labels DL layers with spatial streams 0 ... (maxNumberDISpatialStreams-1)) 65535 (default): no support (MAC need not assign DL spatial streams to DL layers, in this case) Other values: reserved.</p>
0x0152	dl Spatial Stream Channel Priority	uint32_t	<p>For each bit position 0 = not supported / 1 = supported</p> <p>Bit positions:</p> <ul style="list-style-type: none">#0: Higher-priority CSI-RS may share a spatial stream ID with a lower priority PDSCH.#1 - #31: reserved, should be set to 0.
0x0153	max Number UI Spatial Streams	uint16_t	<p>Maximum number of spatial streams for receiving UL transmissions per slot, across all channel types. When PHY declares this capability, MAC signals a set of UL spatial stream IDs to each channel, ensuring that each:</p> <ul style="list-style-type: none">- #UL spatial streams \geq #layers for that UL channel.- an UL spatial stream ID cannot be assigned to channels that overlap in time and frequency domain in a slot. <p>Range: 1-256: the number of spatial streams (MAC labels UL channels with spatial streams 0 ... (maxNumberUISpatialStreams-1)) 65535 (default): no support (MAC need not assign UL spatial streams to UL channels, in this case) Other values: reserved.</p>



Tag	Field	Type	Description
0x0154	ul Tpmi Capability	uint8_t	Indicates whether ul schedule must follow the usage of the latest SRS report, or not: <ul style="list-style-type: none">- 0 ulTpmiIndex must match on latest SRS report (codebook or non-codebook);- 1: ulTpmiIndex need not match latest SRS report

Table 3–32 Spatial Multiplexing and MIMO Capabilities

3.3.2 CONFIG

The CONFIG message exchange was described in section 2.1.1.2.

3.3.2.1 CONFIG.request

The CONFIG.request message is given in Table 3–33. From this table it can be seen that CONFIG.request contains a list of TLVs describing how the PHY should be configured. This message may be sent by the L2/L3 software when the PHY is in any state.

The full list of CONFIG TLVs is given in Section 3.3.2.4 and the PARAM message provided L2/L3 a list of supported CONFIG TLVs and states where these TLVs are valid.

There is no requirement for the L2/L3 software to provide the TLVs in the order specified in the Tables.

Field	Type	Description
Number of TLVs	uint8_t	Number of TLVs contained in the message body.
TLVs	Variable	See Table 3–36

Table 3–33 CONFIG.request message body

3.3.2.2 CONFIG.response

The CONFIG.response message is given in Table 3–34. If the configuration procedure was successful, then the error code returned will be MSG_OK and no TLV tags will be included. If the configuration procedure was unsuccessful, then MSG_INVALID_CONFIG will be returned, together with a list of TLVs identifying the problem.

Regardless of the error code value, the following fields are always present, even if set to 0:

- Number of Invalid or Unsupported TLVs
- Number of Invalid TLVs that can only be configured in IDLE
- Number of Invalid TLVs that can only be configured in RUNNING
- Number of Missing TLVs



Field	Type	Description
Error Code	uint8_t	See Table 3-35.
Number of Invalid or Unsupported TLVs	uint8_t	N1=Number of invalid or unsupported TLVs contained in the message body. Value: 0->255
Number of invalid TLVs that can only be configured in IDLE	uint8_t	N2=Number of TLVs contained in the message body, which are valid in IDLE state but not in this PHY state. If PHY is in IDLE state this will always be 0. Value: 0->255
Number of invalid TLVs that can only be configured in RUNNING	uint8_t	N3=Number of TLVs contained in the message body, which are valid in RUNNING state but not in this PHY state. If PHY is in RUNNING state this will always be 0. Value: 0->255
Number of Missing TLVs	uint8_t	N4=Number of missing TLVs contained in the message body. If the PHY is in the CONFIGURED state this will always be 0. Value: 0->255
A list of invalid or unsupported TLVs – each TLV is represented by its TLV tag number.		
TLV	uint16_t [N1]	Array of TLV tags
A list of invalid TLVs that can only be configured in IDLE – each TLV is represented by its TLV tag number		
TLV	uint16_t [N2]	Array of TLV tags
A list of invalid TLVs that can only be configured in RUNNING – each TLV is represented by its TLV tag number		
TLV	uint16_t [N3]	Array of TLV tags
A list of missing TLVs – each TLV is represented by its TLV tag number		
TLV	uint16_t [N4]	Array of TLV tags

Table 3-34 CONFIG.response message body

3.3.2.3 CONFIG errors

The error codes that can be returned in CONFIG.response are given in Table 3-35.

Error code	Description
MSG_OK	Message is OK.
MSG_INVALID_CONFIG	The configuration provided has missing mandatory TLVs, or TLVs that are invalid or unsupported in this state.

Table 3-35 Error codes for CONFIG.response



3.3.2.4 CONFIG TLVs

PARAM and CONFIG TLVs are used in the PARAM and CONFIG message exchanges, respectively. For both the PARAM and CONFIG TLVs the TLV format is given in Table 3–9.

The TLVs for the CONFIG message exchange are given in this Section 3.3.2.4 and for the PARAM message exchange in Section 3.3.1.3.

CONFIG TLVs used in the CONFIG message exchange are given in Table 3–36.

Field	Type	Description
PHY Config	struct	See Table 3–37
Carrier Config	struct	See Table 3–38
Cell Config	struct	See Table 3–39
SSB Power and PBCH Config	struct	See Table 3–40
PRACH Config	struct	See Table 3–41
Multi-PRACH Config	struct	See Table 3–42
Multi MsgA-PUSCH Config	struct	See Table 3–45
SSB Resource Config Table	struct	See Table 3–46
Multi-SSB Config	struct	See Table 3–47
TDD Table	struct	See Table 3–48
Measurement Config	struct	See Table 3–49
Beamforming Table	struct	See Table 3–50
Precoding Table	struct	See Table 3–50
Uci Configuration	struct	See Table 3–51
PRB-Symbol Rate Match Patterns	struct	See Table 3–52
LTE-CRS Rate Match Patterns	struct	See Table 3–54
PUCCH Semi-static Config	struct	See Table 3–55
PDSCH Config	struct	See Table 3–56
Delay Management Config	struct	See Table 3–57
Rel-16 mTRP Config	struct	See Table 3–58

Table 3–36 CONFIG TLVs

To aid additions of future TLVs in later versions of this API, the highest value tag in the Tables below is 0x1047. In addition, the tag value range 0x0100–0x014F only to be used for alignment with nFAPI tag value ranges.



Tag	Field	Type	Description
This table contains the configuration parameters relating to PHY ID definition.			
0x102A	phy Profile Id	uint16_t	The selected PHY Profile Id (see per Table 3-26) This TLV is only supported for Common Context.
0x102B	num Entries	uint8_t	Number of entries in this TLV Value: 7 in this release
	indication Instances Per Slot	unit8_t [num Entries]	Configuration for transmission of more than one indication per slot. This TLV is only supported for Common Context. Value, for each entry: <ul style="list-style-type: none">• 0: no (change in) configuration• 1: limit to one message instance per slot & numerology• 2: allow generation of more than one message instance per slot & numerology• other values are reserved Scope of each entry: [0]: RX_DATA.indication [1]: CRC.indication [2]: UCI.indication [3]: RACH.indication [4]: SRS.indication [5]: DL_TTI.response [6]: TIMING.indication
0x102C	num Entries	uint8_t	Number of entries in this TLV Value: 4 in this release



Tag	Field	Type	Description
	request Instances Per Slot	unit8_t [num Entries]	<p>Configuration for reception of more than one request per slot.</p> <p>This TLV is only supported for Common Context.</p> <p>Value, for each entry:</p> <ul style="list-style-type: none">• 0: no (change in) configuration• 1: expect at most one message instance per slot & numerology• 2: L2 may send more than one message instance per slot & numerology• other values are reserved <p>Scope of each entry:</p> <ul style="list-style-type: none">[0]: DL_TTI.request[1]: UL_TTI.request[2]: UL_DCI.request[3]: TX_DATA.request
0x103D	SFN/SL TLV: when this TLV is included in CONFIG.request for a minor reconfiguration procedure, it signals the SFN and slot (for the given numerology and cyclic prefix) where the new configuration should be applied, as described in section 2.1.5.		
	SFN	Uint16_t	System frame number where the minor reconfiguration procedure applies. Special value of 0xFFFF indicates earliest SFN available (i.e. ASAP) Range: 0..1023
	Slot	Uint8_t	Slot index (as per TS38.211 ver 15.5.0 sec 4.3.2) where the minor reconfiguration procedure applies. Special value of 0xFF indicates earliest slot available (i.e. ASAP) Range: 0..159
	Subcarrier Spacing	Uint8_t	SubcarrierSpacing index as defined in TS38.211 ver 15.5.0 sec 4.2 0: 15 KHz 1: 30 KHz 2: 60 KHz 3: 120 KHz 4: 240 KHz
	Cyclic Prefix Type	Uint8_t	0: Normal CP 1: Extended CP

Table 3-37 PHY configuration



Tag	Field	Type	Description
This table contains the configuration parameters relating carrier configuration. The PARAM.response message will have indicated which of these parameters is supported by the PHY and in which PHY states these parameters are mandatory or optional.			
0x102D	dl Bandwidth	uint16_t	Carrier bandwidth for DL in MHz [3GPP TS 38.104 [7], sec 5.3.2] Values: 5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100, 200, 400
	dl Frequency	uint32_t	Absolute frequency of DL in KHz [3GPP TS 38.104 [7], sec 5.2 and 3GPP TS 38.211 [2], sec 4.4.4.2] Interpreted as PointA or f0, depending on the PHY capability TLV meaningOfCarrierFrequency Value: 450000 -> 52600000
	dlk0	uint16_t [5]	k_0^μ for each of the numerologies [3GPP TS 38.211 [2], sec 5.3.1] Value: 0 -> 23699
	dl Grid Size	uint16_t [5]	Grid size $N_{grid}^{size,\mu}$ for each of the numerologies [3GPP TS 38.211 [2], sec 4.4.2] Value: 0 -> 275 0 = this numerology not used
	num Tx Ant	uint16_t	Number of Tx antennas Value: 0 -> 65355 If PHY Profiles are supported, per , numTxAnt cannot exceed the number of DL BB ports for the PHY, in the selected profile.
	uplink Bandwidth	uint16_t	Carrier bandwidth for UL in MHz. [3GPP TS 38.104 [7], sec 5.3.2] Values: 5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100, 200, 400
	uplink Frequency	uint32_t	Absolute frequency of UL in KHz [3GPP TS 38.104 [7], sec 5.2 and 3GPP TS 38.211 [2], sec 4.4.4.2] Interpreted as PointA or f0, depending on the PHY capability TLV meaningOfCarrierFrequency Value: 450000 -> 52600000
	ulk0	uint16_t [5]	k_0^μ for each of the numerologies [3GPP TS 38.211 [2], sec 5.3.1] Value: 0 -> 23699
	ul Grid Size	uint16_t [5]	Grid size $N_{grid}^{size,\mu}$ for each of the numerologies [3GPP TS 38.211 [2], sec 4.4.2]. Value: 0 -> 275 0 = this numerology not used
	num Rx Ant	uint16_t	Number of Rx antennas Value: 0 -> 65355 If PHY Profiles are supported, per Table 3-26, numRxAnt cannot exceed the number of



Tag	Field	Type	Description
			UL BB ports for the PHY, in the selected profile
	Frequency Shift 7p5 KHz	uint8_t	Indicates presence of 7.5KHz frequency shift. Value: 0 = false 1 = true
	Power Profile	uint8_t	Indicates the selected Power Profile (see section 2.2.4.5 and capability tag 0x0039) 0 = ProfileNR 1 = ProfileSSS
	Power Offset RsIndex	uint8_t	RS index for configuring power offset, among the RS indices supported, based on capability PowerProfilesSupported (tag 0x0038): Bit index location represents the same channels as in Table 3-12: <ul style="list-style-type: none">- Bit#0 [LSB]: PDCCH-DMRS- Bit#1: PDCCH-Data- Bit#2: PDSCH-DMRS- Bit#3: PDSCH-Data- Bit#4: PDSCH-PTRS- Bit#5: CSI-RS- Bit#6: PSS- Bit#7 [MSB]: SSS Value range for each bit: <ul style="list-style-type: none">• 0: Power offsets as signaled with respect to RS Index 0 in Table 3-12• 1: Power offsets as signaled with respect to RS Index 1 in Table 3-12

Table 3-38 Carrier configuration

Tag	Field	Type	Description
This table contains the configuration parameters relating cell configuration. The PARAM.response message will have indicated which of these parameters is supported by the PHY and in which PHY states these parameters are mandatory or optional.			



Tag	Field	Type	Description
0x100C	phyCellId	uint16_t	Physical Cell ID, N_{ID}^{cell} [3GPP TS 38.211 [2], sec 7.4.2.1] Value: 0 ->1007
0x100D	FrameDuplexType	uint8_t	Frame duplex type Value: 0 = FDD 1 = TDD
0x102E	pdschTransType Validity	uint8_t	0 = L1 must ignore pdschTransType in PDSCH PDU 1 = L1 must interpret pdschTransType in PDSCH PDU
0x102F	puschTransType Validity	uint8_t	0 = L1 must ignore puschTransType in PUSCH PDU 1 = L1 must interpret puschTransType in PUSCH PDU

Table 3–39 Cell configuration

Tag	Field	Type	Description
This table contains the configuration parameters relating SSB/PBCH configuration. The PARAM.response message will have indicated which of these parameters is supported by the PHY and in which PHY states these parameters are mandatory or optional			
0x100E	ssPbchPower	uint32_t	SSB Block Power Value: TBD (-60..50 dBm)
0x1030	ssPbchBlockPowerScaling	int16_t	<p>Baseband power scaling applied to SS-PBCH signal [3GPP TS 38.211 [2], sec 7.4.2.3.1]</p> <p>Values: $[ssPbchBlockPowerScaling-Unit-Choice = dB]: -11000 -> 12000$ representing -110.0 dB to 120.0 dB in 0.01dB, i.e. scaling is applied to the signal generated in the above sections.</p> <p>$[ssPbchBlockPowerScaling-Unit-Choice = dBFS]: -11000 -> 0$ representing -110.0 dBFS to 0.0 dBFS in 0.01dB, i.e. scaling is applied with respect to the maximum power for the cell.</p> <p>L1 uses to TLV if configured with PowerOffsetRsIndex bit#7 is set to 0</p>
0x100F	bchPayloadFlag	uint8_t	<p>Defines option selected for generation of BCH payload, see Table 3-13 (v0.0.011)</p> <p>Value:</p> <ul style="list-style-type: none"> 0: MAC generates the full PBCH payload 1: PHY generates the timing PBCH bits 2: PHY generates the full PBCH payload

Table 3–40 SSB power and PBCH configuration



Tag	Field	Type	Description
The TLVs from this table signaled directly on <code>CONFIG.request</code> apply to PRACH resource configuration index 0. See also the Multi-PRACH Configuration Table 3-42.			
0x1031	prachResConfigIndex	uint16_t	A number uniquely identifying the PRACH resource configuration. If signaled directly in <code>CONFIG.request</code> , it set to 0. Multi-Prach Configuration Table 3-42 is used if multiple configurations are needed.
	prachSequenceLength	uint8_t	RACH sequence length. Long or Short sequence length. Only short sequence length is supported for FR2. [3GPP TS 38.211 [2], sec 6.3.3.1] Value: 0 = Long sequence 1 = Short sequence
	prachSubCSpacing	uint8_t	Subcarrier spacing of PRACH. [3GPP TS 38.211 [2], Tables 6.3.3.1-1, 6.3.3.1-2 and 6.3.3.1-7]. Represents Δf_{RA} in [3GPP TS 38.211 [2], section 5.3.2] Value: 0: 15 kHz 1: 30 kHz 2: 60 kHz 3: 120 kHz 4: 1.25 kHz 5: 5 kHz
	ulBwpPuschScs	uint8_t	The PUSCH channel subcarrier spacing of UL BWP which associated with this PRACH Configuration. Corresponds to “ Δf for PUSCH” in [3GPP TS 38.211 [2], Table 6.3.3.2-1] Value (corresponding to the entries of [3GPP TS 38.211 [2], Table 4.2-1]): 0 -> 3
	restrictedSetConfig	uint8_t	PRACH restricted set config Value: 0: unrestricted 1: restricted set type A 2: restricted set type B
	numPrachFdOccasions	uint8_t	Number of RACH frequency domain occasions. Corresponds to the parameter M in [3GPP TS 38.211 [2],



Tag	Field	Type	Description
			sec 6.3.3.2] which equals the higher layer parameter msg1FDM Value: 1,2,4,8
	prachConfigIndex	uint8_t	PRACH configuration index reflected in RRC parameter prach-ConfigurationIndex for the configuration. Value: per 3GPP TS 38.331 [6], section 6.3.2 For instance, 0...262 in Rel-16.
	prachFormat	uint8_t	RACH format information for this PRACH configuration. This corresponds to one of the supported formats i.e., 0, 1, 2, 3, A1, A2, A3, B1, B4, C0, C2, A1/B1, A2/B2 or A3/B3. [3GPP 38.211 [2], sec 6.3.3.2] Values: 0 = 0 1 = 1 2 = 2 3 = 3 4 = A1 5 = A2 6 = A3 7 = B1 8 = B4 9 = C0 10 = C2 11 = A1/B1 12 = A2/B2 13 = A3/B3
	numPrachTdOccasions	uint8_t	$N_t^{RA,slot}$: the "number of Time domain PRACH occasions within a PRACH slot", per 3GPP TS 38.211 [2] section 6.3.3.2. Set to 1 for long sequences (0-3). Note: this field can be determined directly from prachConfigIndex
	numberOfPreambles	uint8_t	total number of preambles used by the RACH configuration. Corresponds to $N_{preamble}^{total}$ in 3GPP TS 38.213 [4], section 8.1 Value: 1 ... 64



Tag	Field	Type	Description
	startPreambleIndex	uint8_t	preamble index start for each PRACH occasion in this PRACH configuration, per 3GPP TS 38.213 [4], section 8.1 Note: For example, this may be 0 for separate PRACH configurations (type 1 or 2), or 'R' for a type-2 configuration common with a type-1 configuration with 'R' preambles.
For numPRACHFdOccasions {			
	prachRootSequenceIndex	uint16_t	Starting logical root sequence index, i , equivalent to higher layer parameter <i>prach-RootSequenceIndex</i> [3GPP TS 38.211 [2], sec 6.3.3.1] Value: 0 -> 837
	numRootSequences	uint8_t	Number of root sequences for a particular FD occasion that are required to generate the necessary number of preambles
	prachFrequencyOffset	int16_t	Frequency offset (from UL bandwidth part) for each FD, with exact meaning based on PARAM prachFrequencyOffsetMeaning as one of: <ul style="list-style-type: none">offset between Point-A and a PRB overlapping with the lowest RE of the PRACH signal, measured in PRBs at the minimum numerology between the PRACH & PUSCH numerologies (ulBwpPuschScs)parameter k_1 [3GPP TS 38.211 [2], sec 5.3.2] in units of PRBs at ulBwpPuschScs
	PRACH guardband offset	uint16_t	Offset (in units of PRACH SCS) between the PRB indicated by <i>prachFrequencyOffset</i> and the first RE of the PRACH signal.
	prachZeroCorrConf	uint8_t	PRACH Zero CorrelationZone Config which is used to derive N_{cs} [3GPP TS 38.211 [2], sec 6.3.3.1] Value: from 0 to 15
	numUnusedRootSequences	uint16_t	Number of unused sequences available for noise estimation per FD occasion. At least one unused root sequence is required per FD occasion.
	For numUnusedRootSequences {		



Tag	Field		Type	Description
		unusedRootSequences	uint16_t	Unused root sequence or sequences per FD occasion. Required for noise estimation.
	}			
	}			
	SsbPerRach	uint8_t	SSB-per-RACH-occasion Value: 0: 1/8 1:1/4, 2:1/2 3:1 4:2 5:4, 6:8 7:16	

Table 3-41 PRACH configuration

Tag	Field	Type	Description
This table contains multiple PRACH configurations for the carrier.			
0x1032	numPrachConfigurations	uint16_t	Number of PRACH configurations supported in the carrier (in addition to PRACH configuration index 0)
	numPrachConfigurations TLVs	Set of PRACH configuration TLVs (Table 3-41)	<p>A set of numPrachConfigurations TLVs.</p> <p>Each PRACH configuration TLV shall contain a unique prachResConfigIndex between 1 and numPrachConfigurations.</p> <p>Note: These numPrachConfigurations TLVs are additional to PRACH configuration index 0 – if any – signaled directly via Table 3-41 TLVs in CONFIG.request.</p>

Table 3-42 Multi-PRACH configuration table

Tag	Field	Type	Description
This table contains a MsgA-PUSCH configuration for the carrier.			
0x103E	msgA-puschResConfigIndex	uint16_t	A reference index to the MsgA PUSCH resource on which the MsgA PUSCH reception operation is performed.



Tag	Field	Type	Description
			Value: 0 ... 65534; value 65535 is reserved.
	msgA-prachResConfigIndex	uint16_t	A pointer linked to the PRACH resource config which carries the MsgA preamble transmission associated with this MsgA PUSCH configuration
	msgA-groupAorB	uint8_t	The group for this MsgA-PUSCH configuration (c.f. 3GPP TS 38.331 [13], section 6.3.2; 3GPP TS 38.321 [21], sections 5.1.1 and 5.1.2a) Value: 0 = Group A 1 = Group B
	nIdMsgAPusch	uint16_t	The scrambling ID used to generate MsgA PUSCH data scramble sequence, corresponding to n_ID in 3GPP TS 38.211 [12], section 6.3.1.1 [2]. Note: This is used together with RAPID and RA-RNTI to determine the scrambling sequence generator. Value: 0 -> 1023
	dmrsPorts	uint16_t	Bitmap of all the DMRS ports amongst which ports are picked for each PUSCH occasion of the MsgA PUSCH configuration. [3GPP TS 38.212 [3], 7.3.1.1.2] based on msgA-PRACH → msgA-PUSCH mapping Bit mapping is as follows: <ul style="list-style-type: none">• bit 0 : antenna port 1000• bit 7: antenna port 1007 For each bit <ul style="list-style-type: none">• 0: DMRS port not used• 1: DMRS port used numDmrsPorts = sum of all bits in dmrsPorts Note: the actual DMRS port to use is based on the msgA-PRACH → msgA-PUSCH & associated DMRS mapping
	available-DmrsSequenceIds	uint8_t	Bitmap indicating the DMRS sequence IDs available for this MsgA-PUSCH Configuration. For each bit position: <ul style="list-style-type: none">• 0: ID not supported• 1: ID supported Bit significance: <ul style="list-style-type: none">- [0]: n_SCID = 0 available.



Tag	Field	Type	Description
			<ul style="list-style-type: none">- [1]: nSCID = 1 available- [2-7]: n/a (bit value shall be set to 0) numDmrsSequences = sum of all bits in avaialbleDmrsSequenceIds
	puschDmrsScramblingIds	uint16_t [2]	The values of parameter N_{ID}^0 (first entry) and N_{ID}^1 (second entry), as specified in 3GPP TS 38.211 [12], section 6.4.1.1.1.1 for this MsgA-PUSCH Value: 0 -> 65535
	numPuschOcasFd	uint8_t	The number of frequency domain PUSCH occasions within the UL slot for this Msg-A PUSCH Configuration. Value: 1, 2, 4, 8
	numPuschOcasTd	uint8_t	The number of time domain PUSCH occasions for this Msg-A PUSCH Configuration. Value: 1, 2, 3, 6
	numPuschOcasSlots	uint8_t	The number of slots configured for this MsgA-PUSCH configuration; Value: 1...4
	msgA-PUSCH-TimeDomainOffset	uint8_t	The time domain offset, as specified in [3GPP TS 38.213 [22], section 8.1A] in slots between the PRACH Configuration slot where a MsgA-PRACH preamble may be triggered and the corresponding MsgA-PUSCH occasion for this configuration. Value: 1...32
	Npreambles	uint16_t	$N_{preambles}$, per section 8.1A of 38.213, as computed by L2 for the particular mapping of MsgA-PRACH to Msg-A PUSCH mapping relevant to this MsgA-PUSCH configuration. Values: 1 ... 3,584
	Association Pattern Period (in slots)	uint8_t	#ApPSlots = The size of the association pattern period, measured in PRACH slots, for the PRACH configuration pointed to by msgA-prachResConfigIndex Expected to correspond to (10, 20, 40, 80, 160) ms in PRACH slots
	For each {set of PRACH slot} in the association pattern period {		
	numPrachSlots-for Prach-to-PruAndDmrs-Map	uint16_t	#sameMapSlots Number of PRACH slots sharing the same PRACH to MsgA-PUSCH&Dmrs map.



Tag	Field	Type	Description
	prachSlotIndices-Modulus ApPSlots	uint16_t [n]	<p>$n = [\# \text{sameMapSlots}]$ Set of PRACH slot indices modulus $\# \text{ApPSlots}$ For this computation: PRACH slot index = SFN * $\# \text{slots}/10\text{ms} + (\text{slot indication within}$ $10 \text{ ms})$</p>
	MsgA-PRACH-to-(PRU & DMRS) mapping	See Table 3-44	<p>For this mapping, the following are determined from the PRACH configuration indicated by msgA-prachResConfigIndex:</p> <ol style="list-style-type: none"> 1. $\text{prachFd1} \dots \text{prachFdMax} = 0 \dots \text{numPrachFdOccasions}-1$ 2. $\text{prachTd1} \dots \text{prachTdMax} = 0 \dots \text{numPrachTdOccasions}-1$ 3. $\text{grpStart} = \text{grpEnd}$ (= group A or B or a dedicated group; exact index is irrelevant, since only the number of groups matters in Table 3-44) <p>The preamble group is the group for the MsgA-PUSCH configuration, i.e.</p> <p>Furthermore, the following parameters are used to interpret the contents of the mapping:</p> <ul style="list-style-type: none"> • $N_{\text{preambles}}$: Npreambles • numPuschOcasFd: numPuschOcasFd • numDmrsPorts: sum of all bits in dmrsPorts • numDmrsSequences: sum of all bits in available-DmrsSequenceIds • numPuschOcasTd: numPuschOcasTd • numPuschOcasSlots: numPuschOcasSlots • $\text{startPreambleIndex}$: startPreambleIndex in the PRACH configuration indicated by msgA-prachResConfigIndex • numberOfPreambles: numberOfPreambles in the PRACH configuration indicated by msgA-prachResConfigIndex
	}		

Table 3-43 **MsgA-PUSCH configuration table**



Field	Type	Description	
This table contains MsgA-PRACH-to-(PRU & DMRS) map multiple PRACH configurations for the carrier. This structure is parametrized by:			
1. a range of PRACH FD occasions (<i>prachFd1</i> ... <i>prachFdMax</i>), from which preambles are mapped 2. a range of PRACH TD occasions (<i>prachTd1</i> ... <i>prachTdMax</i>), from which preambles are mapped 3. a range of prach preamble groups (<i>grpStart</i> ... <i>grpEnd</i>), to which preambles belong			
Furthermore, interpretation of the contents requires awareness of the following:			
<ul style="list-style-type: none">• <i>N_preambles</i>: the number of preambles that can map to a “valid PUSCH occasion and the associated DMRS resource” [3GPP 38.213, section 8.1A])• <i>numPuschOcasFd</i>: number of frequency domain occasions for the MsgA-PUSCH• <i>numDmrsPorts</i>: number of DMRS ports for the MsgA-PUSCH• <i>numDmrsSequences</i>: number of DMRS sequences for the MsgA-PUSCH• <i>numPuschOcasTd</i>: number of time domain occasions for the MsgA-PUSCH• <i>numPuschOcasSlots</i>: number of time domain occasions for the MsgA-PUSCH• <i>startPreambleIndex</i>: start preamble for each PRACH RO.• <i>numberOfPreambles</i>: total number of preambles for each PRACH RO			
All the above italicized parameters are provided as an input for this structure.			
For fdIndex = <i>prachFd1</i> ... <i>prachFdMax</i>			
For tdIndex = <i>prachTd1</i> ... <i>prachTd2</i>			
	validRO	uint8_t	Validity of the PRACH RO opportunity corresponding to the fdIndex and tdIndex, as described in [3GPP TS 38.213 [22][4], section 8.1]. Range: <ol style="list-style-type: none">1. 0: not valid (in which case, no preambles from this RO are mapped to MsgA-PRUs)2. 1: valid (in which case, preambles from this RO may be mapped to valid MsgA-PRUs, and associated DMRS, as described in [3GPP TS 38.213 [22], section 8.1A]
	For groupIndex = <i>grpStart</i> ... <i>grpEnd</i>		Group A, B or other dedicated group(s). The PRU preamble map indexing is per MsgA-PUSCH configuration, as described in 3GPP TS 38.213 [22], section 8.1A, hence per PRACH preamble group.
	pruPreambleIndex-for-firstPreambleInGroup	uint16_t	Offset for the MsgA-PUSCH mapped preamble in msgA-Pusch-assocDmrs-Occasion[0] that corresponds to a <i>offsetPreambleFromstartPreambleIndex</i> . L2 ensures that this offset is consecutive to the MsgA-PUSCH mapped preamble index in the last valid MsgA-PUSCH & Associated DMRS resource. The Subsequent MsgA-PUSCH numPreamblesInGroup indices starting from this offset are mapped



Field	Type	Description
		<p>consecutively first to the preambles for the same msgA-Pusch-assocDmrs-Occasion[0], then for consecutively ordered msgA-Pusch-assocDmrs-Occasion[1], msgA-Pusch-assocDmrs-Occasion[2], etc.</p> <p>As described in [3GPP TS 38.213 [22], section 8.1A], there are $N_{\text{preambles}}$ mappings possible to each PRU and associated DMRS.</p> <p>Note: in alignment with [3GPP TS 38.213 [22], section 8.1A], an L2 implementation may choose to identify pruPreambleIndex = 0 as corresponding to the first preamble mapping to the first valid Msg-A PUSCH time and frequency domain opportunity, associated with the first DMRS resource, for the MsgA-PUSCH configuration.</p>
	<i>firstPreambleIndexInGroup</i>	uint8_t <p>The first preamble index in this group (cannot be less than <i>startPreambleIndex</i> for the PRACH occasion) for the MsgA-PRACH PDU that corresponds to pruPreambleIndex-for-firstPreambleInGroup in the MsgA-PUSCH preamble space.</p>
	numPusch-assocDmrs-Occasions-For-ThisPrachRO	uint8_t <p>#Pru-AssocDmrs: number of "valid PUSCH occasion and the associated DMRS", as defined in 38.213 [22], section 8.1A, to which this RO's RACH preambles map to.</p> <p>Set to 0 for invalid ROs, otherwise L2 ensures that enough MsgA-PUSCH and Associated Dmrs resource occasions signaled to map all of the preambles signaled in catAorB-Bitmap, for this PRACH occasion, except possibly the last one.</p>
	msgA-Pusch-assocDmrs-Occasion	uint16_t [#Pru-AssocDmrs] <p>Each entry is coded to correspond to the MsgA-PUSCH and Associated Dmrs resources as follows:</p> $\text{Fd_idx} + 16 * (\text{DmrsPort} + 16 * (\text{DmrsSeq} + 4 * (\text{Td_idx} + 8 * \text{Slot_idx}))),$ <ul style="list-style-type: none">• Fd_Idx: LSB 0:3• DmrsPort: LSB 4:7• DmrsSeq: LSB 8:9• Td_idx: LSB 10:12• Slot_idx: LSB 13:15 <p>The following ranges are assumed:</p> <ul style="list-style-type: none">• Fd_idx:



Field	Type	Description
		<ul style="list-style-type: none"> ○ 0...7: an individual MsgA-PUSCH index; ○ 15: all <i>numPuschOcasFd</i> frequency domain indices for the MsgA-PUSCH • DmrsPort: <ul style="list-style-type: none"> ○ 0..7: an individual DMRS port associated with the MsgA-PUSCH; ○ 15: all <i>numDmrsPorts</i> DMRS ports associated with the MsgA-PUSCH • DmrsSeq: <ul style="list-style-type: none"> ○ 0...1: an individual DMRS sequence associated with the MsgA-PUSCH; ○ 3: all <i>numDmrsSequences</i> DMRS sequences associated with the MsgA-PUSCH. • Td_idx: <ul style="list-style-type: none"> ○ 0...5: an individual MsgA-PUSCH time index ○ 7: all <i>numPuschOcasTd</i> time indices for the MsgA-PUSCH • T_slot: <ul style="list-style-type: none"> ○ 0..3: an individual MsgA-PUSCH slot index ○ 7: all <i>numPuschOcasSlots</i> slots indices for the MsgA-PUSCH <p>Entries are ordered consistently with 3GPP TS 38.213 [22], section 8.1A, to ensure that PRACH preambles map consecutively to the MsgA-PUSCH and associated DMRS occasions signaled in this array. If the wildcard "all" entries is used, the actual number of entries is determined from <i>numPuschOcasFd</i>, <i>numDmrsPorts</i>, <i>numDmrsSequences</i>, <i>numPuschOcasTd</i>, <i>numPuschOcasTd</i></p>

Table 3-44 MsgA-PRACH-to-(PRU & DMRS) mapping structure

Tag	Field	Type	Description
This table contains multiple MsgA-PUSCH configurations for the carrier.			
0x103F	numMsgA-Pusch-Configurations	uint16_t	Number of MsgA-PUSCH configurations supported in the carrier



Tag	Field	Type	Description
	numPrachConfigurations TLVs	Set of MsgA-PUSCH configuration TLVs (Table 3-43)	A set of numMsgA-Pusch-Configurations TLVs. Each PRACH configuration TLV shall contain a unique msgA-puschResConfigIndex between 0 and numMsgA-Pusch-Configurations-1.

Table 3-45 Multi-MsgA-PUSCH configuration table

Tag	Field	Type	Description
This table contains the configuration parameters used for storing an SSB transmission pattern (up to 64 SSB) in the PHY, which the PHY can then auto-generate. The <code>PARAM.response</code> message will have indicated which of these parameters is supported by the PHY and in which PHY states these parameters are mandatory or optional.			
The TLVs from this table signaled directly on <code>CONFIG.request</code> apply to SSB configuration index 0. See also the Multi-SSB Configuration Table 3-47..			
0x1033	ssbConfigIndex	uint16_t	A number uniquely identifying the SSB configuration. If not signaled, or signaled directly in <code>CONFIG.request</code> , it set to 0. Multi-SSB Configuration Table 3-47 is used if multiple configurations are needed.
	ssbOffsetPointA	uint16_t	Offset of lowest subcarrier of lowest resource block used for SS/PBCH block. Given in PRB [3GPP TS 38.211 [2], section 4.4.4.2] Value: 0->2199
	betaPssProfileNR	uint8_t	PSS EPRE to SSS EPRE in a SS/PBCH block [3GPP TS 38.213 [4], sec 4.1] Values: 0 = 0dB 1 = 3dB L1 uses to TLV if if configured with ProfileNR, or if PowerOffsetRsIndex bit#6 is set to 1
	betaPssProfileSSS	int16_t	Ratio of PSS EPRE to SSS EPRE. Values: -32767 -> -32767 representing -32.767dB to 32.767dB in 0.001dB L1 uses to TLV if if configured with ProfileNR and PowerOffsetRsIndex bit#6 is set to 0 3GPP reference: 3GPP TS 38.213 [4], section 4.1
	ssbPeriod	uint8_t	SSB periodicity in msec



Tag	Field	Type	Description
			Value: 0: ms5 1: ms10 2: ms20 3: ms40 4: ms80 5: ms160
	ssbSubcarrierOffset	uint8_t	ssbSubcarrierOffset or k_{SSB} (3GPP TS 38.211 [2], section 7.4.3.1) Value: 0->31
	Case	uint8_t	First symbol mapping for candidate SSB locations, per [3GPP TS 38.213 [4], sec 4.1] 0: Case A 1: Case B 2: Case C 3: Case D 4: Case E
	subcarrierSpacing	uint8_t	subcarrierSpacing [3GPP TS 38.211 [2], sec 4.2] of the SSB Value:0->4
	subCarrierSpacing Common	uint8_t	subcarrierSpacingCommon [3GPP TS 38.331 [6], sec 6.2.2], as broadcast in MIB. Value:0->3
	IMax	uint8_t	L_{max} , as defined in 3GPP TS 38.214 [5], section 7.4.1.4.1.
	ssbMask	uint32_t [2]	Bitmap for actually transmitted SSB. MSB->LSB of first 32 bit number corresponds to SSB 0 to SSB 31 MSB->LSB of second 32 bit number corresponds to SSB 32 to SSB 63 Value for each bit: <ul style="list-style-type: none">• 0: not transmitted• 1: transmitted Note that bits beyond the the first L_{max} are expected to be 0.
	beamIdPresence	uint8_t	Flag indicating whether the array of beam IDs below is present or not. Value: <ul style="list-style-type: none">• 0: absent• 1: present
	beamId	uint16_t [IMax]	Beam ID for each SSB in ssbMask. For example, if SSB mask bit 26 is set to 1, then beamId[26] will be used to indicate beam ID of SSB 26. Value: from 0 to 65535



Tag	Field	Type	Description
	rmsiPresence	uint8_t	<p>Indicates whether the SSB is cell-defining or not (i.e. is associated with an RMSI).</p> <p>0: RMSI absent 1: RMSI present (default).</p> <p>When signaled directly in CONFIG.request, value 0 is not allowed. RMSI may only be signaled as absent when signaled part of the Multi-SSB configuration (see Table 3-47).</p>

Table 3-46 SSB resource configuration table

Tag	Field	Type	Description
This table contains multiple SSB configurations for the carrier.			
0x1034	numSsbConfigurations	uint8_t	<p>Number of SSB configurations supported in the carrier (in addition to SSB configuration index 0)</p>
	ssbConfigurations TLVs	Set of SSB table TLVs (See Table 3-46)	<p>A set of numSsbConfigurations TLVs.</p> <p>Each TLV shall contain a unique SsbConfigIndex.</p> <p>Notes: these numSsbConfigurations TLVs are additional to SSB configuration index 0 – if any – signaled directly via Table 3-46 TLVs in CONFIG.request.</p>

Table 3-47 Multi-SSB resource configuration table

Tag	Field	Type	Description
This table contains the configuration parameters used for storing a TDD pattern in the PHY. The PARAM.response message will have indicated which of these parameters is supported by the PHY and in which PHY states these parameters are mandatory or optional.			
0x1035	TddPeriod	uint8_t	<p>DL UL Transmission Periodicity</p> <p>Value:</p> <ul style="list-style-type: none"> 0: ms0p5 1: ms0p625 2: ms1 3: ms1p25 4: ms2 5: ms2p5 6: ms5 7: ms10
	For MAX_TDD_PERIODICITY {		
	For MAX_NUM_OF_SYMBOL_PER_SLOT {		



Tag	Field		Type	Description
		SlotConfig	uint8_t	For each symbol in each slot a uint8_t value is provided indicating: 0: DL symbol 1: UL symbol 2: Flexible symbol
	}			
	}			

Table 3–48 TDD table

Tag	Field	Type	Description
This table contains the configuration parameters relating Measurement configuration. The PARAM.response message will have indicated which of these parameters is supported by the PHY and in which PHY states these parameters are mandatory or optional			
0x1028	rssi Measurement	uint8_t	RSSI measurement unit. See Table 3–23 for RSSI definition. Value: 0: Do not report RSSI 1: dBm 2: dBFS
0x1040	rsrp Measurement	uint8_t	RSRP measurement unit. See Table 3–23 for RSRP definition. Value: 0: Do not report RSRP 1: dBm 2: dBFS
0x1047	preamble Power Unit Selection	uint8_t	preamblePwr measurement unit. See TLV preamblePowerUnit in Table 3–23 for definition. Value: 0: Do not report preamblePwr 1: dBm 2: dBFS Note: If both units are supported, the default is dBFS
0x1042	srs Power Unit Selection	uint8_t	srsRsrp measurement unit. See TLV srsRsrpUnit in Table 3–23 for definition. Value: 0: Do not report srsRsrp 1: dBm 2: dBFS

Table 3–49 Measurement configuration

Tag	Field	Type	Description
This table contains the configuration parameters relating to digital Beamforming and Precoding.			
0x1043	Beamforming table	Table 3–66	The Digital Beamforming table (for DL precoding of logical ports into BB ports



Tag	Field	Type	Description
			and UL combining or BB ports into logical ports)
0x1044	Precoding table	Table 3-67	The Digital Precoding table (for precoding DL Layers into logical ports)

Table 3-50 Beamforming and Precoding

Tag	Field	Type	Description
This table contains the configuration parameters relating UCI indication.			
0x1036	num Uci2 Maps	uint16_t	Number of UCI maps indicating how a set of UCI part1 parameters map to a length of a corresponding UCI part2 Value: 0 ... maxNumUciMaps
for mapIndex = 0 ... numUci2Maps-1			the map Index
	num Part1 Params	unit8	1 ... 4 in this specification release
	sizes Part1 Params	unit8 [n]	n = numPart1Params $\Sigma(\text{sizesPart1Params}[*])$ shall not exceed 12.
	mapSize	uint8_t	mapSize = $\Sigma(\text{sizesPart1Params}[*])$ Value: 1 ... 12
	map	uint16 [m]	<p>m = mapSize Array indicating the uci Part2 size, as a function the part1 parameters.</p> <p>Interpretation: map[concatenate(param1Vals)]</p> <p>e.g.:</p> <ul style="list-style-type: none">- numPart1Params = 2- sizesPart1Params = [2,6] - determined from the part1 param offsets and sizes.- $\Sigma(\text{sizesPart1Params}) = 2+6 = 8$ bits.- map has $2^8 = 256$ entries of 16 bits each. <p>To determine the size of part2 for the following parameter values in part 1, the example below use makes use of highlighting to illustrate the concatenation:</p> <ul style="list-style-type: none">- p1Param[0] = 27 = 11b- p1Param[1] = 26 = 011010b



Tag	Field	Type	Description
			Size of part2 is determined as map[11011010b] = map[218]

Table 3-51 UCI configuration

Tag	Field	Type	Description
This table contains PRB-symbol rate-match patterns, for reference in PDSCH PDU (choice 'bitmap' of RateMatchPattern in 3GPP TS 38.331 [6], section 6.3.2)			
0x1037	numberOfPrbSymbRateMatchPatterns	uint8_t	Number of PRB-Symbol-based rate match patterns configured for this PHY
For each PrbSymbRateMatchPattern:			
	prbSymbRateMatchPatternID	uint8_t	0..255 can be signaled via a bitmap in PDSCH PDU.
	freqDomainRB	uint8_t [35]	275 LSB interpreted as resourceBlocks parameter in 3GPP TS 38.331 [6], RateMatchPattern IE definition
	symbolsInRB	uint32_t	LSB interpreted as oneSlot or twoSlots parameter in 3GPP TS 38.331 [6], RateMatchPattern IE definition
	subcarrierSpacing	uint8_t	Value 0, 1, 2, 3 for 15/30/60/120kHz respectively or 255 to indicate this is a BWP-level pattern and it uses the same SCS as the PDSCH PDU.

Table 3-52 PRB-symbol rate match patterns bitmap (non-CORESET) configuration

Tag	Field	Type	Description
This table contains CORESET rate-match patterns, for reference in PDSCH PDU (choice 'controlResourceSet' of RateMatchPattern in 3GPP TS 38.331 [6], section 6.3.2).			
0x1041	numberOfCoresetRateMatchPatterns	uint8_t	Number of Coreset-based rate match patterns configured for this PHY
For each CoresetRateMatchPattern:			
	coresetRateMatchPatternID	uint8_t	0..255 can be signaled via a bitmap in PDSCH PDU.
	freqDomainResources	uint8_t [6]	45 LSB interpreted same as frequencyDomainResources in ControlResourceSet of 3GPP TS 38.331 [6] LSB0 of at index 0 represents the first group of 6 PRBs
	symbolsPattern	uint16_t	Patterns of symbols to rate match around: 0: no rate match 1: rate match Derived to reflect slot-level gating of the search space associated with the Coreset, as described in controlResourceSet choice of the RateMatchPattern in 3GPP TS 38.331 [6], section 6.3.2
	subcarrierSpacing	uint8_t	Value 0, 1, 2, 3 for 15/30/60/120kHz respectively or 255 to indicate this is a



Tag	Field	Type	Description
			BWP-level pattern and it uses the same SCS as the PDSCH PDU.

Table 3-53 CORESET rate match patterns bitmap configuration

Tag	Field	Type	Description
This table contains LTE CRS rate-match patterns, for reference in PDSCH PDU			
0x1038	numberOfLteCRS RateMatchPatterns	uint8_t	Number of LTE CRS rate match patterns configured for this PHY
For each LteCRS RateMatchPattern:			
	crsRateMatchPatternID	uint8_t	0..255 can be signaled via a bitmap in PDSCH PDU.
	carrierFreqDL	uint16_t	Same interpretation as carrierFreqDL in RateMatchPatternLTE-CRS in 3GPP TS 38.331 [6]
	carrierBandwidthDL	uint8_t	Same interpretation as carrierBandwidthDL in RateMatchPatternLTE-CRS in 3GPP TS 38.331 [6]
	nrofCrsPorts	uint8_t	Same interpretation as nrofCRS-Ports in RateMatchPatternLTE-CRS in 3GPP TS 38.331 [6]: 1 = n1, 2 = n2, 4 = n4.
	vShift	uint8_t	Same interpretation as v-Shift in RateMatchPatternLTE-CRS in 3GPP TS 38.331 [6]: 0 = n0, 1 = n1, 2 = n2, 3 = n3, 4 = n4, 5 = n5.
	sizeMbsfnSubframeConfigList	uint8_t	range 0...8
For sizeMbsfnSubframeConfigList			
	radioframeAllocationPeriod	uint8_t	Same interpretation as radioframeAllocationPeriod in EUTRA-MBSFN-SubframeConfig in 3GPP TS 38.331 [6]: 1 = n1, 2 = n2, 4 = n4, 8 = n8, 16 = n16, 32 = n32.
	radioframeAllocationOffset	uint8_t	Same interpretation as radioframeAllocationOffset in EUTRA-MBSFN-SubframeConfig in 3GPP TS 38.331 [6]
	lteFrameStructureType	uint8_t	LTE frame structure type for MBSFN subframe allocation, Value 0: FDD Value 1: TDD
	subframeAllocLength	uint8_t	Number of frame and subframe defined as MBSFN subframe in DL Value 0: one_six Value 1: one_eight Value 2: four_twenty-four Value 3: four_thirty-two
	subframeAllocationBitmap	uint32_t	MBSFN subframe allocation pattern defined as bitmap, bitmap size is 32, exact valid bits depends on



Tag	Field		Type	Description
				<p>configuration of LteFrameStructureType and subframeAllocLength, If combination of above two parameters are as below, then the interpretation of this field shall follow below description accordingly.</p> <p>FDD + one_six: 6 LSB bits valid, each associated with FDD subframe 1,2,3,6,7,8 of one frames, starting from bit 0 map to SF1, bit 1 map to SF2, and so on;</p> <p>FDD + one_eight: 8 LSB bits valid, each associated with FDD subframe 1,2,3,4,6,7,8,9 of one frames, starting from bit 0 map to SF1, bit 1 map to SF2, and so on;</p> <p>FDD + four_twenty-four: 24 LSB bits valid, each 6 bits associated with FDD subframe 1,2,3,6,7,8 of one frame out of four consecutive frames, starting from bit 0 map to SF1 in first frame, bit 6 map to SF1 in second frame, and so on;</p> <p>FDD + four_thirty-two: 32 LSB bits valid, each associated with FDD subframe 1,2,3,4,6,7,8,9 of one frame out of four consecutive frames, starting from bit 0 map to SF1 in first frame, bit 8 map to SF1 in second frame, and so on;</p> <p>TDD + one_six: 5 LSB bits valid, each associated with TDD subframe 3,4,7,8,9 of one frames, starting from bit 0 map to SF3, bit 1 map to SF4, and so on;</p> <p>TDD + one_eight: does not apply to TDD, L2/L3 should avoid using this combination for DSS with TDD-LTE, and PHY should ignore it otherwise;</p> <p>TDD + four_twenty-four: 20 LSB bits valid, each 5 bits associated with TDD subframe 3,4,7,8,9 of one frame out of four consecutive frames, starting from bit 0 map to SF3 in first frame, bit 5 map to SF3 in second frame, and so on;</p> <p>TDD + four_thirty-two: does not apply to TDD, L2/L3 should avoid using this combination for DSS with TDD-LTE, and PHY should ignore it otherwise.</p>

Table 3-54 LTE-CRS rate match patterns configuration



Tag	Field	Type	Description
This table contains the configuration parameters relating to semi-static signaling of PUCCH parameters linked to UL BWP ID			
0x1039	numUlBwpIds	uint8_t	Number of UL BWP IDs to which PUCCH semistatic parameters are linked
	for ulBwpId = 0 ... numUlBwpIds-1		
	pucchGroupHopping	uint8_t	See the definition and range of of pucchGroupHopping in PUCCH PDU (Table 3-111)
	nIdPucchHopping	uint16_t	<p>Cell-specific scrambling ID for PUCCH group hopping and sequence hopping in this UL BWP. Corresponds nID in 3GPP TS 38.211 [2], sec 6.3.2.2.1.</p> <p>For example, when MAC is implemented according to 3GPP TS 38.211 [2] Rel-15 and higher layer(RRC) does not include hoppingId IE for UE, MAC must ensure value of this field aligns with physical cell ID, i.e., L1 always assume the value is aligned with that of used by UE.</p> <p>Valid for formats 0, 1, 3 and 4. Value: 0->1023</p>

Table 3-55 PUCCH semi-static configuration

Tag	Field	Type	Description
This table contains the configuration parameters relating to semi-static PDSCH parameters.			
0x103A	pdsch Cbg Scheme	uint8_t	<p>Value: 0 = No DL CBG Re-tx 1 = DL CBG Re-tx with CBG segmentation in L2 2 = DL CBF Re-Tx, with segmentation in L1</p>

Table 3-56 PDSCH semi-static configuration

Tag	Field	Type	Description
This table contains configuration delay management parameters defined by reference to [10], or reserved configuration tag ranges			
0x011E	Timing window	uint16_t	<p>See section 3.2.9 of [10]</p> <p>Applies to Delay Management with or without Timestamps, whichever is configured.</p>
0x011F	Timing mode	uint8_t	See section 3.2.9 of [10]



Tag	Field	Type	Description
			Applies to Timing Info or Timing Indication APIs, depending on whether Delay Management with or without Timestamps is configured
0x0120	Timing period	uint8_t	<p>See section 3.2.9 of [10]</p> <p>Applies to Timing Info or Timing Indication APIs, depending on whether Delay Management with or without Timestamps is configured</p>
0x1045	DL_TTI Rx Window for SSB	uint8_t	<p>Configures the anchoring of the RxWindow for SSB PDUs received in DL_TTI.request (if PHY supports Delay Management)</p> <p>Value:</p> <ul style="list-style-type: none"> 1 – SSB Numerology 2 – Highest PDSCH numerology, if lower than the SSB numerology
0x1046	UL_TTI Rx Window for PRACH	uint8_t	<p>Configures the anchoring of the RxWindow for short PRACH PDUs received in UL_TTI.request (if PHY supports Delay Management)</p> <p>Value:</p> <ul style="list-style-type: none"> 1 – short PRACH Numerology 2 – highest PUSCH numerology, if lower than the short PRACH numerology
0x0F00	<reserved>	N/A	See section 3.2.2 of [10]

Table 3–57 Delay management configuration

Tag	Field	Type	Description
This table contains configuration pertinent to Rel-16 mTRP (3GPP TS 38.300 [11], section 6.12) configuration			
0x103B	numTxPortsTRP1	uint8_t	<p>[Rel-16] Number of DL baseband ports corresponding to the first TRP, for a PHY hosting baseband for both TRPs a Rel-16 mTRP scheme:</p> <p>Value</p> <ul style="list-style-type: none"> 0: only hosting one TRP (and all DL baseband ports are assigned to it). > 0: number of baseband ports assigned to TRP1 (in which case TRP2 is assigned numTxAnt-numTxPortsTRP1 antennas)
0x103C	numRxPortsTRP1	uint8_t	<p>[Rel-16] Number of UL baseband ports corresponding to the first TRP, for a PHY hosting baseband for both TRPs a Rel-16 mTRP scheme:</p> <p>Value</p> <ul style="list-style-type: none"> 0: only hosting one TRP (and all UL baseband ports are assigned to it). > 0: number of baseband ports assigned to TRP1 (in which case TRP2 is assigned numRxAnt-numRxPortsTRP1 antennas)

Table 3–58 Rel-16 mTRP configuration



3.3.3 Vendor TLVs

The range 0xA000 – 0xFFFF are reserved for vendor-specific TLVs. These TLVs can be either CONFIG or PARAM TLVs.

3.3.4 START

The START message exchange was described in section 2.1.1.3.

3.3.4.1 START.request

This message can be sent by the L2/L3 when the PHY is in the CONFIGURED state. If it is sent when the PHY is in the IDLE, or RUNNING state an ERROR.indication message will be sent by the PHY. No message body is defined for START.request. The message length in the generic header = 0.

3.3.4.2 START errors

The error codes returned in an ERROR.indication generated by the START.request message are given in Table 3-59.

Error code	Description
MSG_INVALID_STATE	The START.request was received when the PHY was in the IDLE or RUNNING state.
MSG_INVALID_PHY_ID	The PHY ID is not defined
MSG_UNINSTANTIATED_PHY	The PHY ID is not instantiated
MSG_INVALID_DFE_Profile	No valid DFE Profile exists for the PHY ID
FEU Not Initialized	DFE or RF chains associated with the PHY are not initialized
DFE Profile Not Selected	No DFE profile was selected for associationg DFE and RF chains with the PHY

Table 3-59 Error codes for START.request

3.3.4.3 START.response

When PHY supports both nFAPI message format and Delay Management, successful initialization requires a START.response procedure, as documented in section 3.1.9 of SCF-225 [10].

3.3.5 STOP

The STOP message exchange was described in Figure 2-6.

3.3.5.1 STOP.request

This message can be sent by the L2/L3 when the PHY is in the RUNNING state. If it is sent when the PHY is in the IDLE, or CONFIGURED, state an ERROR.indication message will be sent by the PHY. No message body is defined for STOP.request. The message length in the generic header = 0.



3.3.5.2 STOP. Indication

This message is sent by the PHY to indicate that it has successfully stopped and returned to the CONFIGURED state. No message body is defined for STOP.indication. The message length in the generic header = 0.

3.3.5.3 STOP errors

The error codes returned in an ERROR.indication generated by the STOP.request message are given in Table 3-60.

Error code	Description
MSG_INVALID_STATE	The STOP.request was received when the PHY was in the IDLE or CONFIGURED state.

Table 3-60 Error codes for STOP.request

3.3.6 PHY Notifications

The PHY notification messages are used by the PHY to inform the L2/L3 software of an event which occurred.

3.3.6.1 ERROR.indication

This message is used to report an error to the L2/L3 software. These errors all relate to API message exchanges. The format of ERROR.indication is given in Table 3-61.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
Message ID	uint8_t	Indicate which message received by the PHY has an error. Values taken from Table 3-5.
Error code	uint8_t	The error code, see Section 3.3.6.2 for values
Expected SFN	uint16_t	If the error code is OUT_OF_SYNC, this field is set to the expected SFN, otherwise it is set to 0xFFFF Value: 0 -> 1023 or 0xFFFF
Expected Slot	uint16_t	If the error code is OUT_OF_SYNC, this field is set to the expected Slot, otherwise it is set to 0xFFFF Value: 0 -> 159 or 0xFFFF
Extended Status	Table 3-62	Field signaling an extended status;

Table 3-61 ERROR.indication message body



Field	Type	Description
Extended Status Length	uint16_t	xlen = Length of the extended status payload Value: 0 ... 65535
Extended Status Contents	uint8_t [xlen]	The extended status payload. Content shall depend on the Error Code in ERROR.indication (Table 3-61)

Table 3-62 Extended Status format

3.3.6.2 Error Codes

The list of possible error codes returned in either .response messages or the ERROR.indication message is given in Table 3-63

Value	Error code	Description
0x0	MSG_OK	Message is OK.
0x1	MSG_INVALID_STATE	The received message is not valid in the PHY's current state.
0x2	MSG_INVALID_CONFIG	The configuration provided in the request message was invalid
0x3	OUT_OF_SYNC	The DL_TTI.request was received with a different SFN or slot than the PHY expected.
0x4	MSG_SLOT_ERR	The DL_TTI.request or UL_TTI.request had an invalid format.
0x5	MSG_BCH_MISSING	A SSB PDU was expected in the DL_TTI.request message for this subframe. However, it was not present.
0x6	MSG_INVALID_SFN	The received UL_DCI.request or TX_DATA.request message included a SFN/SL value which was not expected. The message has been ignored.
0x7	MSG_UL_DCI_ERR	The UL_DCI.request had an invalid format
0x8	MSG_TX_ERR	The TX_DATA.request had an invalid format.
0x9	MSG_INVALID_PHY_ID	The PHY ID is not defined
0xA	MSG_UNINSTANTIATED_PHY	The PHY ID is not instantiated
0xB	MSG_INVALID_DFE_Profile	No valid DFE Profile exists for the PHY ID
0xC	PHY PROFILE_SELECTION_INCOMPATIBLE_WITH_RUNNING_PHY	There is a RUNNING PHY Id whose definition would be changed by the PHY Profile selection
0xD	FEU_NOT_INITIALIZED	DFE or RF chains associated with the PHY are not initialized
0xE	DFE PROFILE_NOT_SELECTED	No DFE profile was selected for associating DFE and RF chains with the PHY
0xFF	MSG_EXTENDED_VENDOR_STATUS	In this release, this value shall be used to signal vendor-specific extended status codes in ERROR.indication

Table 3-63 PHY API error codes



Field	Type	Description
Extended Status Length	uint16_t	xlen = Length of the extended status payload Value: 0 ... 65535
Extended Status Contents	uint8_t [xlen]	The extended status payload. Content shall depend on the Error Code in ERROR.indication (Table 3-61)

Table 3-64 Extended Status format

3.3.6.3 CONNECTIVITY.indication

This message is used to report an connectivity event L2/L3 software regarding the PHY. It may be scoped for a particular PHY, in which case the observation applies to that PHY, or may be scoped for the common context, in which case the observation applies to any actual or potential PHY. If an observation applies to the common context, it is also raised – if supported – for all dedicated PHYs (and only applies to PHY-side observations).

The format of CONNECTIVITY.indication is given in Table 3-65.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
Observation	Uint16_t	An indication of the connectivity status: 0: Connection lost 1: Synchronization lost 2: Synchronization gained

Table 3-65 CONNECTIVITY.indication message body

3.3.7 Storing Precoding and Beamforming Tables

The format of the beamforming table is given in Table 3-66 and the format of the precoding table is given in Table 3-67.

Field	Type	Description	
numDigBeams	uint16_t	Number of digital beams Value: 0->65535	
numBasebandPorts	uint16_t	Number of ports at the digital beamformer output. Value: 0->65535	
for each digBeam {			
	beamIdx	uint16_t	Identifying number for the beam index Value: 0->65535
for each Baseband port {			
	digBeamWeightRe	int16_t	Real part of the digital beam weight



Field		Type	Description
	digBeamWeightIm	int16_t	Imag part of the digital beam weight
	}		
	}		

Table 3–66 Digital beam table (DBT) PDU

Field		Type	Description
PMidx		uint16_t	Index which uniquely identifies the precoding matrix (PM). PMidx = 0 is reserved for identity matrix. Value: 0->65535
numLayers		uint16_t	Number of ports at the precoder input Value: 0->65535
numAntPorts		uint16_t	Number of logical antenna ports at the precoder output Value: 0->65535
For numLayers {			
For numAntPorts {			
	precoderWeightRe	int16_t	Real part of the precoder weight
	precoderWeightIm	int16_t	Imag part of the precoder weight
	}		
	}		

Table 3–67 Precoding matrix (PM) PDU

3.3.8 RESET

The RESET message exchange was described in section 2.1.4.

3.3.8.1 RESET.request

This message can be sent by the L2/L3 when the PHY is in the RUNNING or CONFIGURED state. If it is sent when the PHY is in the IDLE, state an ERROR.indication message will be sent by the PHY. No message body is defined for RESET.request. The message length in the generic header = 0.

3.3.8.2 RESET.Indication

This message is sent by the PHY to indicate that it has successfully stopped and returned to the IDLE state. No message body is defined for RESET.indication. The message length in the generic header = 0.

3.3.8.3 RESET errors

The error codes returned in an ERROR.indication generated by the RESET.request message are given in Table 3–68.



Error code	Description
MSG_INVALID_STATE	The RESET.request was received when the PHY was in the IDLE state.

Table 3–68 Error codes for RESET.request

3.4 Slot Messages

The slot messages are used by the L2/L3 software to control the data transmitted, or received, every slot.

3.4.1 Slot.indication

The SLOT.indication message is given in Table 3–69, and is sent from the PHY with a periodicity based on the CONFIG.request message sent to the PHY. Specifically, the CONFIG.request message includes carrier configuration TLVs, which provide values for different numerologies, including options to indicate a numerology is not used. The SLOT.indication message is sent from the PHY based on the highest-value numerology configured in CONFIG.request. If the highest numerology configured is:

- 0 – then SLOT.indication is provided every 1ms
- 1 – then SLOT.indication is provided every 500us
- 2 – then SLOT.indication is provided every 250us
- 3 – then SLOT.indication is provided every 125us
- 4 – then SLOT.indication is provided every 62.5us

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159

Table 3–69 Slot indication message body

3.4.1a TIMING.indication

The TIMING.indication message is given in Table 3–70. It supports Delay Management without Timestamps, as described in section 2.2.0.2, when configured to the PHY.

It is sent from the PHY on an event-triggered basis, or with a periodicity in slots (for the highest configured numerology of this PHY) based on the CONFIG.request message sent to the PHY. It is also sent after a successful START.request. The TIMING.indication message is sent from the PHY based on the highest-value numerology configured in CONFIG.request.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023



Field	Type	Description
Slot	uint16_t	Slot Value: 0 -> 159
numObservations	uint8_t	Value 0 ... 255
For 0 ... numObservations - 1		
messageId	uint16_t	16-bit Message Type ID to which the observation applies. For PHY messages, the possible messages are: <ul style="list-style-type: none">• DL_TTI.request• UL_TTI.request• UL_DCI.request• TX_DATA.request For FEU messages, the possible messages are: <ul style="list-style-type: none">• DFE SCHEDULE.request• RF SCHEDULE.request• FEU.CONFIG.RF SCHEDULE.request• FEU.CONFIG. SELECT_BEAM.REQUEST• FEU.CONFIG. SET_BEAM_SLOT_PATTERN.REQUEST
observation	uint8_t	Value: <ul style="list-style-type: none">• 0: Too-late (invalid timeFromRxWindowStart)• 1: Too-early (invalid time report)• 2: Too-late (valid time report)• 3: Too-early (valid time report)• 8: periodic or initial report• 4-7, 9-255: reserved.
time From Rx Window Start	uint32_t	Absolute value of time difference between message arrival as observed by the PHY (if supported), and the Rx window start, for the message in messageId. Value: <ul style="list-style-type: none">• 0 ... (10^8): time in nanoseconds measured from the start of the Rx window for the message, if such a metric is supported. (Note that accuracy is not necessarily in nanoseconds).• 10^8 ... 2^32-1: reserved

Table 3-70 Timing indication message body

3.4.2 DL_TTI.request

The format of the `DL_TTI.request` message is shown in Table 3-71. A `DL_TTI.request` message indicates the SFN/Slot it contains information for, and this control information is for a downlink slot.



This message can be sent by the L2/L3 when the PHY is in the RUNNING state. If it is sent when the PHY is in the IDLE or CONFIGURED state an `ERROR.indication` message will be sent by the PHY.

- A PDCCH PDU includes 1 or more DCI PDUs

The PDUs included in this structure are signaled in causal order: if PDU A references PDU B, PDU B appears before PDU A in the "Number of PDUs" loop.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
nPDUs	uint16_t	Number of PDUs that are included in this message. All PDUs in the message are numbered in order. Value 0 -> 65535 Backward compatibility note: the field was 8 bits in SCF222 v2.0
nDITypes	uint8_t	Maximum number of DL PDU types or DCIs supported by <code>DL_TTI.request</code> . <code>nDIPduTypes</code> = 5 in this release. Backward compatibility note: SCF222 v2.0 did not have this field
nPDUsOfEachType	uint16_t [nDITypes]	Number of PDUs of each type that are included in this message. Each array entry corresponds to a PDU type as follows: [0]: number of PDCCH PDUs [1]: number of PDSCH PDUs [2]: number of CSI-RS PDUs [3]: number of SSB PDUs [4]: number of DDCIs across all PDCCH PDUs in this message [5]: number of PRS PDUs Value 0 -> 65535, for each entry Backward compatibility note: SCF222 v2.0 did not have this field
nGroup	uint16_t	Number of PDU Groups included in this message. Value 0 -> 3,822
Top-level Rate Match Patterns	struct	See Table 3-72
For Number of PDUs {		



Field	Type	Description
PDUType	uint16_t	0: PDCCH PDU, see Section 3.4.2.1 1: PDSCH PDU, see Section 3.4.2.13.4.2.2 2: CSI-RS PDU, see Section 3.4.2.3 3: SSB PDU, see Section 3.4.2.4 5: PRS PDU, see Section 3.4.2.4a
PDUSize	uint16_t	Size of the PDU control information (in bytes). This length value includes the 4 bytes required for the PDU type and PDU size parameters. Value 0 -> 65535
DL PDU Configuration	structure	See Sections 3.4.2.1 to 3.4.2.4
}		
<p>In this message, if nGroup > 0 then for each group {</p> <p>-- For FAPIv4, groups represent MU MIMO groups, as described in section 2.2.6.2; muMimoGroups are listed in increasing order of their start symbol. muMimogroups starting at the same symbol, are listed in increasing order of their first PRB index.</p>		
nPdus	uint8_t	Number of PDUs (in this release, PDU types in each group are identical) in this group For SU-MIMO, one group includes one PDU only. For MU-MIMO, one group includes up to 32 PDUs. For each PDU type, each PRB in a symbol can be referenced by at most one MuMIMO group. Value 1 -> 32
prgSize	uint16_t	This is just for reducing the layer mapping table. The loop below is only over PRGs which have at least one PRB in prbBitmap. For FAPIv2 grouping: <ul style="list-style-type: none">• prgSize is set to 0• nPdus is limited to 1...12• prbBitmap, symbolBitmap fields are omitted.• the PRG loop below is omitted. Value: 1 → 275 (FAPIv4), or 0 (to mean FAPIv2)
prbBitmap	uint8_t [36]	bitmap of PRBs assumed common to all UEs in the group, for joint spatial precoding. 273 rounded up to multiple of 32. This bitmap is in units of PRBs. LSB of byte 0 of the bitmap represents the PRB 0.
nPRGs	uint16_t	Number of PRGs with at least one PRB allocated in prbBitmap
symbolBitmap	uint16_t	bitmap of symbols for the joint spatial precoding group bit location indicates the symbol index, up to 14 symbols. First symbol corresponds to location #0
For each of the nPDUs in the group		



Field		Type	Description
	PduIdx	uint16_t	<p>This value is an index for number of PDU identified by nPDU in this message</p> <p>Value: 0 -> 65535</p>
	For each of the nPRG PRGs{		The PrgIdx referred to in the Tx Generalized precoding PDU is the index of the PRG in this loop, starting from 0. Note that only PRGs with allocations in the prbBitmap are included in this loop.
	chosenLayer BitmapForCW0	uint8_t	All layers scheduled for CW0 of this PDU. Bit indices are in in the same order as the S-matrix singular values for the layers.
	chosenLayer BitmapForCW1	uint8_t	All layers scheduled for CW1 this PDU. Bit indices are in in the same order as the S-matrix singular values for the layers.
	}		
	}		

Table 3-71 DL_TTI.request message body

Field	Type	Description
Structure for signaling set of all rate match structures. The signaling can be by reference to P5 configuration, or directly embedded in this structure, depending in PHY capability and MAC choice.		
Each rate match pattern signaled in this structure shall be reference in at least one PDSCH PDU in the same <code>DL_TTI.request</code> , otherwise the structure is irrelevant.		
For PHYs that signal support for universal rate match (per TLV <code>universalRateMatch</code>), MAC ensures that all PDSCH PDUs that overlap with any of the patterns in this structure rate match around that pattern.		
<code>DL_TTI.request</code> Top-level: Prb Symbol Rate Match		
numPrbSymRmPatterns	uint16_t	Number of PrbSym rate match patterns signalled at the <code>DL_TTI</code> top level
prbSymbRm ConfigurationMethod	uint8_t	<p>Indicates whether the configuration for the PrbSym rate match patterns is in P5, or in P7: Values:</p> <ul style="list-style-type: none"> • 0: P5-based configuration. Then: <ul style="list-style-type: none"> ◦ <code>numPrbSymbRmP5Ref</code> = numPrbSymRmPatterns ◦ <code>numPrbSymbP7Sig</code> = 0 • 1: P7-based configuration. Then: <ul style="list-style-type: none"> ◦ <code>numPrbSymbRmP5Ref</code> = 0 ◦ <code>numPrbSymbRmP7Sig</code> = numPrbSymRmPatterns <p>Note: value 0 is only relevant for PHYs that signal support for universal rate match (per TLV <code>universalRateMatch</code>)</p>
prbSymRMPatternP5Bitmap	uint8_t	<code>szPsP5</code> = $\text{ceil}(\text{numPrbSymbRmP5Ref} / 8)$



Field	Type	Description
	[szPsP5]	<p>Each bit maps to one PrbSymb-based Rate Match Pattern, bit set to 1 represents rate matching for this PrbSymb -based RMP resources, bit set to 0 represents no rate matching for this PrbSymb -based RMP resources.</p> <p>Note: this field should only be signaled for PHYs expecting universal rate match (per TLV universalRateMatch)</p>
prbSymbRmP7Loop: For 0 ... numPrbSymbRmP7Sig-1		loop index is the identifier of the PrbSymb-based rate match pattern, when referenced in PDSCH PDU
subcarrierSpacing	uint8_t	Value 0, 1, 2, 3 for 15/30/60/120kHz respectively or 255 to indicate this is a BWP-level pattern and it uses the same SCS as the PDSCH PDU.
freqDomainRB	uint8_t [35]	275 LSB interpreted as resourceBlocks parameter in 3GPP TS 38.331 [6], RateMatchPattern IE definition. LSB0 of at index 0 represents PRB 0
symbolsInRB	uint16_t	Patterns of symbols to rate match around: 0: no rate match 1: rate match
DL_TTI.request Top-level: CORESET-based Rate Match		
numCoresetRmPatterns	uint16_t	Number of CORESET-based rate match patterns signalled at the DL_TTI top level
coresetRmConfigurationMethod	uint8_t	<p>Indicates whether the configuration for the CORESET-based rate match patterns is in P5, or in P7:</p> <p>Values:</p> <ul style="list-style-type: none"> • 0: P5-based configuration. Then: <ul style="list-style-type: none"> ◦ numCoresetRmP5Ref = numCoresetRmPatterns ◦ numCoresetRmP7Sig = 0 • 1: P7-based configuration. Then: <ul style="list-style-type: none"> ◦ numCoresetRmP5Ref = 0 ◦ numCoresetRmP7Sig = numCoresetRmPatterns <p>Note: value 0 is only relevant for PHYs that signal support for universal rate match (per TLV universalRateMatch)</p>
coresetRmPattern P5Bitmap	uint8_t [szCsP5]	<p>szCsP5 = ceil(numCoresetRmP5Ref / 8)</p> <p>Each bit maps to one CORESET-based Rate Match Pattern, bit set to 1 represents rate matching for this CORESET-based RMP resources, bit set to 0 represents no rate matching for this CORESET-based RMP resources.</p>



Field	Type	Description
		Note: this field should only be signaled for PHYs expecting universal rate match (per TLV universalRateMatch)
coresetRmP7Loop: For 0 ... numCoresetRmP7Sig-1		loop index is the identifier of the CORESET-based rate match pattern, when referenced in PDSCH PDU
subcarrierSpacing	uint8_t	Value 0, 1, 2, 3 for 15/30/60/120kHz respectively or 255 to indicate this is a BWP-level pattern and it uses the same SCS as the PDSCH PDU.
freqDomainResources	uint8_t [6]	45 LSB interpreted same as frequencyDomainResources in ControlResourceSet of 3GPP TS 38.331 [6] LSB0 of at index 0 represents the first group of 6 PRBs
symbolsInRB	uint16_t	Patterns of symbols to rate match around: 0: no rate match 1: rate match
DL_TTI.request Top-level: LTE CRS-based Rate Match		
numLteCrsRmPatterns	uint16_t	Number of LTE CRS-based rate match patterns signalled at the DL_TTI top level
IteCrsRmConfigurationMethod	uint8_t	<p>Indicates whether the configuration for the LTE CRS-based rate match patterns is in P5, or in P7: Values:</p> <ul style="list-style-type: none"> • 0: P5-based configuration. Then: <ul style="list-style-type: none"> ◦ numLteCrsRmP5Ref = numLteCrsRmPatterns ◦ numLteCrsRmP7Sig = 0 • 1: P7-based configuration. Then: <ul style="list-style-type: none"> ◦ numLteCrsRmP5Ref = 0 ◦ numLteCrsRmP7Sig = numLteCrsRmPatterns <p>Note: value 0 is only relevant for PHYs that signal support for universal rate match (per TLV universalRateMatch)</p>
IteCrsRmPattern P5Bitmap	uint8_t [szLcP5]	<p>szLcP5 = ceil(numLteCrsRmP5Ref / 8)</p> <p>Each bit maps to one LTE CRS-based Rate Match Pattern, bit set to 1 represents rate matching for this LTE CRS-based RMP resources, bit set to 0 represents no rate matching for this LTE CRS-based RMP resources.</p> <p>Note: this field should only be signaled for PHYs expecting universal rate match (per TLV universalRateMatch)</p>
IteCrsRmP7Loop For 0 ... numLteCrsRmP7Sig-1		loop index is the identifier of the LTE CRS-based rate match pattern, when referenced in PDSCH PDU



Field	Type	Description
carrierFreqDL	uint16_t	Same interpretation as carrierFreqDL in RateMatchPatternLTE-CRS in 3GPP TS 38.331 [6]
carrierBandwidthDL	uint8_t	Same interpretation as carrierBandwidthDL in RateMatchPatternLTE-CRS in 3GPP TS 38.331 [6]
nrofCrsPorts	uint8_t	Same interpretation as nrofCRS-Ports in RateMatchPatternLTE-CRS in 3GPP TS 38.331 [6]: 1 = n1, 2 = n2, 4 = n4.
vShift	uint8_t	Same interpretation as v-Shift in RateMatchPatternLTE-CRS in 3GPP TS 38.331 [6]: 0 = n0, 1 = n1, 2 = n2, 3 = n3, 4 = n4, 5 = n5.
sizeMbsfnSubframeConfigList	uint8_t	range 0...8 (L2 shall set this to 0, if L1 only supports P7-based MBSFN subframe determiniantion)
For sizeMbsfnSubframeConfigList		
radioframeAllocationPeriod	uint8_t	Same description as radioframeAllocationPeriod in Table 3-54
radioframeAllocationOffset	uint8_t	Same description as radioframeAllocationOffset in Table 3-54
IteFrameStructureType	uint8_t	Same description as IteFrameStructureType in Table 3-54
subframeAllocLength	uint8_t	Same description as subframeAllocLength in Table 3-54
subframeAllocationBitmap	uint32_t	Same description as subframeAllocationBitmap in Table 3-54

Table 3-72 DL_TTI.request top-level rate match signaling

3.4.2.1 PDCCH PDU

Each DL DCI PDU includes the information that the L2/L3 software must provide the PHY so it can transmit the DCI described in [3GPP TS 38.212 [3], section 7.3.1], however, the L2/L3 constructs the DCI message.

Each CORESET is related to a specific BWP and there can be more than 1 CORESET per BWP. If there is more than 1 CORESET per BWP then multiple PDCCH PDUs should be included in the `DL_TTI.request`.

Field	Type	Description
BWP [3GPP TS 38.213 [4], sec 12]		
CoresetBWPSIZE	uint16_t	Bandwidth part size [3GPP TS 38.213 [4], sec12]. Number of contiguous PRBs allocated to the BWP Note: CoresetBWPStart and CoresetBWPSIZE are expected to correspond to BWP signaled in RRC Value: 1->275
CoresetBWPStart	uint16_t	bandwidth part start RB index from reference CRB [3GPP TS 38.213 [4], sec 12]



Field	Type	Description
		<p>Note: CoresetBWPStart and CoresetBWPSIZE are expected to correspond to BWP signaled in RRC</p> <p>Value: 0->274</p>
SubcarrierSpacing	uint8_t	<p>subcarrierSpacing [3GPP TS 38.211 [2], sec 4.2]</p> <p>Value: 0->4</p>
CyclicPrefix	uint8_t	<p>Cyclic prefix type [3GPP TS 38.211 [2], sec 4.2]</p> <p>0: Normal; 1: Extended</p>
Coreset [3GPP TS 38.211 [2], sec 7.3.2.2]		
StartSymbolIndex	uint8_t	<p>Starting OFDM symbol for the CORESET</p> <p>Value: 0->13</p>
DurationSymbols	uint8_t	<p>Contiguous time duration of the CORESET in number of symbols. Corresponds to L1 parameter $N_{symb}^{CORESET}$ [3GPP TS 38.211 [2], sec 7.3.2.2]</p> <p>Value: 1,2,3</p>
FreqDomainResource	uint8_t [6]	<p>Frequency domain resources. This is a bitmap defining non-overlapping groups of 6 PRBs in ascending order. [3GPP TS 38.213 [4], section 10.1]. Also, corresponds to L1 parameter $N_{RB}^{CORESET}$ [3GPP TS 38.211 [2], sec 7.3.2.2]</p> <p>Bitmap of uint8 array. 45 bits.</p> <p>$FreqDomainResource[0]$ designates LSByte of RRC parameter frequencyDomainResources, and the LSBit of $FreqDomainResource[0]$ carries frequencyDomainResources[0]</p> <p>Note:</p> <ul style="list-style-type: none">- If this CORESET is defined by <i>ControlResourceSet</i> IE for a UE, L2/L3 signals this parameter in a way consistent with <i>frequencyDomainResources</i> IE for the <i>ControlResourceSet</i> IE for the BWP, per 38.331 [6], section 6.3.2- If this CORESET is defined by <i>controlResourceSetZero</i> IE for a UE, L2/L3 signals this parameter in a way consistent with the RBs indicated by <i>controlResourceSetZero</i> IE for BWP#0, by setting CoresetBWPStart and CoresetBWPSIZE to start point and size of CORESET 0 directly, per 3GPP TS 38.213 [4], section 13. For this case, freqDomainResource always have bit value 1 at start of bitmap.
CceRegMappingType	uint8_t	<p>CORESET-CCE-to-REG-mapping-type [3GPP TS 38.211 [2], sec 7.3.2.2]</p> <p>0: non-interleaved 1: interleaved</p>



Field	Type	Description
RegBundleSize	uint8_t	<p>The number of REGs in a bundle. Must be 6 for cceRegMappingType = nonInterleaved. For cceRegMappingType = interleaved, must belong to {2,6} if duration = 1,2 and must belong to {3,6} if duration = 3. Corresponds to parameter L. [3GPP TS 38.211 [2], sec 7.3.2.2]</p> <p>Value: 2,3,6</p>
InterleaverSize	uint8_t	<p>The interleaver size. For interleaved mapping belongs to {2,3,6} and for non-interleaved mapping is NA. Corresponds to parameter R. [3GPP TS 38.211 [2], sec 7.3.2.2]</p> <p>Value: 2,3,6</p>
CoreSetType	uint8_t	<p>[3GPP TS 38.211 [2], sec 7.3.2.2 and sec 7.4.1.3.2]</p> <p>0: CORESET is configured by the PBCH or SIB1 (subcarrier 0 of the CORESET)</p> <p>1: otherwise (subcarrier 0 of CRB0 for DMRS mapping)</p>
ShiftIndex	uint16_t	<p>[3GPP TS 38.211 [2], sec 7.3.2.2]</p> <p>Not applicable for non-interleaved mapping.</p> <p>For interleaved mapping and a PDCCH transmitted in a CORESET configured by the PBCH or SIB1 this should be set to phy cell ID.</p> <p>Value: 10 bits</p> <p>Otherwise, for interleaved mapping this is set to 0-> max num of PRBs.</p> <p>Value 0-> 275</p>
precoderGranularity	uint8_t	<p>Granularity of precoding [3GPP TS 38.211 [2], sec 7.3.2.2]</p> <p>0: sameAsRegBundle</p> <p>1: allContiguousRBs</p>
numDIDci	uint16_t	<p>Number of DCIs in this CORESET.</p> <p>Value: 0->MaxDciPerSlot</p>
For number of DL DCIs {		
DIDci	structure	See Table 3-74
}		
Pdcch Maintenance Parameters added in FAPIv3	structure	See Table 3-75
PDCCH PDU parameters FAPIv4	structure	See Table 3-76

Table 3-73 PDCCH PDU



Field	Type	Description
RNTI	uint16_t	The RNTI used for identifying the UE when receiving the PDU Value: 1 -> 65535.
nIdPdcchData	uint16_t	Parameter n _{ID} used for PDCCH Data scrambling in [3GPP TS 38.211 [2], sec 7.3.2.3]. Value: 0->65535
nRntiPdcchData	uint16_t	Parameter nRNTI used for PDCCH data scrambling, in [3GPP TS 38.211 [2], sec 7.3.2.3] Value: 0 -> 65535
CceIndex	uint8_t	CCE start Index used to send the DCI Value: 0->135
AggregationLevel	uint8_t	Aggregation level used [3GPP TS 38.211 [2], sec 7.3.2.1] Value: 1,2,4,8,16
Beamforming info		
Precoding and Beamforming	structure	See Table 3-93. If the PHY supports Rel-16 see also Table 3-94. For PHYs supporting FAPIv4 MU-MIMO groups, see also Table 3-95.
Tx Power info		
beta_PDCCH_1_0 [deprecated]	uint8_t	In this FAPI release, for ProfileNR, L2/L3 signals PDCCH power via powerControlOffsetSSProfileNR (next field), regardless of DCI format. PHY may ignore this beta_PDCCH_1_0.
powerControlOffsetSS ProfileNR	int8_t	PDCCH power value used for all PDCCH Formats. This is ratio of PDCCH and PDCCH DMRS EPRE to SSB/PBCH block EPRE 3GPP TS 38.214 [5] 3GPP Spec reference (informative): - RRC parameter <i>powerControlOffsetSS</i> : NZP CSI-RS EPRES to SSB/PBCH block EPRE is given in [3GPP TS 38.214 [5], sec 5.2.2.3.1] and PDCCH EPRE to NZP CSI-RS EPRE is – in cases of link recovery – assumed as 0 dB [3GPP TS 38.214 [5], section 4.1]. “UE may assume that the ratio of PDCCH DMRS EPRE to SSS EPRE is within -8 dB and 8 dB when the UE monitors PDCCHs for a DCI format 1_0 with CRC scrambled by SI-RNTI, P-RNTI, or RA-RNTI.” [3GPP TS 38.213 [4], section 4.1] Value range: -8 ... 8 representing -8 to 8 dB in 1dB steps -127: L1 is configure with ProfileSSS
PayloadSizeBits	uint16_t	The total DCI length (in bits) including padding bits [3GPP TS 38.212 [3], sec 7.3.1] Range 0-> DCI_PAYLOAD_BYTLEN*8



Field	Type	Description
Payload	uint8_t[DCI_PAYLOAD_BT YE_LEN]	DCI payload, where the actual size is defined by PayloadSizeBits. The bit order is as following bit0-bit7 are mapped to first byte of MSB - LSB

Table 3-74 DL DCI PDU

Field	Type	Description
Set of Coreset and DCI parameters added in FAPIv3		
Coreset parameter		
pdcchPduIndex	uint16_t	Unique across all PDCCH PDUs in the slot.
<obsolete in FAPIv4>	uint16_t	Obsolete in FAPIv4
DCI-specific parameters		
For number of DL DCIs {		The same number of DL DCIs as
dciIndex	uint16_t	Unique across all DCIs in this PDU Value: 0→65535
collocatedAl16Candidate	unit8_t	true if an AL-16 candidate exists at the same CCE Index, which UE expects to rate match around, per 3GPP TS 38.214 [5], section 5.1.4.1
pdcchDmrsPowerOffset ProfileSSS	int16_t	Ratio of PDCCH DMRS EPRE to SSS EPRE Values: -32767 → -32767 representing -32.767dB to 32.767dB in 0.001dB steps -32768: ProfileNR is in use
pdcchDataPowerOffset ProfileSSS	int16_t	Ratio of PDCCH Data EPRE to SSS EPRE Values: -32767 → -32767 representing -32.767dB to 32.767dB in 0.001dB steps -32768: L1 determines PDCCH data power from the PDCCH dmrs power.
}		

Table 3-75 PDCCH PDU maintenance FAPIv3

Field	Type	Description
Set of Coreset and DCI parameters added in FAPIv4		
DCI-specific parameters		
For number of DL DCIs {		
nIdPdcchDmrs	uint16_t	Parameter N _{ID} used for PDCCH DMRS scrambling in [3GPP TS 38.211 [2], sec 7.4.1.3.1]. Value: 0->65535
}		
MU-MIMO support in FAPIv4		
num Spatial Stream Indices	uint16_t	The same number of DL DCIs, or zero.



Field	Type	Description
		Must match the number of DL DCIs, if PHY indicates that it requires DL spatial stream index assignment, via capability maxNumberDISpatialStreams (TLV 0x0151), may be set to zero otherwise.
For numSpatialStreamIndices DL DCIs {		this loop is present if and only if numSpatialStreamIndices > 0
dciIndex	uint16_t	Unique across all DCIs in this PDU
spatialStreamIndex	uint16_t	Spatial Stream index for mapping of PDCCH, based on PHY capability indicating need for spatial stream index signaling Value: 0 → (max # spatial streams - 1, per TLV 0x0151)
}		

Table 3-76 PDCCH PDU parameters FAPIv4

3.4.2.2 PDSCH PDU

The format of the PDSCH PDU is shown in Table 3-77.

The PDSCH PDU includes both mandatory and optional parameters, where the presence of the optional elements is defined in the parameter pduBitmap. The presence of these optional elements is included based on the following:

- pdschPtrs and pdschPtrsV3 are included if and only if PTRS are included in the downlink transmission
- cbgRetxCtrl is included if CBG is supported and retransmit is used.

(It should be noted that some parameters are only applicable for certain modes/types etc. These are still included but their value is ignored by the PHY.)

Field	Type	Description
pduBitmap	uint16_t	Bitmap indicating presence of optional PDUs Bit 0: pdschPtrs and pdschPtrsV3 - Indicates PTRS included Bit 1:cbgRetxCtrl (Present when CBG based retransmit is used.) All other bits reserved
RNTI	uint16_t	The RNTI used for identifying the UE when receiving the PDU Value: 1 -> 65535
pduIndex	uint16_t	PDU index incremented for each PDSCH PDU sent in TX control message. This is used to associate control information to data and is reset every slot. Value: 0 -> 65535
BWP [3GPP TS 38.213 [4], section 12]		



Field	Type	Description	
BWPSize	uint16_t	<p>If L1 uses pdschTransType, this field signals the bandwidth part size [3GPP TS 38.213 [4], sec 12]. Number of contiguous PRBs allocated to the BWP</p> <p>If L1 does not use pdschTransType, this field represents the number of VRBs to which PRBs are mapped for the allocation, according to 3GPP TS 38.211 [2], section 7.3.1.6.</p> <p>Note: The two interpretations align, except when the allocation is scheduled by DCI format 1_0 in CSS.</p> <p>Value: 1->275</p>	
BWPStart	uint16_t	<p>If L1 uses pdschTransType, this field signals the bandwidth part start RB index from reference CRB [3GPP TS 38.213 [4], sec 12]</p> <p>If L1 does not use pdschTransType, this field represents the CRB corresponding to VRB0, according to 3GPP TS 38.211 [2], section 7.3.1.6.</p> <p>Note: The two interpretations align, except when the allocation is scheduled by DCI format 1_0 in CSS.</p> <p>Value: 0->274</p>	
SubcarrierSpacing	uint8_t	<p>subcarrierSpacing [3GPP TS 38.211 [2], sec 4.2]</p> <p>Value:0->4</p>	
CyclicPrefix	uint8_t	<p>Cyclic prefix type [3GPP TS 38.211 [2], sec 4.2]</p> <p>0: Normal; 1: Extended</p>	
Codeword information			
NrOfCodewords	uint8_t	<p>Number of code words for this RNTI (UE)</p> <p>Value: 1 -> 2</p>	
For each codeword {			
	targetCodeRate	uint16_t	<p>Target coding rate [3GPP TS 38.212 [3], sec 5.4.2.1 and 3GPP TS 38.214 [5], sec 5.1.3.1]. This is the number of information bits per 1024 coded bits expressed in 0.1 bit units</p>
	qamModOrder	uint8_t	<p>QAM modulation [3GPP TS 38.212 [3], sec 5.4.2.1 and 3GPP TS 38.214 [5], sec 5.1.3.1]</p> <p>Value: 2,4,6,8</p>
	mcsIndex	uint8_t	<p>MCS index [3GPP TS 38.214 [5], sec 5.1.3.1], should match value sent in DCI</p> <p>Value : 0->31</p>
	mcsTable	uint8_t	<p>MCS-Table-PDSCH [3GPP TS 38.214 [5], sec 5.1.3.1]</p> <p>0: notqam256</p>



Field		Type	Description
			1: qam256 2: qam64LowSE
	rvIndex	uint8_t	Redundancy version index [3GPP TS 38.212 [3], Table 5.4.2.1-2 and 3GPP TS 38.214 [5], Table 5.1.2.1-2], should match value sent in DCI Value : 0->3
	TBSize	uint32_t	Transport block size (in bytes) [3GPP TS 38.214 [5], sec 5.1.3.2]
}			
nIdPdsch	uint16_t	Parameter n_{ID} from [3GPP TS 38.211 [2], sec 7.3.1.1] Value: 0->1023	
nrOfLayers	uint8_t	Number of layers [3GPP TS 38.211 [2], sec 7.3.1.3] Value : 1->8	
transmissionScheme	uint8_t	PDSCH transmission schemes [3GPP TS 38.214 [5], sec 5.1.1] 0: Up to 8 transmission layers	
refPoint	uint8_t	Reference point for PDSCH DMRS "k" - used for tone mapping [3GPP TS 38.211 [2], sec 7.4.1.1.2] Resource block bundles [3GPP TS 38.211 [2], sec 7.3.1.6] Value: 0 -> 1 If 0, the 0 reference point for PDSCH DMRS is at Point A [3GPP TS 38.211 [2], sec 4.4.4.2]. Resource block bundles generated per sub-bullets 2 and 3 in [3GPP TS 38.211 [2], sec 7.3.1.6]. For sub-bullet 2, the start of bandwidth part must be set to the start of actual bandwidth part + $NstartCORESET$ and the bandwidth of the bandwidth part must be set to the bandwidth of the initial bandwidth part. If 1, the DMRS reference point is for PDSCH DM-RS is subcarrier 0 of the lowest numbered resource block bandwidth part which given by the field BWPStart. L2/L3 shall only set this value to 1 for PDSCH PDU carrying payload of RMSI. L2/L3 shall ensure that BWP start corresponds to start of a CORESET#0 Resource block bundles generated per sub-bullets 1 [3GPP TS 38.211 [2], sec 7.3.1.6]	



Field	Type	Description
DMRS [3GPP TS 38.211 [2], sec 7.4.1.1]		
dIDmrsSymbPos	uint16_t	<p>DMRS symbol positions [3GPP TS 38.211 [2], sec 7.4.1.1.2 and Tables 7.4.1.1.2-3 and 7.4.1.1.2-4]</p> <p>Bitmap occupying the 14 LSBs with:</p> <p>bit 0: first symbol and for each bit 0: no DMRS 1: DMRS</p>
dmrsConfigType	uint8_t	<p>DL DMRS config type [3GPP TS 38.211 [2], sec 7.4.1.1.2]</p> <p>0: type 1 1: type 2</p>
pdschDmrsScramblingId	uint16_t	<p>PDSCH DMRS-Scrambling-ID [3GPP TS 38.211 [2], sec 7.4.1.1.1] as provided by parameter n_{ID}^{SCID}.</p> <p>For example, in reference [12], this field applies to below case:</p> <ol style="list-style-type: none">1. PDSCH transmission with Rel-15 DM-RS configuration2. PDSCH transmission having DM-RS port(s) within CDM group 0 or 2 with Rel-16 DM-RS configuration <p>Value: 0->65535</p>
pdschDmrsScramblingIdComplement	uint16_t	<p>PDSCH DL-DMRS-Scrambling-ID [3GPP TS 38.211 [2], sec 7.4.1.1.1 7.4.1.1.2] as provided by parameter $n_{ID}^{\bar{n}_{SCID}}$ (Rel-16) when $\bar{n}_{SCID} \neq n_{SCID}$</p> <p>PHY shall disregard this parameter if lowPaprDmrs = 0</p> <p>For example, in reference [12], this is the scrambling ID associated with CDM group 1 for PDSCH DM-RS sequence generation and only valid if DM-RS port(s) of the CDM group indicated by antenna ports field in DCI</p> <p>Value: 0->65535</p>
lowPaprDmrs	uint8_t	<p>1: \bar{n}_{SCID} depends on DM-RS CDM group</p> <p>0: $\bar{n}_{SCID} = n_{SCID}$ (i.e. independent of DM-RS CDM group)</p>
nSCID	uint8_t	<p>DMRS sequence initialization [3GPP TS 38.211 [2], sec 7.4.1.1.2]], as provided by parameter n_{SCID}.</p> <p>For example, in reference [12], this value is associated with DM-RS scrambling ID given by dIDmrsScramblingId</p> <p>Value : 0->1</p>



Field	Type	Description
numDmrsCdmGrpsNo Data	uint8_t	<p>Number of DM-RS CDM groups without data [3GPP TS 38.212 [3], sec 7.3.1.2.2] [3GPP TS 38.214 [5], Table 4.1-1] it determines the ratio of PDSCH EPRE to DM-RS EPRE.</p> <p>Value: 1->3</p>
dmrsPorts	uint16_t	<p>DMRS ports. [3GPP TS 38.212 [3], 7.3.1.2.2] provides description between DCI 1-1 content and DMRS ports.</p> <p>Bitmap occupying the 12 LSBs with:</p> <p>bit 0: antenna port 1000 bit 11: antenna port 1011 and for each bit 0: DMRS port not used 1: DMRS port used</p>
Pdsch Allocation in frequency domain [3GPP TS 38.214 [5], sec 5.1.2.2]		
resourceAlloc	uint8_t	<p>Resource Allocation Type [3GPP TS 38.214 [5], sec 5.1.2.2] 0: Type 0 1: Type 1</p>
rbBitmap	uint8_t [36]	<p>For resource alloc type 0. 3GPP TS 38.212 [3], section 7.3.1.2.2 bitmap of RBs, 273 rounded up to multiple of 32. This bitmap is in units of VRBs. LSB of byte 0 of the bitmap represents the VRB 0, per section 7.3.1.6 of 3GPP TS 38.211 [2]</p>
rbStart	uint16_t	<p>For resource allocation type 1. [3GPP TS 38.214 [5], sec 5.1.2.2.2] The starting resource block corresponding to VRB 0 for this PDSCH, per section 7.3.1.6 of 3GPP TS 38.211 [2].</p> <p>Value: 0->274</p>
rbSize	uint16_t	<p>For resource allocation type 1. [3GPP TS 38.214 [5], sec 5.1.2.2.2] The number of resource blocks within for this PDSCH.</p> <p>Value: 1->275</p>
VRBtoPRBMapping	uint8_t	<p>VRB-to-PRB-mapping [3GPP TS 38.211 [2], sec 7.3.1.6] 0: non-interleaved 1: interleaved with RB size 2 2: Interleaved with RB size 4</p>
Resource Allocation in time domain [3GPP TS 38.214 [5], sec 5.1.2.1]		



Field	Type	Description
StartSymbolIndex	uint8_t	Start symbol index of PDSCH mapping from the start of the slot, S. [3GPP TS 38.214 [5], Table 5.1.2.1-1] Value: 0->13
NrOfSymbols	uint8_t	PDSCH duration in symbols, L [3GPP TS 38.214 [5], Table 5.1.2.1-1] Value: 1->14
PTRS Parameters (existing in FAPIv2)		
pdschPtrs	structure	See Table 3-78; this structure is included if and only if pduBitmap[0] = 1. It shall only apply to the first PTRS port.
Beamforming		
Precoding and Beamforming	structure	See Table 3-93. If the PHY supports Rel-16 see also Table 3-94. For PHYs supporting FAPIv4 MU-MIMO groups, see also Table 3-95.
Tx Power info		
powerControlOffsetProfileNR	uint8_t	Ratio of PDSCH EPRE to NZP CSI-RSEPRE [3GPP TS 38.214 [5], sec 5.2.2.3.1] Values: 0->23: representing -8 to 15 dB in 1dB steps 255: L1 is configured with ProfileSSS
powerControlOffsetSSProfileNR	uint8_t	Ratio of NZP CSI-RS EPRE to SSB/PBCH block EPRE to [3GPP TS 38.214 [5], sec 5.2.2.3.1] Values: 0: -3dB, 1: 0dB, 2: 3dB, 3: 6dB 255: L1 is configured with ProfileSSS
cbgReTxCtrl: CBG fields for Segmentation in L2. This structure is only included if and only if (pduBitmap[1] = 1 and L1 is configured to support L2 CBG segmentation)		
IsLastCbPresent	uint8_t	Indicates whether the last code block is presented in the CBG re-transmission or not. Bitmap of length 8. LSB 0 is signaled for 1st TB and LSB1 for 2nd TB respectively. LSB 0: 0: Last code block is not presented for 1st TB (e.g. last CB is not present in the CBG-based re-Tx or 1st TB is initial Tx.) 1: Last code block is presented for 1st TB LSB 1:



Field	Type	Description
		<p>0: Last code block is not presented for 2nd TB (e.g. last CB is not present in the CBG-based re-Tx or 2nd TB is initial Tx.).</p> <p>1: Last code block is presented for 2nd TB</p> <p>If CBG Re-Tx includes last CB, L1 will add the TB CRC to the last CB in the payload before it is read into the LDPC HW unit</p>
isInlineTbCrc	uint8_t	<p>Indicates whether TB CRC is part of data payload or control message</p> <p>0: TB CRC is part of data payload</p> <p>1: TB CRC is part of control message</p> <p>This field applies to any and all CWs for which CRC is required, per IsLastCbPresent</p>
dITbCrcCW	uint32_t [2]	<p>TB CRC: to be used in the last CB, applicable only if last CB is present</p> <p>dITbCrcCW[0]: CRC for 1st CW</p> <p>dITbCrcCW[1]: CRC for 2nd CW</p> <p>Crc is defined the same way as in DL_TTI.response</p>
PDSCH Maintenance Parameters added in FAPIv3	structure	See Table 3-79
pdschPtrsV3: PDSCH PTRS Maintenance Parameters added in FAPIv3	structure	See Table 3-80 this structure is included if and only if pduBitmap[0] = 1
Rel-16 PDSCH Parameters added in FAPIv3	structure	See Table 3-81
PDSCH Parameters added in FAPIv4	structure	See Table 3-82
PDSCH Parameters added in FAPIv5	structure	See Table 3-83

Table 3-77 PDSCH PDU

Field	Type	Description
PTRS [3GPP TS 38.214 [5], sec 5.1.6.3]		
PTRSPortIndex	uint8_t	<p>PT-RS antenna ports [3GPP TS 38.214 [5], sec 5.1.6.3] [3GPP TS 38.211 [2], table 7.4.1.2.2-1]</p> <p>Bitmap occupying the 6 LSBs with:</p> <p>bit 0: antenna port 1000</p> <p>bit 5: antenna port 1005</p> <p>and for each bit</p> <p>0: PTRS port not used</p> <p>1: PTRS port used</p>



Field	Type	Description
PTRSTimeDensity	uint8_t	PT-RS time density [3GPP TS 38.214 [5], table 5.1.6.3-1] 0: 1 1: 2 2: 4
PTRSFreqDensity	uint8_t	PT-RS frequency density [3GPP TS 38.214 [5], table 5.1.6.3-2] 0: 2 1: 4
PTRSReOffset	uint8_t	PT-RS resource element offset [3GPP TS 38.211 [2], table 7.4.1.2.2-1] Value: 0->3
nEpreRatioOfPDSCHToPTRSProfileNR	uint8_t	PT-RS-to-PDSCH EPRE ratio [3GPP TS 38.214 [5], table 4.1-2] Value : 0->3: index into table 4.1-2 of 3GPP TS 38.214 [5] 255: DL-PTRS uses SSS for power offset

Table 3-78 PDSCH PTRS parameters

Field	Type	Description
Set of PDSCH PDU parameters added in FAPIv3		
BWP information		
pdschTransType	uint8_t	L1 interprets this field as below, if so configured by pdschTransTypeValidity=1, otherwise it ignores it. Indication used to indicate the transmission type of PDSCH PDU. Value 0: non-interleaved PDSCH which is scheduled by DCI format 1_0 in a common search space Value 1: any non-interleaved PDSCH except above case Value 2: interleaved PDSCH which is scheduled by PDCCH DCI format 1_0 in Type0-PDCCH common search space in CORESET 0. If pdschTransType set to 2, then BWP start and size of the PDSCH PDU shall be set to CORESET 0 start and size respectively instead of active downlink BWP start and size. Value 3: interleaved PDSCH which is scheduled by PDCCH DCI format 1_0 in any common search space other than Type0-PDCCH common search space in CORESET 0 when CORESET 0 is configured for the cell Value 4: interleaved PDSCH which is scheduled by PDCCH DCI format 1_0 in any common search space other than Type0-



Field	Type	Description
		PDCCH common search space in CORESET 0 when CORESET 0 is not configured for the cell Value 5: any interleaved PDSCH which not fall into above 3 interleaved PDSCH categories.
coresetStartPoint	uint16_t	L1 interprets this field as below, if so configured by pdschTransTypeValidity=1, otherwise it ignores it. The PRB index of lowest-numbered RB in the CORESET in which PDCCH carrying scheduling info for the PDSCH PDU is received. Refer to 3GPP TS 38.211 [2], section 7.3.1.6 $N_{start}^{CORESET}$. Only valid when pdschTransType field set to 0, 3, 4 Value: 0 to 274
initialDIBwpSize	uint16_t	L1 interprets this field as below, if so configured by pdschTransTypeValidity=1, otherwise it ignores it. The size of initial downlink BWP used for the cell, shall be set to size of CORESET 0 if CORESET 0 is configured for the cell, and initial downlink BWP otherwise. Refer to TS 3GPP 38.211 [2], section 7.3.1.6 $N_{BWP,init}^{size}$. Only valid when pdschTransType field set to 3, 4 value 0 to 274
Codeword information		
ldpcBaseGraph	uint8_t	LDPC base graph to use for CW generation [3GPP TS 38.212 [3], sec 7.2.2]. Values: 1: LDPC base graph 1 2: LDPC base graph 2 (other values are reserved)
tbSizeLbrmBytes	uint32_t	Parameter TBS_{LBRM} from 3GPP TS 38.212 [3], section 5.4.2.1, for computing the size of the circular buffer. value 0 is reserved.
tbCrcRequired	uint8_t	Indicates whether L1 should report TB CRC for the PDSCH PDU's TB(s). Bitmap of 8 length and LSB 0 and 1 are valid. LSB 0: 0: TB CRC of 1 st TB not required 1: TB CRC of 1 st TB required LSB 1: 0: TB CRC of 2 nd TB not required 1: TB CRC of 2 nd TB required CRC report, if any shall be performed via DL_TTI.response (see section 3.4.2b).
Rate Matching references		



Field	Type	Description	
ssbPdusForRateMatching	uint16_t [8]	<p>the SSB PDU indices in the slot, whose RBs are not available for PDSCH allocation.</p> <p>Value</p> <p>< 256 an SSB PDU index 65534: rate matching is indicated only via ssb Configuration Index 65535: no SSB PDU to rate match around, in this position, for this PDSCH allocation;</p> <p>Notes: There can be up to eight 240 kHz SSBs to be rate matched around by a PDSCH PDU allocation with SCS 60 kHz.</p>	
ssbConfigForRateMatching	uint16_t	<p>the SSB PDU indices in the slot, whose RBs are not available for PDSCH allocation.</p> <p>Value</p> <p>< 256 a SSB configuration index. 65534: rate matching is indicated only via ssb PDU Index 65535: no SSB rate matching for this PDSCH allocation</p>	
prbSymRmPatternBitmapSizeByReference	uint8	<p>size of by-reference bit map of Rm Patterns, in bits (set to 0 if L1 is not configured with by-reference rate match patterns.)</p> <p>The size of the bitmap is determined as szRm = ceil(prbSymRmPatternBitmapSizeByReference / 8)</p>	
prbSymRateMatchPatternBitmapByReference	uint8_t [szRm]	<p>each bit maps to one RB-based Rate Match Pattern, bit set to 1 represents rate matching for this RB-based RMP resources, bit set to 0 represents no rate matching for this RB-based RMP resources.</p> <p>If prbSymbRmConfigurationMethod is absent or indicates P5-based references, then this bitmap refers to indices in P5 TLV 0x0137 (see Table 3-52), otherwise it refers to the loop indices in prbSymbP7Loop for this DL_TTI.request.</p>	
numPrbSymRmPatternsByValue	uint8	<p>At most 8 patterns may be signaled (set to 0 if L1 is not configured with by-value rate match patterns.)</p> <p>Shall be set to 0 in this release</p>	
For numPrbSymRmPatternsByValue			
	freqDomainRB	uint8_t [35]	275 LSB interpreted as resourceBlocks parameter in 3GPP TS 38.331 [6], RateMatchPattern IE definition. LSB0 of at index 0 represents PRB 0
	symbolsInRB	uint16_t	Patterns of symbols to rate match around: 0: no rate match 1: rate match



Field	Type	Description
numCoresetRmPatterns	uint8	<p>At most 8 patterns may be signaled (set to 0 if L1 is configured with by-reference rate match patterns.)</p> <p>Shall be set to 0 in this release</p>
For numCoresetRmPatterns		
freqDomainResources	uint8_t [6]	45 LSB interpreted same as frequencyDomainResources in ControlResourceSet of 3GPP TS 38.331 [6] LSB0 of at index 0 represents the first group of 6 PRBs
symbolsPattern	uint16_t	Patterns of symbols to rate match around: 0: no rate match 1: rate match
pdcchPduIndex	uint16_t	PDCCH PDU index for rate matching Set to 65535 if not applicable.
dciIndex	uint16_t	DCI index in the PDCCH PDU indexm for rate matching Set to 65535 if not applicable.
IteCrsRateMatchPattern BitmapSize	uint8_t	<p>Size of bitmap signaling the LTE CRS pattern(s) to apply.</p> <p>Rel-15: a single pattern per UE Cf. 3GPP TS 38.331 [6], section 6.2.2</p> <p>Rel-16: multiple patterns supported per UE Cf. 3GPP TS 38.331 [13], section 6.2.2</p> <p>The of the bitmap array is determined as szCrs = ceil(IteCrsRateMatchPatternBitmapSize / 8)</p>
IteCrsRateMatchPattern	uint8_t [szCrs]	<p>Each bit map to one CRS Rate Match Pattern, bit set to 1 represents rate matching for this CRS pattern, bit set to 0 means no rate matching for this CRS pattern.</p> <p>If IteCrsRmConfigurationMethod is absent or indicates P5-based references, then this bitmap refers to indices in P5 TLV 0x0138 (Table 3-54), otherwise it refers to the loop indices in IteCrsP7Loop for this DL_TTI.request.</p>
numCsiRsForRateMatching	uint8_t	Number of CSI-RS PDUs that signal REs to rate match around for this PDSCH PDU allocation
csiRsForRateMatching	uint16_t [szCsiRs]	Array of szCsiRs = numCsiRsForRateMatching CSI-RS PDU indices to rate match around
Tx Power info		
pdschDmrsPowerOffset ProfileSSS	int16_t	<p>Ratio of PDSCH DMRS EPRE to SSS EPRE Values:</p> <ul style="list-style-type: none"> -32767 -> -32767 representing -32.767dB to 32.767dB in 0.001dB steps -32768: ProfileNR is in use



Field	Type	Description
pdschDataPowerOffset ProfileSSS	int16_t	<p>Ratio of PDSCH DMRS EPRE to SSS EPRE Values: -32767 -> -32767 representing -32.767dB to 32.767dB in 0.001dB steps -32768: L1 determines PDSCH data power from PDSCH dmrs power 3GPP reference: 3GPP TS 38.214 [5], section 4.1</p>
cbgReTxCtrlL2Seg: CBG fields for Segmentation in L2. This structure is only included if and only if pduBitmap[1] = 1 and L1 is configured to support L2 CBG segmentation)		
cbgReTxCtrlL1Seg: CBG fields for Segmentation in L1 This structure is only included if and only if pduBitmap[1] = 1 and L1 is configured to support L1 CBG segmentation)		
maxNumCbgPerTb	uint8_t	<p>Valid bits in the CbgTxInformation field per TB for associated PDSCH payload Value 2,4,6,8 For single TB/CW case, maximum value could be 8; For double TB/CW case, maximum value restricted to 4;</p>
For each codeword {		
cbgTxInformation	uint8_t	<p>Bitmap with size of 8 bits used to indicate which CBG(s) is/are (re)transmitted. Actual valid bits depends on above parameter maxNumCbgPerTb and NrOfCodewords. LSB Bit 0 corresponding to first CBG in the TB, bit 1 corresponding to second CBG in the TB and so on.</p>

Table 3-79 PDSCH maintenance parameters V3

Field	Type	Description
PDSCH-PTRS parameter added in FAPIv3		
Tx Power info		
pdschPtrsPowerOffset ProfileSSS	int16_t	<p>Ratio of PDSCH PT-RS EPRE to SSS EPRE Values: -32767 -> -32767 representing -32.767dB to 32.767dB in 0.001dB steps -32768: L1 determines PT-RS data power from PDSCH dmrs power 3GPP reference: 3GPP TS 38.214 [5], section 4.1</p>

Table 3-80 PDSCH PTRS maintenance parameters V3

Field	Type	Description
Rel-16 parameters added in FAPIv3		
repetitionScheme	uint8_t	Signals the Rel-16 single-DCI mTRP repetition scheme.



Field	Type	Description
		<p>Description: If any fdmSchemeB scheme is signaled:</p> <ul style="list-style-type: none"> - NrOfCodewords = 2 - Each codeword uses nrOfLayers/2 layers, per the mapping in 3GPP TS 38.214 [5], section 7.3.1 - Per CW RV determination is as described in 3GPP TS 38.214 [5], section 5.1.2.3. - Per CW PRGs allocation is as described in 3GPP TS 38.214 [5], section 5.1.2.3. <p>Values:</p> <ul style="list-style-type: none"> 1: fdmSchemeB-Prg2 2: fdmSchemeB-Prg4 3: fdmSchemeB-PrgWideband 255: repetition scheme not relevant [other values reserved]
PTRS Parameters this section is included if and only if pduBitmap[0] = 1		
pdschPtrs2	structure	<p>See Table 3-78.</p> <p>If the PDSCH is setup to use two PTRS ports, as described in 3GPP TS 38.214 [5], section 5.1.6.4, then PTRSPortIndex in pdschPtrs may be non-zero, otherwise it is all-zero</p>
pdschPtrsV3: PDSCH PTRS in FAPIv3	structure	<p>See Table 3-80 this structure is included if and only if pduBitmap[0] = 1 and applies to pdschPtrs2</p>

Table 3-81 Rel-16 PDSCH parameters V3

Field	Type	Description
coresetRmPatternBitmapSizeByReference	uint8	<p>size of by-reference bit map of CORESET-based Rm Patterns, in bits (set to 0 if L1 is not configured with by-reference coresets RM patterns.)</p> <p>The size of the bitmap is determined as $szRm = \text{ceil}(\text{coresetRmPatternBitmapSizeByReference} / 8)$</p>
coresetRmPatternBitmapByReference	uint8_t [szRm]	<p>each bit maps to one CORESET-based Rate Match Pattern, bit set to 1 represents rate matching for this Coreset-based RMP resources, bit set to 0 represents no rate matching for this Coreset-based RMP resource.</p> <p>If coresitRmConfigurationMethod is absent or indicates P5-based references, then this bitmap refers to indices in P5 TLV 0x1041 (see Table 3-53), otherwise</p>



Field	Type	Description
		it refers to the loop indices in coresetRmP7Loop for this DL_TTI.request.
IteCrsMbsfnDerivationMethod	uint8_t	Signals how L1 should derive whether a LTE-CRS pattern overlaps should be MBSFN or non-MBSFN. Value: <ul style="list-style-type: none">• 0: Based on P5 signaling (in this case, szCrs-Mbsfn = 0)• 1: Based on IteCrsMbsfnPattern (in this case, szCrs-Mbsfn = szCrs)
MU-MIMO support in FAPIv4		
num Spatial Stream Indices	uint8_t	Set to the number codewords, or zero. Must be set to the number codewords if PHY indicates that it requires DL spatial stream index assignment, via capability maxNumberDISpatialStreams (TLV 0x0151), otherwise it may be set to 0. 0 = false 1 = true
For each CW 0 ... numSpatialStreamIndices -1)		This for loop is present iff numSpatialStreamIndices > 0
numLayersForCW	uint8_t	Number of layers mapping to the CW for the loop.
spatialStreamIndicesForCW	uint16_t [#layers for CW]	Spatial Stream indices for mapping of PDSCH layers, based on PHY capability indicating need for layer index signaling. For reciprocity-based MU-MIMO, these spatial streams correspond to CW layers for each PRG bitmap, in the same order



Field	Type	Description
		<p>as they appear in the chosenLayersBitmap, for each PRG. E.g. chosenLayerBitmapForCW0 may take value 00000011_b for PRG-A and 00010100_b for PRG-B. In this example:</p> <ul style="list-style-type: none"> - CW0 spatial stream#0 (index spatialStreamIndicesForCW[0]) uses layer#0 over PRG-A and layer#2 for PRG-B - CW0 spatial stream#1 (index spatialStreamIndicesForCW[1]) uses layer#1 over PRG-A and layer#4 for PRG-B <p>For semistatic precoding, the order of the indices corresponds to index of the layer dimension in the Precoding Matrix for the Precoding and Beamforming PDU.</p> <p>Value: 1 → (max # spatial streams, per TLV 0x0151)</p>

Table 3-82 PDSCH parameters FAPIv4

Field	Type	Description
Rel-16 parameters added in FAPIv5		
numPrsForPuncturing	uint8_t	Number of PRS PDUs that signal REs to puncture for this PDSCH PDU allocation
prsForPuncturing	uint16_t [szPrs]	Array of szPrs = numPrsForPuncturing PRS PDU indices to rate match around

Table 3-83 PDSCH parameters FAPIv5

3.4.2.3 CSI-RS PDU

The format of the CSI-RS PDU is shown in Table 3-84.

Field	Type	Description
BWP [3GPP TS 38.213 [4], sec 12]		
SubcarrierSpacing	uint8_t	subcarrierSpacing [3GPP TS 38.211 [2], sec 4.2] Value: 0->4
CyclicPrefix	uint8_t	Cyclic prefix type [3GPP TS 38.211 [2], sec 4.2] 0: Normal; 1: Extended
StartRB	uint16_t	PRB where this CSI resource starts related to common resource block #0 (CRB#0). Only multiples of 4 are allowed. [3GPP TS 38.331 [6], sec 6.3.2 parameter CSI-FrequencyOccupation] Value: 0 ->274
NrOfRBs	uint16_t	Number of PRBs across which this CSI resource spans. Only multiples of 4 are allowed. [3GPP TS 38.331 [6], sec 6.3.2 parameter CSI-FrequencyOccupation]



Field	Type	Description
		Value: 24 -> 276
CSIType	uint8_t	CSI Type [3GPP TS 38.211 [2], sec 7.4.1.5] Value: 0:TRS 1:CSI-RS NZP 2:CSI-RS ZP
Row	uint8_t	Row entry into the CSI Resource location table. [3GPP TS 38.211 [2], sec 7.4.1.5.3 and table 7.4.1.5.3-1] Value: 1-18
FreqDomain	uint16_t	Bitmap defining the frequencyDomainAllocation [3GPP TS 38.211 [2], sec 7.4.1.5.3] [3GPP TS 38.331 [6] CSI-Resource Mapping] Value: Up to the 12 LSBs, actual size is determined by the Row parameter
SymbL0	uint8_t	The time domain location l0 and firstOFDMSymbolInTimeDomain [3GPP TS 38.211 [2], sec 7.4.1.5.3] Value: 0->13
SymbL1	uint8_t	The time domain location l1 and firstOFDMSymbolInTimeDomain2 [3GPP TS 38.211 [2], sec 7.4.1.5.3] Value: 2->12
CDMType	uint8_t	The cdm-Type field [3GPP TS 38.211 [2], sec 7.4.1.5.3 and table 7.4.1.5.3-1] Value: 0: noCDM, 1: fd-CDM2, 2: cdm4-FD2-TD2, 3: cdm8-FD2-TD4
FreqDensity	uint8_t	The density field, p and comb offset (for dot5). [3GPP TS 38.211 [2], sec 7.4.1.5.3 and table 7.4.1.5.3-1] Value: 0: dot5 (even RB), 1: dot5 (odd RB), 2: one, 3: three
ScrambId	uint16_t	ScramblingID of the CSI-RS [3GPP TS 38.214 [5], sec 5.2.2.3.1] Value: 0->1023
Tx Power Info		
powerControlOffset ProfileNR	uint8_t	Ratio of PDSCH EPRE to NZP CSI-RSEPRE [3GPP TS 38.214 [5], sec 5.2.2.3.1]



Field	Type	Description
		Value :0->23 representing -8 to 15 dB in 1dB steps 255: L1 is configured with ProfileSSS
powerControlOffsetSS ProfileNR	uint8_t	Ratio of NZP CSI-RS EPRE to SSB/PBCH block EPRE [3GPP TS 38.214 [5], sec 5.2.2.3.1] Values: 0: -3dB, 1: 0dB, 2: 3dB, 3: 6dB 255: L1 is configured with ProfileSSS
Beamforming		
Precoding and Beamforming	structure	See Table 3-93. If the PHY supports Rel-16 see also Table 3-94. For PHYs supporting FAPIv4 MU-MIMO groups, see also Table 3-95.
CSI-RS Maintenance Parameters added in FAPIv3		
CSI-RS Parameters added in FAPIv4	structure	See Table 3-86

Table 3-84 CSI-RS PDU

Field	Type	Description
Set of CSI-RS PDU parameters added in FAPIv3		
Basic		
csiRsPduIndex	uint16_t	Unique across all CSI-RS PDUs in the slot
Tx Power info		
csiRsPowerOffset ProfileSSS	int16_t	Ratio of CSI-RS EPRE to SSS EPRE Values: -32767 -> -32767 representing -32.767dB to 32.767dB in 0.001dB steps -32768: ProfileNR is in use

Table 3-85 CSI-RS PDU maintenance parameters from FAPIv3

Field	Type	Description
MU-MIMO support in FAPIv4		
numSpatial Stream Indices	uint8_t	<p>Set to the number of ports in the cdm group, or zero.</p> <p>Must be set to set to the number of ports in the cdm group if PHY indicates that it requires DL spatial stream index assignment, via capability maxNumberDISpatialStreams (TLV)</p>



Field	Type	Description
		0x0151), may be set to zero otherwise.
spatialStreamIndices	uint8_t [n]	<p>n = numSpatialStreamIndices</p> <p>Spatial Stream indices for mapping of CSI ports in the cdm group, based on PHY capability indicating need for spatial stream index signaling.</p> <p>In this release, the order of the indices corresponds to index of the layer dimension in the Precoding Matrix for the Precoding and Beamforming PDU.</p> <p>Value: 0 → (max # spatial streams - 1, per TLV 0x0151)</p>

Table 3–86 CSI-RS PDU parameters for FAPIv4

3.4.2.4 SSB PDU

The format of the SSB/BCH PDU is shown in Table 3–87.

The SSB/BCH PDU has several options for the payload length and contents these are:

- If the MAC generates the full PBCH payload, the payload length is 32 bits and Table 3–89 defines payload
- If the PHY generates the timing PBCH information, the payload length is 24 bits and Table 3–89 defines payload
- If the PHY generates the full PBCH payload, Table 3–90 defines the payload
- The choice of PBCH payload generation will be determined when the PHY is configured.

Field	Type	Description
physCellId	uint16_t	Physical cell ID. [3GPP TS 38.211 [2], sec 7.4.2.1] N_{ID}^{cell} Value: 0->1007
betaPssProfileNR	uint8_t	PSS EPRE to SSS EPRE in a SS/PBCH block [3GPP TS 38.213 [4], sec 4.1] Values: 0 = 0dB 1 = 3dB 255: power offset is conveyed in betaPssProfileSSS
ssbBlockIndex	uint8_t	SS/PBCH block index within a SSB burst set [3GPP TS 38.211 [2], section 7.3.3.1]. Required for PBCH DMRS scrambling. Value: 0->63 (L_{max})



Field	Type	Description
ssbSubcarrierOffset	uint8_t	ssbSubcarrierOffset or k_{SSB} (3GPP TS 38.211 [2], section 7.4.3.1) Value: 0->31
SsbOffsetPointA	uint16_t	Offset of lowest subcarrier of lowest resource block used for SS/PBCH block. Given in PRB [3GPP TS 38.211 [2], section 4.4.4.2] Value: 0->2199
bchPayloadFlag	uint8_t	A value indicating how the BCH payload is generated. This should match the PARAM/CONFIG TLVs. Value: 0: MAC generates the full PBCH payload, see Table 3-89, where bchPayload has 32 bits 1: PHY generates the timing PBCH bits, see Table 3-89, where the bchPayload has 24 bits 2: PHY generates the full PBCH payload, see Table 3-90.
bchPayload	struct	See Table 3-89 and Table 3-90
Beamforming		
Precoding and Beamforming	structure	See Table 3-93. If the PHY supports Rel-16 see also Table 3-94. For PHYs supporting FAPIv4 MU-MIMO groups, see also Table 3-95.
SSB/PBCH Maintenance Parameters added in FAPIv3	structure	See Table 3-88
SSB/PBCH Parameters added in FAPIv4	structure	See Table 3-91

Table 3-87 SSB/PBCH PDU

Field	Type	Description
Set of SSB PDU parameters added in FAPIv3		
Basic Parameters		
ssbPduIndex	uint8_t	Unique PDU id across all SSB PDUs signaled in this slot.
Case	uint8_t	First symbol mapping for candidate SSB location, per [3GPP TS 38.213 [4], sec 4.1] 0: Case A 1: Case B 2: Case C 3: Case D 4: Case E
SubcarrierSpacing	uint8_t	subcarrierSpacing [3GPP TS 38.211 [2], sec 4.2] Value:0->4
IMax	uint8_t	L_{max} , as defined in 3GPP TS 38.214 [5], section 7.4.1.4.1.
Tx Power info		



Field	Type	Description
ssPbchBlockPowerScaling	int16_t	Baseband power scaling applied to SS-PBCH signal [3GPP TS 38.211 [2], sec 7.4.2.3.1) Values: -11000 -> 12000 representing -110.0 dB to 120.0 dB in 0.01dB -32768: L1 determines PSS power as in SCF-222 v2.0
betaPSSProfileSSS	int16_t	Ratio of PSS EPRE to SSS EPRE. Values: -32767 -> -32767 representing -32.767dB to 32.767dB in 0.001dB -32768: L1 determines PSS power from betaPSSProfileNR 3GPP reference: 3GPP TS 38.213 [4], section 4.1

Table 3-88 SSB/PBCH PDU maintenance FAPIv3

Field	Type	Description
bchPayload	uint32_t	BCH payload. The valid bits are indicated in the PARAM/CONFIG TLVs. If PARAM/CONFIG TLVs indicate MAC generates full bchPayload: <ul style="list-style-type: none">- the payload length is 32 bits with the 8 LSB bits being (SFN, half-radio frame bit, SS/PBCH block idx) , in the same order as specified in [3GPP TS 38.212 [3], sec 7.1.1]- a₀ in 3GPP TS 38.212 [3], section 7.1.1 is mapped to LSB 31 and a₃₁ is mapped to LSB 0 of bchPayload field Otherwise: <ul style="list-style-type: none">- timing PBCH bits are generated by the PHY.- for bchPayload the 24 LSB are used, in the same order as specified in [3GPP TS 38.212 [3], sec 7.1.1].- a₀ in 3GPP TS 38.212 [3], section 7.1.1 is mapped to LSB 23 and a₂₃ is mapped to LSB 0 of bchPayload field

Table 3-89 MAC generated MIB PDU

Field	Type	Description
DmrsTypeAPosition	uint8_t	The position of the first DM-RS for downlink and uplink. Value: 0 -> 1
PdcchConfigSib1	uint8_t	The parameter PDCCH-ConfigSIB1 that determines a common <i>ControlResourceSet</i> (CORESET) a common search space and necessary PDCCH parameters. Value: 0 -> 255



Field	Type	Description
CellBarred	uint8_t	The flag to indicate whether the cell is barred Value: 0 -> 1
IntraFreqReselection	uint8_t	The flag to controls cell selection/reselection to intra-frequency cells when the highest ranked cell is barred, or treated as barred by the UE. Value: 0 -> 1

Table 3–90 PHY generated MIB PDU

Field	Type	Description
MU-MIMO support in FAPIv4		
spatialStreamIndexPresent	uint8_t	Indicates whether a spatial stream index is signaled or not. Must be set to true if PHY indicates that it requires DL spatial stream index assignment, via capability maxNumberDISpatialStreams (TLV 0x0151). 0 = false 1 = true
spatialStreamIndex	uint16_t	This entry is present if and only if spatialStreamIndexPresent is true. Spatial stream index for mapping SSB PDU. In this release there is a single spatial stream per SSB in the Precoding Matrix for the Precoding and Beamforming PDU. Value: 0 → (max # spatial streams-1, per TLV 0x0151)

Table 3–91 SSB PDU parameters for FAPIv4

3.4.2.4a PRS PDU

Field	Type	Description
SubcarrierSpacing	uint8_t	subcarrierSpacing [3GPP TS 38.211 [2], sec 4.2] Value: 0->4
CyclicPrefix	uint8_t	Cyclic prefix type [3GPP TS 38.211 [2], sec 4.2] 0: Normal; 1: Extended
nIdPrs	uint16_t	Parameter $n_{ID,seq}^{PRS}$, as provided in [3GPP TS 38.211 [2], sec 7.4.1.7.2] Value: 0 ... 4095
pduIndex	uint16_t	PDU index incremented for each PRS PDU sent in TX control message. This is used to associate control information to data and is reset every slot.



Field	Type	Description
		Value: 0 -> 65535
combSize	uint8_t	Transmission comb size K_{comb}^{PRS} [3GPP TS 38.211 [2], sec 7.4.1.7.3] Value: 2, 4, 6, 12
combOffset	uint8_t	Transmission comb offset k_{offset}^{PRS} [3GPP TS 38.211 [2], sec 7.4.1.7.3] Value: 0 ... (K_{comb}^{PRS} -1)
numSymbols	uint8_t	Number of symbols L_{PRS} [3GPP TS 38.211 [2], sec 7.4.1.7.3] Value: 2, 4, 6, 12
firstSymbol	uint8_t	Symbol offset l_{start}^{PRS} [3GPP TS 38.211 [2], sec 7.4.1.7.3] Value: 0 ... 12
numRBs	uint16_t	PRS bandwidth RBs, as provided by parameter dl-PRS-ResourceBandwidth [3GPP TS 37.355 [28], sec 6.4.3] Value: 24 ... 272 PRBs, in multiples of 4.
startRB	uint16_t	PRB offset with respect to Point-A, as provided by parameter dl-PRS-StartPRB [3GPP TS 37.355 [28], sec 6.4.3] Value: 0 ... 2176
symbolPuncturing Bitmap	uint16_t	A bitmap of PRS symbols that are punctured For each bit: <ul style="list-style-type: none">• 0: symbol punctured• 1: symbol transmitted Bit positions: <ul style="list-style-type: none">• [0]: symbol #0• [1]: symbol #1• ...• ...• [14]: reserved (set to 0, in this release)• [15]: reserved (set to 0, in this release) Note: this bitmap serves for handling SSB overlaps.
spatialStreamIndexPresent	uint8_t	Indicates whether a spatial stream index is signaled or not. Must be set to true if PHY indicates that it requires DL spatial stream index assignment, via capability maxNumberDISpatialStreams (TLV 0x0151). 0 = false 1 = true
spatialStreamIndex	uint16_t	This entry is present if and only if spatialStreamIndexPresent is true.



Field	Type	Description
		Spatial stream index for mapping PRS PDU. In this release there is a single spatial stream per PRS in the Precoding Matrix for the Precoding and Beamforming PDU. Value: 0 → (max # spatial streams-1, per TLV 0x0151)
Beamforming		
Precoding and Beamforming	structure	See Table 3-93. If the PHY supports Rel-16 see also Table 3-94. For PHYs supporting FAPIv4 MU-MIMO groups, see also Table 3-95.
Tx Power info		
prsPowerOffset	int16_t	Ratio of PRS EPRE to SSS EPRE; see parameter β_{PRS} , [3GPP TS 38.211 [2], sec 7.4.1.7.3] Values: -32767 -> -32767 representing -32.767dB to 32.767dB in 0.001dB steps -32768: reserved

Table 3-92 PRS PDU

3.4.2.5 Tx Precoding and Beamforming PDU

The precoding and beamforming PDU is included in the PDCCH, PDSCH, CSI-RS and SSB PDUs. The format is shown in Table 3-93.

Field	Type	Description
TRP scheme	uint8_t	This field shall be set to 0, to identify that this table is used.
numPRGs	uint16_t	Number of PRGs spanning this allocation. Value : 1->275
prgSize	uint16_t	Size in RBs of a precoding resource block group (PRG) - to which same precoding and digital beamforming gets applied. Value: 1->275
digBFInterfaces	uint8_t	Number of logical antenna ports (parallel streams) feeding into the digBF Value: 0->255
For number of PRGs {		
PMidx	uint16_t	Index to precoding matrix (PM) pre-stored at cell configuration. Note: If precoding is not used this parameter should be set to 0. Value: 0->65535.
for each digBFInterface {		



Field		Type	Description
	beamIdx	uint16_t	Index of the digital beam weight vector pre-stored at cell configuration. The vector maps the logical port indexed by digBFIInterface to output baseband ports. Value: 0->65535
}			}

Table 3–93 Tx precoding and beamforming PDU

3.4.2.6 Rel-16 mTRP Tx Precoding and Beamforming PDU

The precoding and beamforming PDU is included in the PDCCH, PDSCH, CSI-RS and SSB PDUs, when PHY hosts both ports of a Rel-16 mTRP configuration, per 3GPP 38.300 [11], section 6.12. The format is shown in Table 3–94.

L1 only expects this table if it supports Rel-16 mTRP.

Field		Type	Description
TRP scheme	uint8_t	0: refer to Table 3–93, instead, for the format of the Tx Precoding and Beamforming PDU 1: sTRP: all baseband ports are used (like in Rel-15) => #TRPs = 1 2: mTRP: single TRP (first) in PHY => #TRPs = 1 3: mTRP: single TRP (second) in PHY => #TRPs = 1 4: mTRP: both TRPs in PHY => #TRPs = 2	
numPRGs	uint16_t	Number of PRGs spanning this allocation. Value: 1->275	
prgSize	uint16_t	Size in RBs of a precoding resource block group (PRG) – to which same precoding and digital beamforming gets applied. Value: 1->275	
digBFIInterfaces	uint8_t [#TRPs]	Number of logical antenna ports (parallel streams) feeding into the digBF Value: 0->255	
For number of PRGs {			
Layers	uint8_t	Bitmap of layers: MSB = layer 1, LSB = layer 8. Bit = 0 => layer precoded via PMidx[0] Bit = 1 => layer (if any) precoded via PMidx[1] (if #TRPs = 2, else dropped)	
PMidx	uint16_t[#TRPs]	Index to precoding matrix (PM) pre-stored at cell configuration. Note: If precoding is not used, this parameter should be set to 0 and is interpreted as PMidx[.] = Identity. Value: 0->65535.	
For each TRP			



Field		Type	Description
for each digBFIInterface {			
	beamIdx	uint16_t	<p>Index of the digital beam weight vector pre-stored at cell configuration. The vector maps the logical port indexed by dibBFIInterface to output baseband ports for the TRP.</p> <p>The first N beam weights used is equal to the number of N of baseband ports assigned to the TRP.</p> <p>Value: 0->65535</p>
	}		
	}		
	}		

Table 3–94 Tx precoding and beamforming PDU

3.4.2.7 Tx Precoding based on Channel Reciprocity PDU

The Tx Precoding PDU based on Channel Reciprocity sampling; in this release, this PDU may be included, when precoding is performed as described in section 2.2.6.2. The format is shown in Table 3–95.

Field		Type	Description
TRP scheme		uint8_t	5: Single TRP, based on channel reciprocity
numMuMimo Groups		uint8_t	<p>Number of DL MuMIMO groups covering the allocation of this PDU.</p> <p>Any PRBs in the allocation for the channel must belong to exactly one MuMIMO group. For SU-MIMO numMuMIMOGroups to be set to 1.</p>
For each of the numMuMimoGroups {			
	muMimoGroupIdx	uint16_t	<p>muMimoGroups are listed in increasing muMimoGroup index order. A PRG is defined in exactly one muMimoGroup.</p> <p>Value: 0→3,821</p> <p>Note: muMimoGroupIdx are defined in increasing order of their start symbol and on a given start symbol they are defined in increasing order of PRB</p>
	}		

Table 3–95 Tx generalized precoding PDU

3.4.2b DL_TTI.response

The format of the `DL_TTI.response` message is shown in Table 3–96. It is sent by L1 if L2 requests TB CRC in `DL_TTI.request` PDSCH PDU setting `pduBitmap[1]` = 1 on a new PDSCH transmission, when L1 is configured for CBG segmentation in L2. The TB CRC returned by L1 can be used for subsequent CBG retransmission

A `DL_TTI.response` message indicates the SFN/Slot of the downlink slot it contains.



This message can be sent by the L1 when the PHY is in the RUNNING state.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
nPDUs	uint16_t	Number of PDSCH PDUs for which TBS CRC are supplied in this message. Value 0 -> 65535
For number of PDUs		
pduIndex	uint16_t	Used to associate the TB CRC Payload PDU with the DL_TTI PDSCH PDU Value: 0 → 65535
CW	uint8_t	0 or 1, in this release, to match the number of codewords in the PDSCH PDU with index pduIndex
tbCrc	uint32_t	The TB-level CRC calculation results for transport block corresponding to the indicated CW that transmitted in this slot. Only up to 24 LSB of each array element are valid, L2 shall extract the correct TB CRC from the array based on scheduling information for this slot for this UE. See 3GPP TS 38.212 [3], section 7.2.1

Table 3–96 DL_TTI.response message body

3.4.3 UL_TTI.request

The format of the UL_TTI.request message is shown in Table 3–97. An UL_TTI.request message indicates the SFN and slot it contains information for. This control information is for an uplink slot.

This message can be sent by the L2/L3 when the PHY is in the RUNNING state. If it is sent when the PHY is in the IDLE or CONFIGURED state an ERROR.indication message will be sent by the PHY.

The following combinations of PDUs are required:

- In order to support PRACH in the subframe, the RACH present field must be true
- The PUSCH PDU is present when a UE has been instructed to send uplink data and/or control data on the PUSCH
- The PUCCH PDU is present when a UE has been instructed to send control data on the PUCCH
- The PRACH PDU is present when a UE may transmit RACH.
- The SRS PDU is present when a UE has been instructed to send SRS.



The PDUs included in this structure have no ordering requirements.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
nPDUs	uint16_t	Number of PDUs that are included in this message. All PDUs in the message are numbered in order. Value: 0 -> 65535 Backward compatibility note: the field was 8 bits in SCF222 v2.0
nUITypes	uint8_t	Maximum number of UL PDU types supported by <code>UL_TTI.request</code> . nUITypes = 6 in this release. Backward compatibility note: SCF222 v2.0 did not have this field
nPDUsOfType	uint16_t [nUITypes]	Number of PDUs of each type that are included in this message. Each array entry corresponds to a PDU type as follows: [0]: number of PRACH PDUs [1]: number of PUSCH PDUs [2]: number of Format 0/1 PUCCH PDUs [3]: number of Format 2/3/4 PUCCH PDUs [4]: number of SRS PDUs [5]: number of MsgA-PUSCH PDUs Value 0 -> 65535, for each entry Backward compatibility note: SCF222 v2.0 did not have this field
nGroup	Uint16_t	Number of PDU Groups included in this message. Value: 0 -> 3,822
For Number of PDUs {		
PDUType	uint16_t	0: PRACH PDU, see Section 3.4.3.1 1: PUSCH PDU, see Section 3.4.3.2 2: PUCCH PDU, see Section 3.4.3.3 3: SRS PDU, see Section 0 4: MsgA-PUSCH PDU, see Section 3.4.3.2A
PDUSize	uint16_t	Size of the PDU control information (in bytes). This length value includes the 4 bytes required for the PDU type and PDU size parameters. Value: 0 -> 65535



Field		Type	Description
	UL PDU Configuration	structure	See Sections 3.4.3.1 to 0
}			In this message, if nGroup > 0 then for each group { -- For FAPIv4, groups represent MU MIMO groups, as described in section 2.2.6.2; muMimoGroups are listed in increasing order of their start symbol. muMimogroups starting at the same symbol, are listed in increasing order of their first PRB index.
	nPDUs	uint8_t	Number of Channel PDU in this group. PDUs in a group must be of the same type. For SU-MIMO, one group includes one UE only. For MU-MIMO, one group includes up to 32 UEs. Value: 1->32
In the group, for each PDU {			
	PduIdx	uint16_t	This value is an index for number of PDU identified by nPDU in this message. Value: 0->255
	prgSize	uint16_t	This field signals the number of UL PRBs with common precoding, for PHY to use in constructing UL combiners. For FAPIv2 grouping: <ul style="list-style-type: none"> • prgSize is set to 0 • nPdus is limited to 1...12 • prbBitmap, symbolBitmap fields are omitted. the PRG loop below is omitted. Value: 1 → 275 (FAPIv4), or 0 (to mean FAPIv2)
	prbBitmap	uint8_t [36]	bitmap of RBs assumed common to all RBs in the group, for joint spatial precoding. 273 rounded up to multiple of 32. This bitmap is in units of VRBs. LSB of byte 0 of the bitmap represents the PRB 0.
	symbolBitmap	uint16_t	bitmap of symbols for the joint spatial precoding group bit location indicates the symbol index, up to 14 symbols. First symbol corresponds to location #0
}			
}			

Table 3-97 UL_TTI.request message body

3.4.3.1 PRACH PDU

The format of the PRACH PDU is shown in Table 3-98.

Field		Type	Description
physCellID	uint16_t	10 bits corresponding to N_{ID}^{cell} . Value: 0 ->1007	



Field	Type	Description
numPrachOcas	uint8_t	<p>Number of time-domain PRACH occasions within a PRACH slot, N_t^{RAslot} [3GPP 38.211 [2], sec 6.3.3.2]</p> <p>Value: 1->7</p>
prachFormat	uint8_t	<p>RACH format information for the PRACH occasions signaled in this PDU.</p> <p>This corresponds to one of the supported formats i.e., 0, 1, 2, 3, A1, A2, A3, B1, B4, C0, C2, A1/B1, A2/B2 or A3/B3.</p> <p>[3GPP 38.211 [2], sec 6.3.3.2]</p> <p>Values:</p> <p>0 = 0 1 = 1 2 = 2 3 = 3 4 = A1 5 = A2 6 = A3 7 = B1 8 = B4 9 = C0 10 = C2 11 = A1/B1 12 = A2/B2 13 = A3/B3</p> <p>If any of the values 11-13 is signaled, the last of the numPrachOcas time domain occasions is of the B type; any occasions preceding it are of the A type.</p>
indexFdRa	uint8_t	<p>Frequency domain occasion index $n \in \{0,1,\dots, M - 1\}$, where M equals the higher-layer parameter msg1-FDM which can take values {1,2,4,8} [3GPP TS 38.211 [2], sec 6.3.3.2]</p> <p>Values: 0->7</p>
prachStartSymbol	uint8_t	<p>Starting symbol for the first PRACH TD occasion in the current PRACH FD occasion. Corresponds to the parameter.</p> <p>[3GPP TS 38.211 [2], sec 6.3.3.2 and Tables 6.3.3.2-2 to 6.3.3.2-4]</p> <p>Values: 0->13</p>
numCs	uint16_t	<p>Zero-correlation zone configuration number (RRC parameter zeroCorrelationZoneConfig). Corresponds to the L1 parameter N_{cs}. [3GPP TS 38.211 [2], sec 6.3.3.1 and Table 6.3.3.1-5, 6.3.3.1-6 and 6.3.3.1-7]</p> <p>Value: 0->419</p>
IsMsgA-Prach	uint8_t	<p>Signals whether this msgA PRACH is at least partially used for 2-Step RACH.</p> <p>Value:</p> <ul style="list-style-type: none">- 0 = False (no <i>MsgA-PRACH to MsgA-PUSCH map signalling</i>)



Field	Type	Description
		<ul style="list-style-type: none"> - 1 = True (<i>MsgA-PRACH to MsgA-PUSCH map signalling</i> is included in this PDU) - 2 = True (<i>MsgA-PRACH to MsgA-PUSCH map signalling</i> is not included in this PDU, since PHY determines it for P5 configuration).
hasMsgA-PuschBeamforming	uint8_t	<p>When IsMsgA-Prach is true, signals whether Beamforming structure is present for the MsgA-PRACH (in which case, the structure signals beams for all PRBs for valid MsgA-PUSCH PDUs).</p> <p>0: not included 1: included</p>
Beamforming		
Beamforming	structure	See Table 3-117; for PHYs supporting MU-MIMO groups, see also Table 3-118
PRACH Maintenance Parameters added in FAPIv3	structure	See Table 3-99
Uplink spatial stream assignment FAPIv4	structure	See Table 3-100
MsgA-PRACH to MsgA-PUSCH map signalling in FAPIv4	structure	<p>Only included if IsMsgA-Prach = 1</p> <p>See Table 3-101</p>
MsgA-PUSCH Beamforming	Structure	<p>Only included if IsMsgA-Prach = <i>True</i> and hasMsgA-PuschBeamforming = <i>included</i></p> <p>See Table 3-117; for PHYs supporting MU-MIMO groups, see also Table 3-118</p>

Table 3-98 PRACH PDU

Field	Type	Description
Set of PRACH PDU parameters added in FAPIv3		
handle	uint32_t	An opaque handle returned in the RACH.indication
prachConfigScope	uint8	<p>prachResConfigIndex being referred to:</p> <p>0: the index shall be looked up in the Common Context (Shared by all PHYs)</p> <p>1: the index index shall be looked up in this PHY's Context</p>
prachResConfigIndex	uint16_t	The PRACH configuration for which this PRACH PDU is signaled (c.f. Table 3-42)
numFdRa	uint8_t	<p>Number of frequency domain occasions, starting with indexFdRa [3GPP TS 38.211 [2], sec 6.3.3.2]</p> <p>Values: 1->8</p>
startPreambleIndex	uint8_t	Start of preamble logical index to monitor in the PRACH occasions signaled in this slot.



Field	Type	Description
		<p>Value: 0 – 63 255: all preambles from the PRACH configuration linked by prachResConfigIndex</p> <p>Preamble logical indices for a RACH occasion are defined as in 3GPP TS 38.211 [2], section 6.3.3.1, starting from the <i>prachRootSequenceIndex</i> of the PRACH configuration for the RO.</p>
numPreambleIndices	uint8_t	<p>Number of preamble logical indices, starting with startPreambleIndex, to monitor in the PRACH occasions signaled in this slot.</p> <p>Value: 1 – 64</p>

Table 3–99 PRACH maintenance FAPIv3

Field	Type	Description
Set of PRACH PDU parameters added in FAPIv4		
numSpatialStreams	uint8_t	<p>Non-zero if PHY indicates that UL spatial stream assigned is required, in capability maxNumberULSpatialStreams (TLV 0x0153)</p> <p>The number of spatial streams must be at least as large as the number of layers.</p> <p>When semistatic logical ports are used, the number of spatial streams is the same as the number of logical ports.</p>
spatialStreamIndices	uint16_t [n]	<p>n = [numSpatialStreams]</p> <p>The indices of the spatial streams used by this channel.</p> <p>Value: 0 → (max # spatial streams - 1, per TLV 0x0153)</p>

Table 3–100 Uplink spatial stream assignment FAPIv4

Field	Type	Description
Set of 2-Step RACH PRACH PDU parameters added in FAPIv4		
numberOfpreambleGroups	uint8_t	<p>Indicates the number of preamble groups (#cfg) signaled for this PRACH PDU (c.f. 3GPP TS 38.331 [6], section 6.3.2; 3GPP TS 38.321 [21], sections 5.1.1 and 5.1.2a).</p> <p>These can correspond to common groups (A or B) or dedicated groups, including singular groups (consisting of single preambles)</p>
dmrsPorts	uint16_t [#cfg]	Bitmap of all the DMRS ports amongst which ports are picked for each PUSCH occasion of the MsgA-PUSCH PDU(s) mapped from this PRACH PDU, [3GPP TS 38.212 [3], 7.3.1.1.2]



Field	Type	Description
		<p>based on msgA-PRACH → msgA-PUSCH mapping</p> <p>Bit mapping is as follows:</p> <ul style="list-style-type: none">• bit 0 : antenna port 1000• bit 7: antenna port 1007 <p>For each bit</p> <ul style="list-style-type: none">• 0: DMRS port not used• 1: DMRS port used <p>numDmrsPorts = sum of all bits in dmrsPorts</p> <p>Note: the actual DMRS port to use is based on the msgA-PRACH → msgA-PUSCH & associated DMRS mapping</p>
available-DmrsSequenceIds	uint8_t [#cfg]	<p>Bitmap indicating the DMRS sequence IDs available for the MsgA-PUSCH PDU(s) corresponding to this PRACH PDU. For each bit position:</p> <ul style="list-style-type: none">• 0: ID not supported• 1: ID supported <p>Bit significance:</p> <ul style="list-style-type: none">- [0]: n_{SCID} = 0 available.- [1]: n_{SCID} = 1 available- [2-7]: n/a (bit value shall be set to 0) <p>numDmrsSequences = sum of all bits in availableDmrsSequenceIds</p>
numPuschOcasFd	uint8_t [#cfg]	<p>The number of frequency domain PUSCH occasions within the UL slot for the corresponding Msg-A PUSCH PDU(s).</p> <p>Value: 1, 2, 4, 8</p>
numPuschOcasTd	uint8_t [#cfg]	<p>The number of time domain PUSCH occasions for the corresponding MsgA-PUSCH PDU(s).</p> <p>Value: 1, 2, 3, 6</p>
numPuschOcasSlots	uint8_t [#cfg]	<p>The number of slots configured for the corresponding MsgA-PUSCH PDU(s)</p> <p>Value: 1...4</p>
msgA-PUSCH-TimeDomainOffset	uint8_t [#cfg]	<p>The time domain offset, as specified in [3GPP TS 38.213 [22], section 8.1A] in slots between this PRACH PDU slot where a MsgA-PRACH preamble may be triggered and the corresponding MsgA-PUSCH PDU(s) signalling the MsgA-PUSCH occasion.</p> <p>Value: 1...32</p>
Npreambles	uint16_t [#cfg]	<p>N_{preambles}, per section 8.1A of 38.213 [22], section 8.1A as computed by MAC for the particular mapping of MsgA-PRACH to Msg-A PUSCH mapping relevant to the preamble group and this PRACH PDU.</p> <p>Values: 1 ... 3,072</p>



Field	Type	Description
MsgA-PRACH-to-(PRU & DMRS) mapping	See Table 3-44	<p>For this mapping, the following are as signalled for this PRACH PDU:</p> <ul style="list-style-type: none">• $\text{prachFd1} \dots \text{prachFdMax}$ = indexFdRa ... numFdRa-1• $\text{prachTd1} \dots \text{prachTdMax}$ = 0 ... numPrachOcas-1 <p>The preamble group range $\text{grpStart} \dots \text{grpEnd}$ depends on numberOfpreambleGroups:</p> <ul style="list-style-type: none">• $\text{grpStart} = 1$• $\text{grpEnd} = \text{numberOfpreambleGroups}$ <p>Furthermore, the following parameters are used to interpret the contents of the mapping, for each group:</p> <ul style="list-style-type: none">• $N_{preambles}$: Npreambles• numPuschOcasFd: numPuschOcasFd• numDmrsPorts: sum of all bits in dmrsPorts• numDmrsSequences: sum of all bits in available-DmrsSequenceIds• numPuschOcasTd: numPuschOcasTd• numPuschOcasSlots: numPuschOcasSlots• $\text{startPreambleIndex}$: startPreambleIndex in this PRACH PDU• numberOfPreambles: numberOfPreambles in this PRACH PDU

Table 3-101 MsgA-PRACH to MsgA-PUSCH map signalling in FAPIv4

3.4.3.2 PUSCH PDU

The format of the PUSCH PDU is given in Table 3-102.

The PUSCH PDU includes both mandatory and optional parameters, where the presence of the optional elements is defined in the parameter pduBitmap. The presence of these optional elements is included based on the following:

- `puschData` is set if an ULSCH transmission is expected from the UE
- `puschData` and `puschUCI` are both set if ULSCH and UCI transmission on PUSCH is expected from the UE
- `puschUCI` is set and `puschData` is not set if UCI information is expected on PUSCH, but ULSCH is not expected
- `puschPtrs` is included if and only if PTRS are included in the uplink transmission
- `dftsOfdm` is included for DFT S-OFDM transmission

Only parameters signalled by `pduBitmap` are optional, all other parameters are mandatory. (It should be noted that some parameters are only applicable for certain modes etc. These are still included but their value is ignored by the PHY.)

Field	Type	Description
<code>pduBitmap</code>	<code>uint16_t</code>	Bitmap indicating presence of optional PDUs Bit 0: <code>puschData</code> (Indicates data is expected on the PUSCH)



Field	Type	Description
		<p>Bit 1:puschUci (Indicates UCI is expected on the PUSCH)</p> <p>Bit 2: puschPtrs (Indicates PTRS included. L1 behavior is undefined if this bit is 1 when 3GPP procedures require no PT-RS transmission)</p> <p>Bit 3: dftsOfdm (Indicates DFT S-OFDM transmission)</p> <p>All other bits reserved</p>
RNTI	uint16_t	<p>The RNTI used for identifying the UE when receiving the PDU</p> <p>Value: 1 -> 65535</p>
Handle	uint32_t	An opaque handling returned in the RX_DATA.indication and/or UCI.indication message
BWP [3GPP TS 38.213 [4], sec 12]		
BWPSize	uint16_t	<p>If L1 uses puschTransType, this field signals the bandwidth part size [3GPP TS 38.213 [4], sec12]. Number of contiguous PRBs allocated to the BWP.</p> <p>If L1 does not use puschTransType, this field represents the number of VRBs to which PRBs are mapped for the allocation, according to 3GPP TS 38.211 [2], section 6.3.1.7.</p> <p>Note: for BWPSize, the two interpretations align, except when the allocation is scheduled for msg3 (by RAR UL grant), in which case the first interpretation uses to active BWP,i size, whereas the second one uses the initial BWP,0 size referenced by 3GPP TS 38.213 [4], section 8.3.</p> <p>Value: 1->275</p>
BWPStart	uint16_t	<p>If L1 uses puschTransType, this field signals the bandwidth part start RB index from reference CRB [3GPP TS 38.213 [4], sec 12]</p> <p>If L1 does not use puschTransType, this field represents the CRB corresponding to VRB0, according to 3GPP TS 38.211 [2], section 6.3.1.7.</p> <p>Note 1: For BWPStart, the two interpretations align, except when the allocation is scheduled for msg3 <i>“by DCI format 0_0 with CRC scrambled by TC-RNTI in active uplink bandwidth part i starting at $N_{BWP,i}^{start}$, including all resource blocks of the initial uplink bandwidth part starting at $N_{BWP,0}^{start}$, and having the same subcarrier spacing and cyclic prefix as the initial uplink bandwidth part”</i>, in which case the first interpretation uses to active BWP,i start whereas the second one uses the initial BWP,0 start referenced by 3GPP TS 38.211 [2], section 6.3.1.7.</p>



Field	Type	Description
		<p>Note 2: Only msg3 allocations subject to the conditions listed in Note 1 above are referenced to the initial BWP,0 start. Other types of msg3 allocations are referenced to the start of active uplink BWP,I, as specified in 3GPP TS 38.211 [2], section 6.3.1.7 and 3GPP TS 38.213 [4], section 8.3.</p> <p>Value: 0->274</p>
SubcarrierSpacing	uint8_t	subcarrierSpacing [3GPP TS 38.211 [2], sec 4.2] Value:0->4
CyclicPrefix	uint8_t	Cyclic prefix type [3GPP TS 38.211 [2], sec 4.2] 0: Normal; 1: Extended
PUSCH information always included		
targetCodeRate	uint16_t	Target coding rate [3GPP TS 38.214 [5], sec 6.1.4.1]. This is the number of information bits per 1024 coded bits expressed in 0.1 bit units
qamModOrder	uint8_t	QAM modulation [3GPP TS 38.214 [5], sec 6.1.4.1] Value: 2,4,6,8 if transform precoding is disabled Value: 1,2,4,6,8 if transform precoding is enabled
mcsIndex	uint8_t	MCS index [3GPP TS 38.214 [5], sec 6.1.4.1], should match value sent in DCI Value : 0->31
mcsTable	uint8_t	MCS-Table-PUSCH [3GPP TS 38.214 [5], sec 6.1.4.1] Value: 0: notqam256 [3GPP TS 38.214 [5], table 5.1.3.1-1] 1: qam256 [3GPP TS 38.214 [5], table 5.1.3.1-2] 2: qam64LowSE [3GPP TS 38.214 [5], table 5.1.3.1-3] 3: notqam256-withTransformPrecoding [3GPP TS 38.214 [5], table 6.1.4.1-1] 4: qam64LowSE-withTransformPrecoding [3GPP TS 38.214 [5], table 6.1.4.1-2]
TransformPrecoding	uint8_t	Indicates if transform precoding is enabled or disabled [3GPP TS 38.214 [5], sec 6.1.4.1] [3GPP TS 38.211 [2], 6.3.1.4] Value: 0: enabled 1: disabled
nIdPusch	uint16_t	parameter n_ID [3GPP TS 38.211 [2], sec 6.3.1.1] Value: 0->1023
nrOfLayers	uint8_t	Number of layers [3GPP TS 38.211 [2], sec 6.3.1.3] Value: 1->4



Field	Type	Description
DMRS [3GPP TS 38.211 [2], sec 6.4.1.1]		
ulDmrsSymbPos	uint16_t	<p>DMRS symbol positions [3GPP TS 38.211 [2], sec 6.4.1.1.3 and Tables 6.4.1.1.3-3 and 6.4.1.1.3-4]</p> <p>Bitmap occupying the 14 LSBs with:</p> <p>bit 0: first symbol and for each bit 0: no DMRS 1: DMRS</p>
dmrsConfigType	uint8_t	<p>UL DMRS config type [3GPP TS 38.211 [2], sec 6.4.1.1.3]</p> <p>0: type 1 1: type 2</p>
puschDmrsScramblingId	uint16_t	<p>PUSCH DMRS Scrambling-ID [3GPP TS 38.211 [2] and v16 [12], sec 6.4.1.1.1.1],</p> <p>as provided by parameter $N_{ID}^{n_{SCID}}$ (Rel-15) or $N_{ID}^{\bar{n}_{SCID}^{\lambda}}$ (Rel-16).</p> <p>It is only valid when the transform precoding for PUSCH is disabled or when lowPaprDmrs is enabled.</p> <p>For example, in references [2] and [12], this field applies to below case:</p> <ol style="list-style-type: none">1. CP-OFDM PUSCH transmission with Rel-15 DM-RS configuration2. CP-OFDM PUSCH transmission having DM-RS port(s) within CDM group 0 or 2 with Rel-16 DM-RS configuration3. Non-MSG3 and non-DCI 0_0 in CSS scheduled DFT-S-OFDM PUSCH transmission with pi/2-BPSK modulation and Rel-16 DM-RS configuration4. Otherwise, L2 shall set it to Physical Cell ID <p>Value: 0->65535</p>
puschDmrsScramblingIdComplement	uint16_t	<p>PUSCH DMRS Scrambling-ID [3GPP TS 38.211 [12], sec 6.4.1.1.1.1], as provided by parameter $N_{ID}^{\bar{n}_{SCID}^{\lambda}}$ (Rel-16) when $\bar{n}_{SCID}^{\lambda} \neq n_{SCID}$. (i.e. $N_{ID}^{1-n_{SCID}}$)</p> <p>PHY shall disregard this parameter if either of this conditions is satisfied:</p> <ol style="list-style-type: none">1. CP-OFDM PUSCH data and lowPaprDmrs = 02. DFTS-OFDM PUSCH data <p>For example, in reference [12], this field applies to below case:</p> <ol style="list-style-type: none">1. CP-OFDM PUSCH transmission having DM-RS port(s) within CDM group 1 with Rel-16 DM-RS configuration <p>Value: 0->65535</p>



Field	Type	Description
lowPaprDmrs	uint8_t	<p>Without Transform Precoding:</p> <p>1: \bar{n}_{SCID}^{λ} depends on DM-RS CDM group</p> <p>0: $\bar{n}_{SCID}^{\lambda} = n_{SCID}$ (i.e. independent of DM-RS CDM group)</p> <p>With Transform Precoding:</p> <p>When PUSCH Transform precoding is enabled, this parameter indicates whether the DMRS scrambling is computed based on c_{init}, or based on clause 5.2.2, as referenced in 3GPP TS 38.211 [12], section 6.4.1.1.2.</p> <p>Values:</p> <p>1: based on c_{init} 0: based on clause 5.2.2 (Z-C DM-RS sequence generation)</p>
puschDmrsIdentity	uint16_t	PUSCH DMRS ID [3GPP TS 38.211 [2] and [12], sec 6.4.1.1.2], as provided by parameter n_{ID}^{RS} <p>This field only valid when Transform Precoding is enabled.</p> <p>Value: 0->1007 (Rel-15), 0-> 65535 (Rel-16)</p>
nSCID	uint8_t	DMRS sequence initialization [3GPP TS 38.211 [2] and [12], sec 6.4.1.1.1], as provided by parameter n_{SCID} . <p>When lowPaprDmrs = 1, UE uses $\bar{n}_{SCID}^{\lambda} = 1 - n_{SCID}$, as described in the section above.</p> <p>It is only valid when the transform precoding for PUSCH is disabled or when lowPaprDmrs is enabled.</p> <p>For example, in references [2] and [12], this value is associated with DM-RS scrambling ID given by puschDmrsScramblingId</p> <p>Value : 0->1</p>
numDmrsCdmGrpsNoData	uint8_t	Number of DM-RS CDM groups without data [3GPP TS 38.212 [3], sec 7.3.1.1] <p>Value: 1->3</p>
dmrsPorts	uint16_t	DMRS ports. [3GPP TS 38.212 [3], 7.3.1.1.2] provides description between DCI 0-1 content and DMRS ports. <p>Bitmap occupying the 12 LSBs with: bit 0: antenna port 1000 bit 11: antenna port 1011 and for each bit 0: DMRS port not used</p>



Field	Type	Description
		1: DMRS port used
Pusch Allocation in frequency domain [3GPP TS 38.214 [5], sec 6.1.2.2]		
resourceAlloc	uint8_t	Resource Allocation Type [3GPP TS 38.214 [5], sec 6.1.2.2] 0: Type 0 1: Type 1
rbBitmap	uint8_t [36]	For resource allocation type 0. [3GPP TS 38.214 [5], sec 6.1.2.2.1] [3GPP TS 38.212 [3], 7.3.1.1.2] bitmap of RBs, 273 rounded up to multiple of 32. This bitmap is in units of VRBs. LSB of byte 0 of the bitmap represents VRB0, per section 6.3.1.7 of 3GPP TS 38.211 [2]
rbStart	uint16_t	For resource allocation type 1. [3GPP TS 38.214 [5], sec 6.1.2.2.2] The starting resource block corresponds to VRB0 for this PUSCH, per section 6.3.1.7 of 3GPP TS 38.211 [2]. Value: 0->274
rbSize	uint16_t	For resource allocation type 1. [3GPP TS 38.214 [5], sec 6.1.2.2.2] The number of resource block within for this PUSCH. Value: 1->275
VRBtoPRBMapping	uint8_t	VRB-to-PRB-mapping [3GPP TS 38.211 [2], sec 6.3.1.7] Value: 0: non-interleaved
IntraSlotFrequencyHopping	uint8_t	For resource allocation type 1. [3GPP TS 38.212 [3], sec 7.3.1.1] [3GPP TS 38.214 [5], sec 6.3] Indicates if intra-slot frequency hopping is enabled Value: 0: disabled 1: enabled
txDirectCurrentLocation	uint16_t	The uplink Tx Direct Current location for the carrier. Only values in the value range of this field between 0 and 3299, which indicate the subcarrier index within the carrier corresponding to the numerology of the corresponding uplink BWP and value 3300, which indicates "Outside the carrier" and value 3301, which indicates "Undetermined position within the carrier" are used. [3GPP TS 38.331 [6], UplinkTxDirectCurrentBWP IE] Value: 0->4095
uplinkFrequencyShift7p5khz	uint8_t	Indicates whether there is 7.5 kHz shift or not. [3GPP TS 38.331 [6], UplinkTxDirectCurrentBWP IE]



Field	Type	Description
		Value: 0: false 1: true
Resource Allocation in time domain [3GPP TS 38.214 [5], sec 5.1.2.1]		
StartSymbolIndex	uint8_t	Start symbol index of PUSCH mapping from the start of the slot, S. [3GPP TS 38.214 [5], Table 6.1.2.1-1] Value: 0->13
NrOfSymbols	uint8_t	PUSCH duration in symbols, L [3GPP TS 38.214 [5], Table 6.1.2.1-1] Value: 1->14
Optional Data only included if indicated in pduBitmap		
puschData	structure	See Table 3-105
puschUci	structure	See Table 3-106
puschPtrs	structure	See Table 3-107
dftsOfdm	structure	See Table 3-108
Beamforming		
Beamforming	structure	See Table 3-117; for PHYs supporting MU-MIMO groups, see also Table 3-118
PUSCH Maintenance Parameters added in FAPIv3	structure	See Table 3-103
PUSCH Parameters added in FAPIv4	structure	See Table 3-109
Optional puschUci added in FAPIv3	structure	See Table 3-104 Included if and only if pduBitmap[1] = 1

Table 3-102 PUSCH PDU

Field	Type	Description
BWP [3GPP TS 38.213 [4], sec 12]		
puschTransType	uint8_t	<p>L1 interprets this field as below, if so configured by puschTransTypeValidity=1, otherwise it ignores it.</p> <p>Indication used to indicate the transmission type of PUSCH PDU.</p> <p>Value 0: UL BWP start and size of the PUSCH PDU shall be set to initial UL BWP including start and size used for MSG3, instead of active downlink BWP start and size</p> <p>Value 1: UL BWP start and size of the PUSCH PDU shall be set to the active UL BWP start as starting point for RB numbering and initial UL BWP size as RB size for MSG3</p> <p>Value 2: UL BWP start and size of the PUSCH PDU shall be set to the active UL BWP start and size, for all other UL transmissions</p>



Field	Type	Description
deltaBwp0StartFromActiveBwp	uint16_t	<p>L1 interprets this field as below, if so configured by puschTransTypeValidity=1, otherwise it ignores it.</p> <p>L1 interprets this field as below, if so configured by puschTransTypeValidity=1, otherwise it ignores it.</p> <p>The value of $N_{BWP,0}^{start} - N_{BWP,i}^{start}$ as described in 3GPP TS 38.211 [2], section 6.3.1.7, representing an offset from active UL BWP start. It is valid if puschTransType = 0 otherwise PHY shall ignore it.</p> <p>Value: 0 to 274</p>
initialUIBwpSize	uint16_t	<p>L1 interprets this field as below, if so configured by puschTransTypeValidity=1, otherwise it ignores it.</p> <p>The initial bandwidth part size [3GPP TS 38.213 [4], sec 12].</p> <p>Only valid when puschTransType field set to 0 or 1, otherwise PHY shall ignore it.</p> <p>Value: 0 to 274</p>
DMRS [3GPP TS 38.211 [2], sec 6.4.1.1]		
groupOrSequenceHopping	uint8_t	<p>PUSCH DMRS hopping mode [3GPP TS 38.211 [2], sec 6.4.1.1.2].</p> <p>It is only valid when the transform precoding for PUSCH is enabled.</p> <p>Value: 0, 1 or 2</p> <ul style="list-style-type: none">• 0: neither, neither group or sequence hopping is enabled• 1: enable, enable group hopping and disable sequence hopping• 2: disable, disable group hopping and enable sequence hopping• other values are reserved.
Frequency Domain Allocation [3GPP TS 38.214 [5], sec 6.1.2.2] and Hopping [3GPP TS 38.214 [5], sec 6.3]		
puschSecondHopPRB	uint16_t	<p>Index of the first PRB after intra-slot frequency hopping, as indicated by the value of RB_{start} for i=1, per 3GPP TS 38.214 [5], section 6.3</p> <p>Valid: when IntraSlotFrequencyHopping Is true</p> <p>Value: 0->274</p>
ldpcBaseGraph	uint8_t	<p>LDPC base graph to use for UL reception [3GPP TS 38.212 [3], sec 6.2.2].</p> <p>Values:</p> <p>1: LDPC base graph 1</p>



Field	Type	Description
		2: LDPC base graph 2 (other values are reserved)
tbSizeLbrmBytes	uint32_t	Parameter TBS _{LBRM} from 3GPP TS 38.212 [3], section 5.4.2.1, for computing the size of the circular buffer. <ul style="list-style-type: none"> • L2 may set this parameter to 0 to indicate that I_{LBRM} = 0 (i.e. no LBRM)

Table 3-103 PUSCH maintenance FAPIv3

Field	Type	Description
In this specification version, L2 includes this TLV:		
<ul style="list-style-type: none"> - to supplement the optional puschUci structure (see Table 3-106), in PUSCH PDU. - for PUCCH formats 3 and 4, in PUCCH PDU. 		
numPart2s	uint16_t	Max number of UCI part2 that could be included in the CSI report. Value: 0 -> 100
For each part2		
priority	uint16	Priority of the part 2 report; L2 shall signal part2 parameters in increasing order of priorities. For UCI over PUSCH, this corresponds to the CSI priority in 3GPP TS 38.214 [5], section 5.2.3 For UCI over PUCCH, this corresponds to the CSI priority in 3GPP TS 38.214 [5], sections 5.2.4 and 5.2.5
numPart1Params	uint8	Number of Part 1 parameters that influence the size of this part 2. Values 1:4 in this release
paramOffsets	uint16 [numPart1 Params]	Ordered list of parameter offsets (offset from 0 = first bit of part1)
paramSizes	uint8 [numPart1 Params]	Bitsizes of part 1 param in the same order as paramOffsets
part2SizeMapIndex	uint16	Index of one of the maps configured Table 3-51, for determining the size of a part2, from the part 1 parameter values. Note, for the indexed map: <ul style="list-style-type: none"> - The number of parameters must match - The size of each parameter in this table must not exceed the size of the same-index parameter used for the indexed map.
part2SizeMapScope	uint8	part2SizeMapIndex being referred to: 0: the map index shall be looked up in the Common Context (Shared by all PHYs) 1: the map index shall be looked up in this PHY's Context

Table 3-104 Uci information for determining UCI Part1 to Part2 correspondence, added in FAPIv3



Field	Type	Description
rvIndex	uint8_t	Redundancy version index [3GPP TS 38.214 [5], sec 6.1.4], it should match value sent in DCI Value : 0->3
harqProcessID	uint8_t	HARQ process number [3GPP TS 38.212 [3], sec 7.3.1.1], it should match value sent in DCI Value: 0 ->15
newData	uint8_t	Signals whether the PUSCH PDU corresponds to an initial transmission or a retransmission of a MAC PDU for this HARQ process ID for this TB [3GPP TS 38.212 [3], sec 7.3.1.1] [3GPP TS 38.214 [5], sec 5.1.7.2] Value: 0: retransmission 1: new data, i.e. initial transmission Note: Unlike NDI, newData does not toggle to indicate new transmission, but is set to 1.
TBSize	uint32_t	Transport block size (in bytes) [3GPP TS 38.214 [5], sec 6.1.4.2]
numCb	uint16_t	Number of CBs in the TB (could be more than the number of CBs in this PUSCH transmission). Should be set to zero in any of the following conditions: 1) CBG is not supported or requested 2) newData=1 (new transmission) 3) tbSize=0
cbPresentAndPosition	uint8_t [ceil(numCb/8)]	Each bit represents if the corresponding CB is present in the current retx of the PUSCH. 1=PRESENT, 0=NOT PRESENT.

Table 3-105 Optional puschData information

Field	Type	Description
harqAckBitLength	uint16_t	Number of HARQ-ACK bits [3GPP TS 38.212 [3], section 6.3.2.4] Value: 0 -> 11 (Small block length) 12 ->1706 (Polar) Note: Does not include CRC bits
csiPart1BitLength	uint16_t	Number of CSI part1 bits [3GPP TS 38.212 [3], section 6.3.2.4] Value:



Field	Type	Description
		0 -> 11 (Small block length) 12 -> 1706 (Polar) Note: Does not include CRC bits
flagCsiPart2	uint16_t	Number of CSI part2 bits [3GPP TS 38.212 [3], section 6.3.2.4] Value: 0 -> No CSI part 2 (in this case, numPart2s = 0, if signaled in Table 3-104 for this PDU) 65535 -> Determine CSI Part2 length based on Table 3-104 (in this case, numPart2s > 0 in that table)
AlphaScaling	uint8_t	Alpha parameter, α , used to calculate number of coded modulation symbols per layer. [3GPP TS 38.212 [3], section 6.3.2.4]. Value: 0 = 0.5 1 = 0.65 2 = 0.8 3 = 1
betaOffsetHarqAck	uint8_t	Beta Offset for HARQ-ACK bits. [3GPP TS 38.212 [3], section 6.3.2.4] [3GPP TS 38.213 [4], Table 9.3-1] Value: 0->15
betaOffsetCsi1	uint8_t	Beta Offset for CSI-part1 bits. [3GPP TS 38.212 [3], section 6.3.2.4] [3GPP TS 38.213 [4], Table 9.3-2] Value: 0->18
betaOffsetCsi2	uint8_t	Beta Offset for CSI-part2 bits. [3GPP TS 38.212 [3], section 6.3.2.4] [3GPP TS 38.213 [4], Table 9.3-2] Value: 0->18

Table 3-106 Optional puschUci information

Field	Type	Description	
numPtrsPorts	uint8_t	Number of UL PTRS ports [3GPP TS 38.212 [3], sec 7.3.1.1.2] Value: 1->2	
For numPtrsPorts {			
	PTRSPortIndex	uint16_t	PT-RS antenna ports [3GPP TS 38.214 [5], sec 6.2.3.1 and 3GPP TS 38.212 [3], section 7.3.1.1.2] Bitmap occupying the 12 LSBs with: bit 0: antenna port 0 bit 11: antenna port 11 and for each bit



Field	Type	Description
		0: PTRS port not used 1: PTRS port used
PTRSDmrsPort	uint8_t	DMRS port corresponding to PTRS. Value: 0->11
PTRSReOffset	uint8_t	PT-RS resource element offset value taken from [3GPP TS 38.211 [2], table 6.4.1.2.2-1] Value: 0->11
}		
PTRSTimeDensity	uint8_t	PT-RS time density [3GPP TS 38.214 [5], table 6.2.3.1-1] Value: 0: 1 1: 2 2: 4
PTRSFreqDensity	uint8_t	PT-RS frequency density [3GPP TS 38.214 [5], table 6.2.3.1-2] Value: 0: 2 1: 4 Intel: Follow DLSCH PDU
ulPTRSPower	uint8_t	PUSCH to PT-RS power ratio per layer per RE [3GPP TS 38.214 [5], table 6.2.3.1-3] Value: 0: 0dB 1: 3dB 2: 4.77dB 3: 6dB

Table 3-107 Optional puschPtrs information

Field	Type	Description
lowPaprGroupNumber	uint8_t	Group number for Low PAPR sequence generation. [3GPP TS 38.211 [2], sec 5.2.2] For DFT-S-OFDM. Value: 0->29
lowPaprSequenceNumber	uint16_t	[3GPP TS 38.211 [2], sec 5.2.2] For DFT-S-OFDM.
ulPtrsSampleDensity	uint8_t	Number of PTRS groups. [3GPP TS 38.214 [5], sec 6.2.3.2] Value: 1->8



Field	Type	Description
ulPtrsTimeDensityTransformPrecoding	uint8_t	<p>Number of samples per PTRS group. [3GPP TS 38.214 [5], sec 6.2.3.2] [3GPP TS 38.211 [2], sec 6.4.1.2.2].</p> <p>Value: 1->4</p>

Table 3-108 Optional dftsOfdm information

Field	Type	Description
CbCrcStatusRequest	uint8_t	<p>When set to true, PHY – if supported – reports CB CRC status for CBs in cbPresentAndPosition (if numCb >0) or all CBs (if numCb = 0)</p> <p>Shall be ignored in the absence of puschData.</p> <p>Range:</p> <ul style="list-style-type: none">• 0 = false• 1 = true
UL MIMO support in FAPIv4		
srsTxPorts	uint32_t	<p>Bitmap of SRS ports to which the precoding indicated by UITpmiIndex applies.</p> <p>Same order of antenna ports as in the SRS PDU, depending on the flavor (codebook, non-codebook).</p> <p>For FAPIv3-type precoding:</p> <ul style="list-style-type: none">- this bitmap is set to all-zeroes;- UITpmiIndex – may be disregarded <p>In this release, for FAPIv4 type precoding, for instance:</p> <ul style="list-style-type: none">- Codebook-based precoding: MAC ensures that the bitmap of srsTxPorts is a subset of the srs tx ports for a the same Codebook-based SRS Resource.- Non-codebook-based precoding: MAC ensures that the bitmap of srsTxPorts corresponds to single-port non-codebook-based SRS Resources and is equal to the nrOfLayers
UITpmiIndex	uint8_t	<p>Tpmi Index, per 6.3.1.5 in 38.211.</p> <p>Actual precoder corresponding selected based on TransformPrecoding, nrOfLayers and srsTxPorts.</p> <p>Value:</p> <ul style="list-style-type: none">- 0-27: TPMI index- 255: non-codebook
NumULSpatialStreamsPorts	uint8_t	Number of ports must be at least as large as the number of layers, if PHY



Field	Type	Description
		indicates that UL spatial stream assigned is required, in capability maxNumberUISpatialStreams (TLV 0x0153).
UISpatialStreamPorts	uint16_t [n]	n = NumULSpatialStreamsPorts Spatial streams assigned for this PUSCH PDU. Value: 0 → (max # spatial streams - 1, per TLV 0x0153)

Table 3-109 PUSCH parameters V4

3.4.3.2A MsgA-PUSCH PDU

The format of the Msg-A PUSCH PDU is given in Table 3-110.

The MsgA-PUSCH PDU may includes both mandatory and optional parameters, where the presence of the optional elements is defined in the parameter pduBitmap. No optional elements are specified in this release.

Only parameters signalled by pduBitmap are optional, all other parameters are mandatory. (It should be noted that some parameters are only applicable for certain modes etc. These are still included but their value is ignored by the PHY.)

Field	Type	Description
pduBitmap	uint16_t	Bitmap indicating presence of optional PDUs All bits are reserved, in this release.
prachToPruMapType	Uint8_t	Indicates the mapping that PHY uses to determine the frequency/time/DMRS resources relevant to this MsgA-PUSCH PDU, corresponding to a MsgA-PRACH preamble. <ul style="list-style-type: none">- 0 – based on P5 Configuration (mapIndicator signals a MsgA-PUSCHConfigResIndex to the relevant P5 configuration). Relevant for PHYs that support semi-static configuration via P5- 1 – based on P7 PRACH PDU mapping (mapIndicator signals the handle). Relevant to PHYs that support dynamic configuration via P7
prachToPruMapIndicator	Uint32_t	Indicates the map for PHY to determine which MsgA-PUSCH & associated DMRS resources are in use. Value: a 16-bit P5 configuration, or a 32-bit PRACH PDU handle, as signaled by prachToPruMapType Regardless of which mapping type is signaled, PHY determines RA-RNTI from the MsgA-PRACH occasion to which the map applies, and determines RAPID from the As a result of the map, PHY can determine the following set of resources to associated parameters signaled in this PDU <ul style="list-style-type: none">- MsgA-PUSCH Time Domain index



Field	Type	Description
		<ul style="list-style-type: none">- MsgA-PUSCH Frequency Domain index- DMRS Port- DMRS Sequence (nSCID)
MsgA-PRACH SFN	uint16_t	SFN of the MsgA-PRACH PDU whose preambles map to the MsgA-PUSCH occasion signaled by this MsgA-PUSCH PDU. Value: 0 -> 1023
MsgA-PRACH Slot	uint16_t	Slot of the MsgA-PRACH PDU whose preambles map to the MsgA-PUSCH occasion signaled by this MsgA-PUSCH PDU. Value: 0 -> 159
Handle	uint32_t	An opaque handling returned in the RX_DATA.indication and/or CRC.indication message
Preamble Group Index	uint8_t	Index of the preamble group for this MsgA PUSCH PDU, for the case prachToPruMapType = 1 Set to 0 otherwise.
BWP [3GPP TS 38.213 [4], sec 12]		
BWPSize	uint16_t	If L1 uses puschTransType, this field signals the bandwidth part size [3GPP TS 38.213 [4], sec12]. Number of contiguous PRBs allocated to the BWP. If L1 does not use puschTransType, this field represents the number of VRBs to which PRBs are mapped for the allocation, according to 3GPP TS 38.211 [2], section 6.3.1.7. Note: for BWPSize, the two interpretations align, except when the allocation is scheduled for msg3 (by RAR UL grant), in which case the first interpretation uses to active BWP,i size, whereas the second one uses the initial BWP,0 size referenced by 3GPP TS 38.213 [4], section 8.3. Value: 1->275
BWPStart	uint16_t	If L1 uses puschTransType, this field signals the bandwidth part start RB index from reference CRB [3GPP TS 38.213 [4], sec 12] If L1 does not use puschTransType, this field represents the CRB corresponding to VRB0, according to 3GPP TS 38.211 [2], section 6.3.1.7. Note 1: For BWPStart, the two interpretations align, except when the allocation is scheduled for msg3 "by DCI format 0_0 with CRC scrambled by TC-RNTI in active uplink bandwidth part i starting at $N_{BWP,i}^{start}$



Field	Type	Description
		<p><i>including all resource blocks of the initial uplink bandwidth part starting at $N_{BWP,0}^{start}$, and having the same subcarrier spacing and cyclic prefix as the initial uplink bandwidth part</i>", in which case the first interpretation uses to active BWP,i start whereas the second one uses the initial BWP,0 start referenced by 3GPP TS 38.211 [2], section 6.3.1.7.</p> <p>Note 2: Only msg3 allocations subject to the conditions listed in Note 1 above are referenced to the initial BWP,0 start. Other types of msg3 allocations are referenced to the start of active uplink BWP,I, as specified in 3GPP TS 38.211 [2], section 6.3.1.7 and 3GPP TS 38.213 [4], section 8.3.</p> <p>Value: 0->274</p>
SubcarrierSpacing	uint8_t	subcarrierSpacing [3GPP TS 38.211 [2], sec 4.2] Value:0->4
CyclicPrefix	uint8_t	Cyclic prefix type [3GPP TS 38.211 [2], sec 4.2] 0: Normal; 1: Extended
PUSCH information always included		
MsgAMcs	uint8_t	The MCS index used for the MsgA PUSCH configuration and applies to all PUSCH Occasions, as signaled in 3GPP TS 38.331 [13], section 6.3.2 for this MsgA PUSCH. Value: 0 -> 15
TransformPrecoding	uint8_t	Indicates if transform precoding is enabled or disabled [3GPP TS 38.214 [5], sec 6.1.4.1] [3GPP TS 38.211 [2], 6.3.1.4] Value: 0: enabled 1: disabled
nIdMsgAPusch	uint16_t	The scrambling ID used to generate MsgA PUSCH data scramble sequence, corresponding to n_{ID} in 3GPP TS 38.211 [12], section 6.3.1.1. Note, this is used together with RAPID and RA-RNTI to determine the scrambling sequence generator. Value: 0 -> 1023
DMRS [3GPP TS 38.211 [2], sec 6.4.1.1]		
puschDmrsScramblingIds	uint16_t [2]	The values of parameter N_{ID}^0 (first entry) and N_{ID}^1 (second entry), as specified in 3GPP TS 38.211 [12], section 6.4.1.1.1 for this MsgA-PUSCH Value: 0 -> 65535



Field	Type	Description
ulDmrsSymbPos	uint16_t	DMRS symbol positions [3GPP TS 38.211 [2], sec 6.4.1.1.3 and Tables 6.4.1.1.3-3 and 6.4.1.1.3-4] Bitmap occupying the 14 LSBs with: bit 0: first symbol and for each bit 0: no DMRS 1: DMRS
dmrsMappingType	uint8_t	PUSCH mapping Type A or B, as reflected by the mappingTypeMsgA-PUSCH RRC parameter in 38.331 section 6, and further described in 38.213, section 8.1A
Pusch Allocation in frequency domain [3GPP TS 38.214 [5], sec 6.1.2.2, 3GPP TS 38.331 [13], section 6.3.2]		
startRbPuschOcas	uint16_t	The common RB index of first RB configured for first FD MSgA-PUSCH occasion within the UL slot for this MsgA-PUSCH PDU. Value: 0 -> 274 Note: the actual start Rb is determined from the parameters in this PDU frequency-domain subsection, and the frequency domain index mapped from MsgA-PRACH.
numRbPuschOcas	uint8_t	The number of PRBs that is consists of the PUSCH occasion on which all these PUSCH Resource Units reside. Value 1 to 32
guardBand	uint8_t	The guard band between two adjacent frequency domain PUSCH occasions for this MsgA PUSCH PDU in unit of PRB. Value: 0 -> 1
intraSlotFrequencyHopping	uint8_t	Indicates if intra-slot frequency hopping is enabled for this MsgA-PUSCH. [3GPP TS 38.212 [3], sec 7.3.1.1] [3GPP TS 38.214 [5], sec 6.3] Value: 0: disabled 1: enabled
intraSlotFrequencyHoppingBits	uint8_t	The value of RRC parameter msgA-HoppingBits used to determine the location of the second hop, as specified in Table 8.3-1 in 3GPP TS 38.213 [22] (per 3GPP TS 38.213 [22], section 8.1A). Values: 0 ... 3, as determined by Table 8.3-1 in 3GPP TS 38.213 [22], and the value of BWPSIZE. The value of this field is only relevant when intraSlotFrequencyHopping is enabled.
txDirectCurrentLocation	uint16_t	The uplink Tx Direct Current location for the carrier. Only values in the value range of this field between 0



Field	Type	Description
		and 3299, which indicate the subcarrier index within the carrier corresponding to the numerology of the corresponding uplink BWP and value 3300, which indicates "Outside the carrier" and value 3301, which indicates "Undetermined position within the carrier" are used. [3GPP TS 38.331 [6], UplinkTxDirectCurrentBWP IE] Value: 0->4095
uplinkFrequencyShift 7p5khz	uint8_t	Indicates whether there is 7.5 kHz shift or not. [3GPP TS 38.331 [6], UplinkTxDirectCurrentBWP IE] Value: 0: false 1: true
Resource Allocation in time domain [3GPP TS 38.214 [5], sec 5.1.2.1, 3GPP TS 38.331 [13], section 6.3.2]		
startSymbIdPuschOcas	uint8_t	The starting symbol index of the first-time domain PUSCH occasion within the UL slot for this MSA PUSCH PDU. Value: 0 -> 13 Note: The actual start symbol is determined from the parameters in this PDU time-domain subsection, and the time domain index mapped from MsgA-PRACH.
durationPuschOcas	uint8_t	the duration of all time domain PUSCH occasion symbols within the UL slot for this MsgA PUSCH PDU. Value: 1 -> 14 symbols
guardPeriod	uint8_t	The guard period between two adjacent time domain PUSCH occasions in unit of symbols of the MsgA PUSCH SCS Value: 0, 1, 2, 3
MsgA-Pusch Data Information [3GPP TS 38.214 [5], sec 6.1.4.2, 3GPP TS 38.331 [13], section 6.3.2]		
TBSIZE	uint32_t	Transport block size (in bytes) [3GPP TS 38.214 [5], sec 6.1.4.2, 3GPP TS 38.331 [13], section 6.3.2]

Table 3-110 MsgA-PUSCH PDU

3.4.3.3 PUCCH PDU

The format of the PUCCH PDU is given in Table 3-111.

The PUCCH PDU includes information regarding SR, HARQ and CSI and the valid data varies depending on whether it is PUCCH format 0, 1, 2, 3 or 4.



Field	Type	Description
RNTI	uint16_t	The RNTI used for identifying the UE when receiving the PDU Value: 1 -> 65535
Handle	uint32_t	An opaque handling returned in the UCI.indication message
BWP [3GPP TS 38.213 [4], sec 12]		
BWPSize	uint16_t	Bandwidth part size [3GPP TS 38.213 [4], sec12]. Number of contiguous PRBs allocated to the BWP Value: 1->275
BWPStart	uint16_t	Bandwidth part start RB index from reference CRB [3GPP TS 38.213 [4], sec 12] Value: 0->274
SubcarrierSpacing	uint8_t	subcarrierSpacing [3GPP TS 38.211 [2], sec 4.2] Value:0->4
CyclicPrefix	uint8_t	Cyclic prefix type [3GPP TS 38.211 [2], sec 4.2] 0: Normal; 1: Extended
FormatType	uint8_t	PUCCH Format Type [3GPP TS 38.211 [2], sec 6.3.2.1] Value: 0 ->4
multiSlotTxIndicator	uint8_t	This field is to flush, keep or combine the buffer used for multiple-slot PUCCH transmissions [3GPP TS 38.213 [4], sec 9.2.6] Value: 0: No multi slot transmission 1: Multi slot transmission starts 2: Multi slot transmission continues 3: Multi slot transmission ends This filed being '0' or '1' indicates the previous PUCCH transmission for this UE (RNTI and Handle) is ended incase L2 doesn't send '3' for the previous ongoing multi-slot transmission.



Field	Type	Description
pi2Bpsk	uint8_t	When enabled, indicates that PHY shall use pi/2 BPSK for UCI symbols instead of QPSK for PUCCH formats 3/4 for the PUCCH transmission of PUCCH PDU. [3GPP TS 38.211 [2], sec 6.3.2.6.2] Value: 0: disabled, 1: enabled
Pucch Allocation in frequency domain [3GPP TS 38.213 [4], sec 9.2.1]		
prbStart	uint16_t	The starting PRB $RB_{\text{BWP}}^{\text{offset}}$ within the BWP for this PUCCH, or first PRB prior to hopping [3GPP TS 38.213 [4], sec 9.2.1]. Valid for all formats Value: 0->274
prbSize	uint16_t	The actual number of PRBs used by the UE within this PUCCH. Valid for all formats. Value: 1 -> 16
Pucch Allocation in time domain		
StartSymbolIndex	uint8_t	Start symbol index of PUCCH from the start of the slot, S. [3GPP TS 38.213 [4], sec 9.2.2]. Valid for all formats Value: 0->13
NrOfSymbols	uint8_t	PUCCH duration in symbols [3GPP TS 38.213 [4], sec 9.2.2] Values: 1 -2: Valid for formats 0,2 4->14: Valid for formats 1,3,4
Hopping information [3GPP TS 38.211 [2], sec 6.3.2.2.1]		
intraSlotFrequencyHopping	uint8_t	Intra-slot Frequency hopping for a PUCCH resource [3GPP TS 38.211 [2], sec 6.3.2.2.1]. Valid for all formats Value: 0: disabled 1: enabled
secondHopPRB	uint16_t	Index of the first PRB after frequency hopping. Valid for all formats. Value:0->274
pucchGroupHopping	uint8_t	Signaling of group and sequence hopping for PUCCH formats 0, 1, 3 and 4 [3GPP TS 38.211 [2], sec 6.3.2.2.1].



Field	Type	Description
		<p>Value:</p> <p>0: neither, neither group nor sequence hopping is enabled</p> <p>1: enabled, enable group hopping and disable sequence hopping</p> <p>2: disable, disable group hopping and enable sequence hopping</p> <p>(note: These values are only applicable if both L1 and L2/L3 support FAPIv3, otherwise refer to SCF-222 v2, parameter groupHopFlag)</p>
obsolete8bit	uint8_t	This flag is obsolete in FAPIv3. Refer to parameter sequenceHopFlag in SCF-222 v2 if FAPIv3 is not supported.
nIdPucchHopping	uint16_t	<p>The parameter n_{ID} used for sequence hopping. [3GPP TS 38.211 [2], sec 6.3.2.2.1]</p> <p>Valid for formats 0, 1, 3 and 4.</p> <p>Value: 0->1023</p>
InitialCyclicShift	uint16_t	<p>Initial cyclic shift (M_0) used as part of frequency hopping. [3GPP TS 38.213 [4], sec 9.2.1 and 3GPP TS 38.211 [2], sec 6.3.2.2.2].</p> <p>Valid for formats 0, 1, 3 and 4</p> <p>Value: 0->11</p>
nIdPucchScrambling	uint16_t	<p>parameter n_{ID} in [3GPP TS 38.211 [2], sec 6.3.2.5.1 and 6.3.2.6.1]</p> <p>Valid for formats 2, 3 and 4.</p> <p>Value: 0->1023</p>
TimeDomainOccIdx	uint8_t	<p>An index of an orthogonal cover code [3GPP TS 38.211 [2], sec 6.3.2.4.1].</p> <p>Valid for format 1.</p> <p>Value: 0->6</p>
PreDftOccIdx	uint8_t	<p>An index of an orthogonal cover code. [3GPP TS 38.211 [2], sec 6.3.2.6.3].</p> <p>Valid for format 4.</p> <p>Value: 0->3</p>
PreDftOccLen	uint8_t	<p>A length of an orthogonal cover code. [3GPP TS 38.211 [2], sec 6.3.2.6.3].</p> <p>Valid for format 4.</p> <p>Value: 2 or 4</p>
DMRS [3GPP TS 38.211 [2], sec 6.4.1.3]		



Field	Type	Description
AddDmrsFlag	uint8_t	<p>Flag for additional DMRS. [3GPP TS 38.213 [4], sec 9.2.2]. Valid for formats 3 and 4.</p> <p>Value: 0 = disabled 1 = enabled</p>
nId0PucchDmrsScrambling	uint16_t	<p>nID0, as defined for format 2 in 38.211 [2] (Rel-15), section 6.4.1.3.2.1 and referenced in 38.211 [12] (Rel-16) section 6.4.1.3.3.1 for formats 3 and 4</p> <p>For formats 3/4, PHY shall use if and only if pi2Bpsk is enabled</p> <p>Value: 0->65535</p>
m0PucchDmrsCyclicShift	uint8_t	<p>m0, as defined for referenced for formats 3 and 4 in 38.211, section 6.4.1.3.3.1 [3GPP TS 38.211 [2], sec 6.4.1.3.3.1]</p> <p>PHY shall use if and only if pi2Bpsk is disabled</p> <p>Value: 0 ->9</p>
SrBitLen	uint8_t	<p>Indicates the number of SR bits expected in the PUCCH transmission.</p> <p>Valid for all PUCCH formats</p> <p>For PUCCH Format 0/1: Value: 0 = no SR 1 = SR occasion</p> <p>For PUCCH Format 2/3/4: Value: 0 = no SR; 1 = one SR bit; 2 = two SR bits; 3 = three SR bits; 4 = four SR bits</p> <p>Note: When MAC is implemented according to 3GPP TS 38.213 [4][22], the maximum number of SR bits expected for a single UE is 4.</p>
BitLenHarq	uint16_t	<p>Bit length of HARQ payload</p> <p>Valid for all formats.</p> <p>Value: 0 = no HARQ bits 1->2 = Valid for Formats 0 and 1 2 -> 1706 = Valid for Formats 2, 3 and 4</p>



Field	Type	Description
csiPart1BitLength	uint16_t	Bit length of CSI part 1 payload. Valid for formats 2, 3 and 4. Value: 0 = no CSI bits 1 -> 1706
Beamforming		
Beamforming	structure	See Table 3-117; for PHYs supporting MU-MIMO groups, see also Table 3-118
Extension TLVs		
PUCCH Maintenance Parameters added in FAPIv3	structure	See Table 3-112
UCI part1 to part-2 correspondence added in FAPIv3	structure	See Table 3-104
Parameters added in FAPIv4	structure	See Table 3-113

Table 3-111 PUCCH PDU

Field	Type	Description
In this specification version, L2 always includes this TLV to extend PUCCH PDU		
maxCodeRate	uint8_t	<p>Max coding rate to determine how to feedback UCI on PUCCH for format 2, 3 or 4 [3GPP TS 38.213 [4], Table 9.2.5.2-1] Value Range:</p> <ul style="list-style-type: none"> - 0 -> 7: max code rate per Table 9.2.5.2-1 of 3GPP TS 28.213 [4] - 255: not applicable (e.g., formats 0 or 1) <p>Note: In 3GPP TS 38.213 [4], value 7 is reserved</p>
ulBwpId	uint8_t	<p>Index of PUCCH group and sequence hopping parameters to use for this PUCCH PDU. Value: 255: L1 not configured with semi-static signaling of PUCCH group and sequence hopping parameters <255: PHY may assume that the parameters linked to this UL-BWP-ID in the PUCCH Semi-Static Configuration Table 3-55 apply.</p>

Table 3-112 PUCCH basic extension for FAPIv3

Field	Type	Description
In this specification version, L2 always includes this TLV to extend PUCCH PDU		
uciReportFormat	uint8_t	<p>Flag to indicate if joint reporting of different type of UCI payload is required or not. Range: 0: separate reporting (PHY reports HARQ/SR/CSIP1/CSIP2 in separate parameters)</p>



Field	Type	Description
		1: combined reporting (PHY reports the uciPayload(s), without attempting to interpret its mapping to HARQ, SR, CSIp1 or CSIp2 bits) PHY support for this flag is signaled in uciReportFormatPucchForamt234 TLV of PARAM.response
For MU-MIMO support in FAPIv4		
NumULSpatialStreamsPorts	uint8_t	Number of spatial streams used to signal this PUCCH allocation. Non-zero if PHY indicates that UL spatial stream assigned is required, in capability maxNumberUISpatialStreams (TLV 0x0153).
UISpatialStreamPorts	uint16_t [n]	n = NumULSpatialStreamsPorts Spatial streams assigned for this PUSCH PDU. Value: 0 → (max # spatial streams - 1, per TLV 0x0153)

Table 3-113 PUCCH basic extension for FAPIv4

3.4.3.4 SRS PDU

The format of the SRS PDU is shown in Table 3-114.

Field	Type	Description
RNTI	uint16_t	UE RNTI Value: 1->65535
Handle	uint32_t	An opaque handling returned in the SRS.indication
BWP [3GPP TS 38.213 [4], sec 12]		
BWPSize	uint16_t	Bandwidth part size [3GPP TS 38.213 [4], sec 12]. Number of contiguous PRBs allocated to the BWP Value: 1->275
BWPStart	uint16_t	Bandwidth part start RB index from reference CRB [3GPP TS 38.213 [4], sec 12] Value: 0->274
SubcarrierSpacing	uint8_t	subcarrierSpacing [3GPP TS 38.211 [2], sec 4.2] Value:0->4
CyclicPrefix	uint8_t	Cyclic prefix type [3GPP TS 38.211 [2], sec 4.2] 0: Normal; 1: Extended
numAntPorts	uint8_t	Number of antenna ports N_{ap}^{SRS} [3GPP TS 38.211 [2], Sec 6.4.1.4.1] Value: 0 = 1 port 1 = 2 ports 2 = 4 ports



Field	Type	Description
numSymbols	uint8_t	Number of symbols $N_{\text{symb}}^{\text{SRS}}$ [3GPP TS 38.211 [2], Sec 6.4.1.4.1] Value: 0 = 1 symbol 1 = 2 symbols 2 = 4 symbols 3 = 12 symbols
numRepetitions	uint8_t	Repetition factor R [3GPP TS 38.211 [2], Sec 6.4.1.4.3] Value: 0 = 1 1 = 2 2 = 4
timeStartPosition	uint8_t	Starting position in the time domain l_0 [3GPP TS 38.211 [2], Sec 6.4.1.4.1] Note: the MAC undertakes the translation from startPosition to l_0 Value: 0 → 13
configIndex	uint8_t	SRS bandwidth config index C_{SRS} [3GPP TS 38.211 [2], Sec 6.4.1.4.3] Value: 0 → 63
sequenceId	uint16_t	SRS sequence ID $n_{\text{ID}}^{\text{SRS}}$ [3GPP TS 38.211 [2], Sec 6.4.1.4.2] Value: 0 → 65535
bandwidthIndex	uint8_t	SRS bandwidth index B_{SRS} [3GPP TS 38.211 [2], Sec 6.4.1.4.3] Value: 0 → 3
combSize	uint8_t	Transmission comb size K_{TC} [3GPP TS 38.211 [2], Sec 6.4.1.4.2] Value: 0 = comb size 2 1 = comb size 4 2 = comb size 8 (Rel16)
combOffset	uint8_t	Transmission comb offset \bar{k}_{TC} [3GPP TS 38.211 [2], Sec 6.4.1.4.3] Value: 0 → 1 (combSize = 0) Value: 0 → 3 (combSize = 1) Value: 0 → 7 (combSize = 2)
cyclicShift	uint8_t	Cyclic shift $n_{\text{SRS}}^{\text{CS}}$ [3GPP TS 38.211 [2], Sec 6.4.1.4.2] Value: 0 → 7 (combSize = 0) Value: 0 → 11 (combSize = 1) Value: 0 → 5 (combSize = 2)
frequencyPosition	uint8_t	Frequency domain position n_{RRC} [3GPP TS 38.211 [2], Sec 6.4.1.4.3]



Field	Type	Description
		Value: 0 → 67
frequencyShift	uint16_t	Frequency domain shift n_{shift} [3GPP TS 38.211 [2], Sec 6.4.1.4.3] Value: 0 → 268
frequencyHopping	uint8_t	Frequency hopping b_{hop} [3GPP TS 38.211 [2], Sec 6.4.1.4.3] Value: 0 → 3
groupOrSequenceHopping	uint8_t	Group or sequence hopping configuration (RRC parameter groupOrSequenceHopping in SRS-Resource IE) Value: 0 = No hopping 1 = Group hopping groupOrSequenceHopping 2 = Sequence hopping
resourceType	uint8_t	Type of SRS resource allocation [3GPP TS 38.211 [2], Sec 6.4.1.4.3] Value: 0: aperiodic 1: semi-persistent 2: periodic
Tsrs	uint16_t	SRS-Periodicity in slots [3GPP TS 38.211 [2], Sec 6.4.1.4.4] Value: 1,2,3,4,5,8,10,16,20,32,40,64,80,160,320,640,128,0,2560
Toffset	uint16_t	Slot offset value [3GPP TS 38.211 [2], Sec 6.4.1.4.3] Value: 0->2559
Beamforming		
Beamforming	structure	See Table 3-117; for PHYs supporting MU-MIMO groups, see also Table 3-118
SRS Parameters added in FAPIv4	structure	See Table 3-115
SRS Parameters added in FAPIv5	structure	See Table 3-116

Table 3-114



Field	Type	Description
srsBandwidthSize	uint16_t	$m_{SRS,b}$: Number of PRB's that are sounded for each SRS symbol, per 3GPP TS 38.211 [2], section 6.4.1.4.3; Value: 4->272
For each symbol in numSymbols		
srsBandwidthStart	uint16_t	PRB index for the start of SRS signal transmission. The PRB index is relative to the CRB0 or reference Point A. 3GPP TS 38.211 [2], section 6.4.1.4.3; Value: 0->268
sequenceGroup	uint8_t	Sequence group (u) as defined in 3GPP TS 38.211 [2], section 6.4.1.4.2 Value: 0->29
sequenceNumber	uint8_t	Sequence number (v) as defined in 3GPP TS 38.211 [2], section 6.4.1.4.2 TS 38.211 Value: 0->1
SRS Usage	uint32	Indicates the usage of the SRS Resource signaled by this PDU. Value: <ul style="list-style-type: none">- 0 – beamManagement- 1 – codebook- 2 – nonCodebook- 3 – antennaSwitching- 4 – positioning- 5 – 255: reserved. Note: <ul style="list-style-type: none">- Capability TLV 0x0080 (supportedSrsUsage) signals what usages are supported by PHY.
ReportType	uint32	Bitmap of the requested Report Type, based on this SRS PDU. Report types are represented as bit positions: <ul style="list-style-type: none">• [0] Per-PRG and Symbol SNR• [1] Normalized Channel I/Q Matrix; Note:<ul style="list-style-type: none">◦ if requested for <i>codebook</i> Usage, represents PRG I and Q channel estimate, per srs Tx port and gNB antenna element◦ If requested for <i>nonCodebook</i> Usage, represents PRG I and Q channel estimate, per SRI and gNB antenna element• [2] Channel SVD Representation; Note:<ul style="list-style-type: none">◦ If requested for <i>antennaSwitching</i> Usage, it is the SVD representation of UE Rx and gNB sets of antenna element• [3] Positioning



Field	Type	Description
		<ul style="list-style-type: none">[4-31] Reserved <p>Value:</p> <ul style="list-style-type: none">0: This report is not requested1: This report is requested <p>Notes:</p> <ul style="list-style-type: none">- if all bits are set to 0, no report is required.- per-Usage allowed reports are indicated by L1 in Capability TLV 0x0160 (allowedSrsReportsPerUsage)
singular Value Representation	Uint8_t	0 – 8-bit dB 1 – 16-bit linear 255 – not applicable See Table 3-143 Other values reserved Other values reserved
iq Representation	Uint8_t	0 – 16 bit 1 – 32-bit 255 – not applicable See Table 3-142 and Table 3-143 Other values reserved
prgSize	Uint16	1-272 0 – reserved
numTotalUeAntennas	uint8_t	1 ... 16 in this release. This is the total number of UE antennas being sampled for the SRS Usage. Notes: <ul style="list-style-type: none">- a UE's antennas may be sampled via one or multiple SRS PDU- the subset of <i>numTotalUeAntennas</i> antennas sampled by this SRS PDU is indicated in <i>sampledUeAntennas</i>.
sampledUeAntennas	uint32_t	Bitmap of UE antenna indices sampled by the SRS waveform corresponding to this PDU's SRS Resource. <ul style="list-style-type: none">• Codebook: corresponds to antenna ports in SRS Resource; non-overlapping indices are expected for ports sampled via different SRIs, for a given UE. [3GPP TS 38.214 [5], section 6.1.1.2]• Non-codebook: corresponds to SRIs [3GPP TS 38.214 [5], section 6.1.1.2]• Antennas-switch: indices of UE Rx antennas in the total number of antennas. Depending on the UE capability, multiple SRS PDUs may be needed to sample all UE Rx antennas for each PRG. [3GPP TS 38.213 [4], section 6.2.1.2] Notes: <ul style="list-style-type: none">- <i>numAntPorts</i> bits are set to '1' in <i>sampledUeAntennas</i>.



Field	Type	Description
		<ul style="list-style-type: none"> - only bits in positions 0 ... ($numTotalUeAntennas-1$) may be set to '1' to indicate which of the $sampleUeAntennas$ ports for the SRS usage are sampled in this SRS PDU for this UE. - across all SRS PDUs for the <i>SRS Usage</i> for this UE, it is expected that all of the $numTotalUeAntennas$ will be sampled for all PRBs in the SRS bandwidth.
reportScope	uint8_t	<p>Which antennas Report (in ReportType) should account for:</p> <p>Value:</p> <ul style="list-style-type: none"> 0: ports in sampledUeAntennas (i.e. SRS Resource) 1: all numTotalUeAntennas ports for the SRS usage. <p>For <i>SRS Usage</i> = <i>antennaSwith</i> reports of SVD type for , value 0 is only allowed if the number of bits set in <i>sampledUeAntennas</i> = <i>numTotalUeAntennas</i></p>
MU-MIMO support for FAPIv4		
NumULSpatialStreamsPort s	uint8_t	In this release, L2 may set this number to 0 to leave spatial stream index assignment to L1 (e.g. L1 uses spatial streams reserved for SRS), regardless of the information in capability maxNumberUISpatialStreams (TLV 0x0153)
UISpatialStreamPorts	uint16_t [n]	<p>n = NumULSpatialStreamsPorts</p> <p>Numbers of ports used for signaling this SRS allocation.</p> <p>Value: 0 → (max # spatial streams - 1, per TLV 0x0153, or 65535 if the TLV is absent)</p>

Table 3-115 FAPIv4 SRS Parameters

Field	Type	Description
symbolPuncturingBitmap	uint16_t	<p>A bitmap of SRS symbols that are punctured</p> <p>For each bit:</p> <ul style="list-style-type: none"> • 0: symbol punctured • 1: symbol transmitted <p>Bit positions:</p> <ul style="list-style-type: none"> • [0]: symbol #0 • [1]: symbol #1 • ... • ... • [14]: reserved (set to 0, in this release) • [15]: reserved (set to 0, in this release) <p>Note: this bitmaps serves for handling PUSCH overlaps with SRS for positioning</p>

Table 3-116 FAPIv5 SRS Parameters



3.4.3.5 Rx Beamforming PDU

The beamforming PDU is included in the PUCCH, PUSCH, PRACH and SRS PDUs. The format is shown in Table 3-117.

Field	Type	Description			
TRP scheme	uint8_t	This field shall be set to 0, to identify that this table is used.			
numPRGs	uint16_t	Number of PRGs spanning this allocation. Value : 1->275			
prgSize	uint16_t	Size in RBs of a precoding resource block group (PRG) - to which the same digital beamforming gets applied. Value: 1->275			
digBFIInterface	uint8_t	Number of logical antenna ports (parallel streams) resulting from the Rx combining Value: 0->255			
For number of PRGs {					
for each digBFIInterface {					
	beamIdx	uint16_t	Index of the digital beam weight vector pre-stored at cell configuration. The vector maps baseband ports to the logical port corresponding to dibBFIInterface. Value: 0->65535		
	}				
			}		

Table 3-117 Rx beamforming PDU

3.4.3.6 Rx Beamforming PDU for SRS-based combining

In this release, this PDU may be included when UL combining is performed as described in section 2.2.6.2. The format is shown in Table 3-95.

Field	Type	Description	
TRP scheme	uint8_t	5: Single TRP, based on SRS	
numMuMimo Groups	uint8_t	Number of UL MuMIMO groups covering the allocation of this PDU. Any PRBs in the allocation for the channel must belong to exactly one MuMIMO group. For SU-MIMO, the number of groups may be 0.	
For each of the numMuMimoGroups {			
	muMimoGroupIdx	uint16_t	muMimoGroups are listed in increasing muMimoGroup index order. A PRG is defined in exactly one muMimoGroup. Value: 0→3,821



Field	Type	Description
		Note: <i>muMimoGroupIdx</i> are defined in increasing order of their start symbol and on a given start symbol they are defined in increasing order of PRB
{}		

Table 3-118 SRS-based Rx beamforming PDU

3.4.4 UL_DCI.request

The `UL_DCI.request` message includes DCI content used for the scheduling of PUSCH, except when precoding for UL grants is based on section 3.4.2.7; in this latter case, MAC shall transmit UL_DCIs for the grants in the `DL_TTI.request` message, instead. The `UL_DCI.request` message format is shown in Table 3-119.

The separation of DL and UL DCIs introduces greater flexibility into the design and implementation of the MAC scheduler. Specifically, it permits separate DL (generates DL DCI) and UL (generates UL DCI) scheduling and removes the requirement to integrate the UL relevant information with the DL control in `DL_TTI.request`.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
numPDUs	uint16_t	Number of PDCCH PDUs that are included in this message. Value 0 -> 65535
nDITypes	uint8_t	Maximum number of DL PDU types or DCIs supported by <code>UL_DCI.request</code> . nDIPduTypes = 2 in this release. Backward compatibility note: SCF222 v2.0 did not have this field
nPDUsOfEachType	uint16_t [nDITypes]	Number of PDUs of each type that are included in this message. Each array entry corresponds to a PDU type as follows: [0]: number of PDCCH PDUs [1]: number of DCIs across all PDCCH PDUs in this message Value 0 -> 65535, for each entry Backward compatibility note: SCF222 v2.0 did not have this field
For Number of PDUs {		
PDUType	uint16_t	0: PDCCH PDU
PDUSize	uint16_t	Size of the PDU control information (in bytes). This length value includes the 4 bytes required for the PDU type and PDU size parameters.



Field	Type	Description
		Value 0 -> 65535
PDCCH PDU Configuration	structure	See Section 3.4.2.1
}		

Table 3-119 `UL_DCI.request` message body

3.4.5 SLOT errors

The error codes returned in an `ERROR.indication` generated by the `DL_TTI.request` message are given in Table 3-120.

Error code	Description
MSG_INVALID_STATE	The <code>DL_TTI.request</code> was received when the PHY was in the IDLE or CONFIGURED state.
OUT_OF_SYNC	The <code>DL_TTI.request</code> was received with a different SFN/SL than the PHY expected. The PHY has followed the SFN/SL sync process, see Section 2.2.2.
MSG_BCH_MISSING	A SSB PDU was expected in the <code>DL_TTI.request</code> message for this subframe.
MSG_SLOT_ERR	The <code>DL_TTI.request</code> had an invalid format.

Table 3-120 Error codes for `ERROR.indication` generated by `DL_TTI.request`

The error codes returned in an `ERROR.indication` generate by the `UL_TTI.request` message are given in Table 3-121.

Error code	Description
MSG_INVALID_STATE	The <code>UL_TTI.request</code> was received when the PHY was in the IDLE or CONFIGURED state.
MSG_SLOT_ERR	The <code>UL_TTI.request</code> had an invalid format.

Table 3-121 Error codes for `ERROR.indication` generated by `UL_TTI.request`

The error codes returned in an `ERROR.indication` generate by the `UL_DCI.request` message are given in Table 3-122.

Error code	Description
MSG_INVALID_STATE	The <code>UL_DCI.request</code> was received when the PHY was in the IDLE or CONFIGURED state.
MSG_INVALID_SFN	The <code>UL_DCI.request</code> message received in subframe N included a SFN/SF value which was not N-1. The message has been ignored.
MSG_UL_DCI_ERR	The <code>UL_DCI.request</code> had an invalid format.

Table 3-122 Error codes for `ERROR.indication` generated by `UL_DCI.request`

3.4.6 TX_DATA.request

The format of the `TX_DATA.request` message is described in Table 3-123. This message contains the MAC PDU data for transmission over the air interface. The PDUs described in this message must follow the same order as `DL_TTI.request`.



This message can be sent by the L2/L3 when the PHY is in the RUNNING state. If it is sent when the PHY is in the IDLE, or CONFIGURED, state an `ERROR.indication` message will be sent by the PHY.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
Control Length	uint16_t	Length (in bytes) of control plane portion, including any padding bytes, if tag value 3 is used for TBS-TLV (i.e. CP/UP separation), may be zero otherwise. See Table 3-30 for the related capability. It is measured from the first byte of this message until then end of the control portion.
Number of PDUs	uint16_t	Number of PDUs included in this message. Value 0-> MaxDIPDUsPerSlot
For each PDU {		
PDU length	uint32_t	The total length (in bytes) of the PDU description and PDU data, without the padding bytes.
PDU index	uint16_t	Used to correlate the MAC PDU with the DL_TTI PDSCH PDU Value: 0 → 65535
CW index	uint8_t	The CW in PDSCH PDU with PDU index, to which this MAC PDU corresponds. Value: 0, 1
numTLV	uint32_t	The number of TLVs describing the data of the transport block. Value: 0 -> MaxTLVs Shall be set to 1 if tag values 0 or 3 are used in TLVs
TLVs	TBS-TLV [numTLV]	See Table 3-124
}		

Table 3-123 TX_DATA.request message



Field	Type	Description
Tag	uint16_t	Value: 0 -> 3 0: Payload is carried directly in the value field 1: 32-bit pointer to payload is in the value field 2: 32-bit offset from first address 3: offset from the end of control portion 4: 64-bit pointer to payload is in the value field 5: 64-bit offset from first address
Length	uint32_t	Length of the actual payload in bytes, without the padding bytes
Value	Variable or uint32_t	32 bits, or a multiple of the number of bits indicated by the pdschMacPduBitsAlignment capability (default 32-bits). Tag=0: Only the most significant bytes of the size indicated by 'length' field are valid. Remaining bytes are zero padded to the nearest bit boundary indicated by the pdschMacPduBitsAlignment capability (default 32-bit) Tag=1/4: Pointer to the payload. Occupies 32-bits for tag 1 and 64 bits for tag 4. Tag=2/5: Offset from the "first address" to the payload is in the value field. Where the first address is a predefined value. Occupies 32-bits for tag 1 and 64 bits for tag 5. Tag=3: Offset from the end of control portion to the payload is in the value field, where the end of control portion is indicated by Control Length field in TX_DATA.request. Occupies 32-bits. Only the most significant bytes of the size indicated by 'length' field are valid. Remaining bytes are zero padded to the nearest bit boundary indicated by the pdschMacPduBitsAlignment capability (default 32-bit)

Table 3-124 TBS-TLV structure for TX_DATA.request

3.4.6.1 Downlink Data Errors

The error codes returned in an `ERROR.indication` generate by the `TX_DATA.request` message are given in Table 3-125.

Error code	Description
MSG_INVALID_STATE	The <code>TX_DATA.request</code> was received when the PHY was in the IDLE or CONFIGURED state.
MSG_INVALID_SFN	The <code>TX_DATA.request</code> message received in subframe N included a SFN/SF value which was not N. The message has been ignored.
MSG_TX_ERR	The <code>TX_DATA.request</code> had an invalid format.

Table 3-125 Error codes for `ERROR.indication`



3.4.7 RX_DATA.indication

The format of the RX_DATA.indication message is shown in Table 3-126.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
Control Length	uint16_t	Length (in bytes) of control plane portion, including any padding bytes, if Uplink UP/CP separation, may be zero otherwise. See Table 3-30 for the related capability. It is measured from the first byte of this message until then end of the control portion.
Number of PDUs	uint16_t	Number of PDUs included in this message. Only successfully-decoded PDUs are included. RX_DATA.indication shall be sent even if all TBS for the slot have failed to decode, in which case this field is set to 0. Range 0-> MaxULPDUsPerSlot
For each successfully-decoded PDU {		
Handle	uint32_t	The handle passed to the PHY in an UL_TTI.request PUSCH PDU or MsgA-PUSCH PDU.
RNTI	uint16_t	The RNTI passed to the PHY in an UL_TTI.request PUSCH PDU or MsgA-PUSCH PDU. If Handle corresponds to a MsgA-PUSCH message, this field indicates the RA-RNTI associated with the received PDU. Value: 1 → 65535.
RAPID	uint8_t	If Handle corresponds to a MsgA-PUSCH message, this field indicates the RAPID associated with the received PDU, otherwise it is set to 255. Value 0 -> 63, or 255
HarqID	uint8_t	HARQ process ID Value: 0->15
PDU Length	uint32_t	Length of PDU in bytes.
numTLV	uint32_t	The number of TLVs describing the data of the transport block. Value: 0 -> MaxTLVs Shall be set to 1 if tag values 0 or 3 are used in TLVs



Field	Type	Description
TLVs	TBS-TLV [numTLV]	See Table 3-124
}		

Table 3-126 RX_DATA.indication message body

Field	Type	Description
Tag	uint16_t	Value: 0 -> 3 0: Payload is carried directly in the value field 1: 32-bit pointer to payload is in the value field 2: 32-bit offset from first address 3: offset from the end of control portion 1: 64-bit pointer to payload is in the value field 2: 64-bit offset from first address
Length	uint32_t	Length of the actual payload in bytes, without the padding bytes
Value	Variable or uint32_t	32 bits, or a multiple of the number of bits indicated by the puschMacPduBitAlignment capability (default 32-bits). Tag=0: Only the most significant bytes of the size indicated by 'length' field are valid. Remaining bytes are zero padded to the nearest bit boundary indicated by the puschMacPduBitAlignment capability (default 32-bit) Tag=1/4: Pointer to the payload. Occupies 32-bits for tag 1 and 64 bits for tag 2. Tag=2/5: Offset from the "first address" to the payload is in the value field. Where the first address is a predefined value. Occupies 32-bits for tag 2 and 64 bits for tag 5. Tag=3: Offset from the end of control portion to the payload is in the value field, where the end of control portion is indicated by Control Length field in RX_DATA.indication. Occupies 32-bits. Only the most significant bytes of the size indicated by 'length' field are valid. Remaining bytes are zero padded to the nearest bit boundary indicated by the puschMacPduBitAlignment capability (default 32-bit)

Table 3-127 TBS-TLV structure for RX_DATA.indication

3.4.8 CRC.indication

The format of the CRC.indication message is shown in Table 3-128. There can be more than one CRC.indication messages per slot.



Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
NumCRCs	uint16_t	Number of CRCs (PDUs) with status indication included in this message. Range 0-> MaxULPDUsPerSlot
For each CRC {		
Handle	uint32_t	The handle passed to the PHY in an UL_TTI.request PUSCH PDU or MsgA-PUSCH PDU.
RNTI	uint16_t	The RNTI passed to the PHY in an UL_TTI.request PUSCH PDU or MsgA-PUSCH PDU. If Handle corresponds to a MsgA-PUSCH message, this field indicates the RA-RNTI associated with the received PDU, otherwise it is set to 255. Value: 1 → 65535
RAPID	uint8_t	If Handle corresponds to a MsgA-PUSCH message, this field indicates the RAPID associated with the received PDU, otherwise it is set to 255. Value 0 -> 63, or 255
HarqID	uint8_t	HARQ process ID Value: 0->15
TbCrcStatus	uint8_t	Indicates CRC result on TB data. Value: 0 = pass 1 = fail
NumCb	uint16_t	If CB CRC status is not reported, this parameter is set to zero. Otherwise the number of CBs requested in PUSCH PDU. Note: in case of initial transmission or for non CBG Re-Tx is not used, this is the total number of CBs, otherwise (CBG Re-Tx) this is the number of CBs actually scheduled.



Field	Type	Description
		Value: 0->65535
CbCrcStatus	uint8_t [ceil(NumCb/8)]	If NumCb=0 this field is not present. Otherwise each bit indicates CRC result on CB data. Value: 0 = pass 1 = fail Note: for the special case where the CW is composed of a single CB, PHY may indicate NumCb=1 with CbCrcStatus set to the TB CRC status
UISinrMetric	int16_t	A metric of channel quality. Up to PHY implementation whether this is Signal-to-Thermal, Signal-to-Interference+Thermal or any other reasonable equivalent interpretation. Value: - 65.534 dB ... +65.534 dB in steps of 0.002 dB (0 corresponds to 0 dB). 0xFFFF = -32768 = invalid.
Timing advance offset	uint16_t	Timing advance T_A measured for the UE in multiples of $16 * 64 * T_c / 2^\mu$ [3GPP TS 38.213 [4], Section 4.2] Value: 0 → 63 0xffff should be set if this field is invalid
Timing advance offset in nanoseconds	int16_t	Timing advance measured for the UE between the reference uplink time and the observed arrival time for the UE Value: - 16800 ... +16800 nanoseconds. 0xffff should be set if this field is invalid
RSSI	uint16_t	RSSI. See Table 3-23 for RSSI definition. If RSSI is reported in dBFS (see PARAM and CONFIG measurement TLVs) then this value represents -128dBFS to 0dBFS with a step size of 0.1dB If RSSI is reported in dBm (see PARAM and CONFIG measurement TLVs) then this value represents -128dBm to 0dB, with a step size of 0.1dB



Field	Type	Description
		Value: 0-1280 0xffff should be set if this field is invalid
RSRP	uint16_t	RSRP. See Table 3-23 for RSRP definition. If RSRP is reported in dBFS (see PARAM and CONFIG measurement TLVs) then this value represents -128dBFS to 0dBFS with a step size of 0.1dB If RSRP is reported in dBm (see PARAM and CONFIG measurement TLVs) then this value represents -140dBm to -12dBm, with a step size of 0.1dB Value: 0-1280 0xffff should be set if this field is invalid
}		

Table 3-128 CRC.indication message body

3.4.9 UCI.indication

The format of the UCI.indication message is shown in Table 3-129.

The UCI.indication message includes optional parameters and the same message is used for UCI received on both PUSCH and PUCCH. There can be more than one UCI.indication message per slot.

The format of the UCI.indication message is dependent on whether the UCI is transmitted on:

- PUSCH
- PUCCH Format 0 or 1
- PUCCH Format 2,3 or 4

The optional parameters are included based on information expected on UCI.

- SR parameters are included if a PUCCH PDU included as SR opportunity
- HARQ parameters are included if PUCCH or PUSCH PDU included HARQ
- CSI part 1 parameters are included if PUCCH or PUSCH PDU included CSI
- CSI part 2 parameters are included if PUCCH or PUSCH PDU included CSI part 2

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159



Field	Type	Description
Control Length	uint16_t	Length (in bytes) of control plane portion, including any padding bytes, if tag value 3 is used for TBS-TLV (i.e. CP/UP separation), may be zero otherwise. See Table 3-30 for the related capability. It is measured from the first byte of this message until then end of the control portion.
NumUCIs	uint16_t	Number of UCIs included in this message. Range 0-> MaxUCIsPerSlot
For each UCI {		
PDUType	uint16_t	0: UCI indication PDU carried on PUSCH, see Section 3.4.9.1. 1: UCI indication PDU carried on PUCCH Format 0 or 1, see Section 3.4.9.2. 2: UCI indication PDU carried on PUCCH Format 2, 3 or 4, see Section 3.4.9.3.
PDUSize	uint16_t	Size of the PDU information (in bytes). This length value includes the 4 bytes required for the PDU type and PDU size parameters. Value 0 -> 65535
UCI PDU Information	structure	See Sections 3.4.9.1 to 3.4.9.3.
}		

Table 3-129 UCI.indication message body

3.4.9.1 PUSCH PDU

The format of a PUSCH PDU is shown in Table 3-130 and used for UCI received on the PUSCH.

Field	Type	Description
pduBitmap	uint8_t	Bitmap indicating presence of optional PDUs. Value: 0 = not present, 1 = present Bit 0: not used Bit 1: HARQ Bit 2: CSI Part 1 Bit 3: CSI Part 2
Handle	uint32_t	The handle passed to the PHY in an <code>UL_TTI.request</code> PUSCH PDU.
RNTI	uint16_t	The RNTI passed to the PHY in an <code>UL_TTI.request</code> PUSCH PDU. Value: 1 → 65535.
UISnrMetric	int16_t	A metric of channel quality. Up to PHY implementation whether this is Signal-to-Thermal, Signal-to-



Field	Type	Description
		Interference+Thermal or any other reasonable equivalent interpretation. Value: - 65.534 dB ... +65.534 dB in steps of 0.002 dB (0 corresponds to 0 dB). 0xFFFF = -32768 = invalid.
Timing advance offset	uint16_t	Timing advance T_A measured for the UE in multiples of $16 * 64 * T_c / 2^{\mu}$ [[3GPP TS 38.213 [4], Section 4.2] Value: 0 → 63 0xffff should be set if this field is invalid
Timing advance offset in nanoseconds	int16_t	Timing advance measured for the UE between the reference uplink time and the observed arrival time for the UE Value: - 16800 ... +16800 nanoseconds. 0xffff should be set if this field is invalid
RSSI	uint16_t	RSSI. See Table 3-23 for RSSI definition. If RSSI is reported in dBFS (see PARAM and CONFIG measurement TLVs) then this value represents -128dBFS to 0dBFS with a step size of 0.1dB If RSSI is reported in dBm (see PARAM and CONFIG measurement TLVs) then this value represents -128dBm to 0dB, with a step size of 0.1dB Value: 0-1280 0xffff should be set if this field is invalid
RSRP	uint16_t	RSRP. See Table 3-23 for RSRP definition. If RSRP is reported in dBFS (see PARAM and CONFIG measurement TLVs) then this value represents -128dBFS to 0dBFS with a step size of 0.1dB If RSRP is reported in dBm (see PARAM and CONFIG measurement TLVs) then this value represents -140dBm to -12dBm, with a step size of 0.1dB Value: 0-1280 0xffff should be set if this field is invalid
HARQ information	structure	Included if indicated by pduBitmap. See Table 3-136 for details
CSI part1 information	structure	Included if indicated by pduBitmap. See Table 3-137 for details
CSI part2 information	structure	Included if indicated by pduBitmap. See Table 3-138 for details

Table 3-130 UCI PUSCH PDU

3.4.9.2 PUCCH PDU Format 0/1

The format of a PUCCH PDU Format 0/1 is shown in Table 3-131 and used for UCI received on the PUCCH with either Format 0 or 1.

Field	Type	Description
pduBitmap	uint8_t	Bitmap indicating presence of optional PDUs. Value: 0 = not present, 1 = present



Field	Type	Description
		Bit 0: SR Bit 1: HARQ
Handle	uint32_t	The handle passed to the PHY in an <code>UL_TTI.request</code> PUCCH PDU.
RNTI	uint16_t	The RNTI passed to the PHY in an <code>UL_TTI.request</code> PUCCH PDU. Value: 1 → 65535.
PucchFormat	uint8_t	PUCCH format Value: 0 → 1 0: PUCCH Format0 1: PUCCH Format1
UISinrMetric	int16_t	A metric of channel quality. Up to PHY implementation whether this is Signal-to-Thermal, Signal-to-Interference+Thermal or any other reasonable equivalent interpretation. Value: - 65.534 dB ... +65.534 dB in steps of 0.002 dB (0 corresponds to 0 dB). 0xFFFF = -32768 = invalid.
Timing advance offset	uint16_t	Timing advance T_A measured for the UE in multiples of $16 * 64 * T_c / 2^{\mu}$ [3GPP TS 38.213 [4], Section 4.2] Value: 0 → 63 0xffff should be set if this field is invalid
Timing advance offset in nanoseconds	int16_t	Timing advance measured for the UE between the reference uplink time and the observed arrival time for the UE Value: - 16800 ... +16800 nanoseconds. 0xffff should be set if this field is invalid
RSSI	uint16_t	RSSI. See Table 3-23 for RSSI definition. If RSSI is reported in dBFS (see PARAM and CONFIG measurement TLVs) then this value represents -128dBFS to 0dBFS with a step size of 0.1dB If RSSI is reported in dBm (see PARAM and CONFIG measurement TLVs) then this value represents -128dBm to 0dB, with a step size of 0.1dB Value: 0-1280 0xffff should be set if this field is invalid
RSRP	uint16_t	RSRP. See Table 3-23 for RSRP definition. If RSRP is reported in dBFS (see PARAM and CONFIG measurement TLVs) then this value represents -128dBFS to 0dBFS with a step size of 0.1dB If RSRP is reported in dBm (see PARAM and CONFIG measurement TLVs) then this value represents -140dBm to -12dBm, with a step size of 0.1dB Value: 0-1280 0xffff should be set if this field is invalid
SR information	structure	Included if indicated by pduBitmap. See Table 3-133 for details



Field	Type	Description
HARQ information	structure	Included if indicated by pduBitmap. See Table 3-134 for details

Table 3-131 UCI PUCCH format 0 or 1 PDU

3.4.9.3 PUCCH PDU Format 2/3/4

The format of a PUCCH PDU Format 2/3/4 is shown in Table 3-132 and used for UCI received on the PUCCH with either Format 2, 3 or 4.

Field	Type	Description
pduBitmap	uint8_t	<p>Bitmap indicating presence of optional PDUs. Value: 0 = not present, 1 = present</p> <p>Interpretation depends on the setting of uciReportFormat in the request PUCCH PDU:</p> <p>separate reporting:</p> <ul style="list-style-type: none">- Bit 0: SR- Bit 1: HARQ- Bit 2: CSI Part 1- Bit 3: CSI Part 2 <p>combined reporting (i.e. transparent UCI payloads):</p> <ul style="list-style-type: none">- Bit 2: uciPayload (or uciPart1Payload, if uci is composed of two parts)- Bit 3: uciPart2Payload (if uci is composed of two parts)
Handle	uint32_t	The handle passed to the PHY in an <code>UL_TTI.request</code> PUCCH PDU.
RNTI	uint16_t	<p>The RNTI passed to the PHY in an <code>UL_TTI.request</code> PUCCH PDU.</p> <p>Value: 1 → 65535.</p>
PucchFormat	uint8_t	<p>PUCCH format</p> <p>Value: 0 -> 2 0: PUCCH Format2 1: PUCCH Format3 2: PUCCH Format4</p>
UISinrMetric	int16_t	<p>A metric of channel quality. Up to PHY implementation whether this is Signal-to-Thermal, Signal-to-Interference+Thermal or any other reasonable equivalent interpretation.</p> <p>Value: - 65.534 dB ... +65.534 dB in steps of 0.002 dB (0 corresponds to 0 dB). 0xFFFF = -32768 = invalid.</p>
Timing advance offset	uint16_t	<p>Timing advance T_A measured for the UE in multiples of $16 * 64 * T_c / 2^{\mu}$ [3GPP TS 38.213 [4], Section 4.2]</p> <p>Value: 0 → 63 0xffff should be set if this field is invalid</p>



Field	Type	Description
Timing advance offset in nanoseconds	int16_t	Timing advance measured for the UE between the reference uplink time and the observed arrival time for the UE Value: -16800 ... +16800 nanoseconds. 0xffff should be set if this field is invalid
RSSI	uint16_t	RSSI. See Table 3-23 for RSSI definition. If RSSI is reported in dBFS (see PARAM and CONFIG measurement TLVs) then this value represents -128dBFS to 0dBFS with a step size of 0.1dB If RSSI is reported in dBm (see PARAM and CONFIG measurement TLVs) then this value represents -128dBm to 0dB, with a step size of 0.1dB Value: 0-1280 0xffff should be set if this field is invalid
RSRP	uint16_t	RSRP. See Table 3-23 for RSRP definition. If RSRP is reported in dBFS (see PARAM and CONFIG measurement TLVs) then this value represents -128dBFS to 0dBFS with a step size of 0.1dB If RSRP is reported in dBm (see PARAM and CONFIG measurement TLVs) then this value represents -140dBm to -12dBm, with a step size of 0.1dB Value: 0-1280 0xffff should be set if this field is invalid
SR information	structure	Included if indicated by pduBitmap. See Table 3-135 for details
HARQ information	structure	Included if indicated by pduBitmap. See Table 3-136 for details
CSI part1 information	structure	Included if indicated by pduBitmap. See Table 3-137 for details
CSI part2 information	structure	Included if indicated by pduBitmap. See Table 3-138 for details
uciPayload (or Part1 Payload, if uci is composed of two parts)	structure	Included if indicated by pduBitmap. See Table 3-137 for details
uciPart2Payload (if uci Payload, if uci is composed of two parts)	structure	Included if indicated by pduBitmap. See Table 3-138 for details

Table 3-132 UCI PUCCH format 2, 3 or 4 PDU

3.4.9.4 SR, HARQ and CSI Part 1 / 2 PDUs

The UCI tables for the SR, HARQ, CSI Part 1 and CSI Part 2 are provide in this Section.

Field	Type	Description
SRindication	uint8_t	Indicates if an SR was detected.



Field	Type	Description
		<p>Value:</p> <p>0 = SR not detected</p> <p>1 = SR detected</p>
SRconfidenceLevel	uint8_t	<p>Confidence level of detected SR</p> <p>Indicates the likelihood that the decoded output for this PDU is correct.</p> <p>Value:</p> <p>0 = Good</p> <p>1 = Bad</p> <p>0xff should be set if this field is invalid</p>

Table 3–133 SR PDU for format 0 or 1

Field	Type	Description
NumHarq	uint8_t	<p>Number of HARQ bits present in UCI</p> <p>Value: 1->2</p>
HarqconfidenceLevel	uint8_t	<p>Confidence level of detected HARQ.</p> <p>Indicates the likelihood that the decoded output for this PDU is correct.</p> <p>Value:</p> <p>0 = Good</p> <p>1 = Bad</p> <p>0xff should be set if this field is invalid</p>
For NumHarq {		
	HarqValue	<p>Indicates result on HARQ data.</p> <p>Value:</p> <p>0 = NACK</p> <p>1 = ACK</p> <p>2 = DTX</p>
}		

Table 3–134 HARQ PDU for format 0 or 1

Field	Type	Description
SrBitLen	uint16_t	<p>Indicates the number of SR bits which are expected on the PUCCH transmission.</p> <p>Valid for PUCCH Format 2/3/4</p> <p>Value:</p> <p>1 = one SR bit;</p> <p>2 = two SR bits;</p> <p>3 = three SR bits;</p> <p>4 = four SR bits</p>



Field	Type	Description
		Note: When MAC is implemented according to 3GPP TS 38.215 [4][22] or [5], then maximum SR bits expected for a single UE is 4.
SrPayload	uint8_t [ceil(SrBitLen/8)]	Contents of SR

Table 3-135 SR PDU for format 2, 3 or 4

Field	Type	Description
Harq Detection Status	uint8_t	<p>Indicates CRC result on UCI containing this HARQ data.</p> <p>Value:</p> <ul style="list-style-type: none"> 1 = CRC Pass (used when CRC was attached) 2 = CRC Failure (used when UE is expected to attach CRC) 3 = DTX (undetected UCI) 4 = No DTX (indicates UCI detection) 5 = DTX not checked (indicates that PHY did not check for UCI DTX; in this case L2 may make use of other fields – e.g. UL SINR – to determine UCI validity)
Expected HarqBitLen	uint16_t	<p>Length of HARQ payload in bits</p> <p>Value: 1 -> 1706</p>
HarqPayload	uint8_t [ceil(HarqBitLen/8)]	<p>Contents of HARQ, excluding any CRC</p> <p>Actual size is the same Expected Size only if detection status is 1 (CRC Pass), 4 (No DTX) or 5 (DTX not checked), otherwise the size is zero.</p> <p>Bit mapping for this payload shall be MSBO. If PHY signals padding capability other than 'implementation', padding bits are appended to the payload.</p> <p>Notes:</p> <ul style="list-style-type: none"> - MSBO is interpreted as in 3GPP TS 38.212 [3], section 6.2.1: "lowest order information bit [position 0] is mapped to the most significant bit" - FAPIv4 is the first release that spells out bit mapping and padding for this field.

Table 3-136 HARQ PDU for format 2, 3 or 4 or for PUSCH

Field	Type	Description
This table describes both UCI part 1 and CSI part 1 transport, since they share data structure, yet are mutually exclusive (i.e. L1 either signals CSI part 1 or UCI part 1, not both at the same time)		
part1 Detection Status	uint8_t	Indicates detection outcome on UCI/CSI containing this UCI/CSI Part1 data.



Field	Type	Description
		<p>Value:</p> <p>1 = CRC Pass (used when CRC was attached) 2 = CRC Failure (used when UE is expected to attach CRC) 3 = DTX (undetected UCI/CSI) 4 = No DTX (indicates UCI/CSI detection) 5 = DTX not checked (indicates that PHY did not check for UCI/CSI DTX; in this case L2 may make use of other fields – e.g. UL SINR – to determine UCI/CSI validity)</p>
Expected part1BitLen	uint16_t	<p>Length of payload in bits</p> <p>Value: 1 -> 1706</p>
part1Payload	uint8_t [len]	<p>Contents of UCI/CSI, excluding any CRC. This will be raw data matching UCI/CSI payload built by UE.</p> <p>len = ceil(part1BitLen/8) or 0</p> <p>len is derived from Expected Size only if detection status is 1 (CRC Pass), 4 (No DTX) or 5 (DTX not checked), otherwise len is zero (and this part1Payload shall be absent).</p> <p>Bit mapping for this payload shall be MSB0. If PHY signals padding capability other than 'implementation', padding bits are appended to the payload.</p> <p>Note:</p> <ul style="list-style-type: none"> - MSB0 is interpreted as in 3GPP TS 38.212 [3], section 6.2.1: "lowest order information bit [position 0] is mapped to the most significant bit" - FAPIv4 is the first release that spells out bit mapping and padding for this field.

Table 3–137 UCI/CSI Inline Transport PDU

Field	Type	Description
		<p>This table describes both UCI part 2 and CSI part 2 transport, since they share data structure, yet are mutually exclusive (i.e. L1 either signals CSI part 2 or UCI part 2, not both at the same time)</p>
Part2 Detection Status	uint8_t	<p>Indicates CRC result on UCI/CSI Part2 data.</p> <p>Value:</p> <p>1 = CRC Pass (used when CRC was attached) 2 = CRC Failure (used when UE is expected to attach CRC) 3 = DTX (undetected UCI) 4 = No DTX (indicates UCI detection) 5 = DTX not checked (indicates that PHY did not check for UCI DTX; in this case L2 may make use of other fields – e.g. UL SINR – to determine UCI validity)</p>



Field	Type	Description
Expected Part2BitLen	uint16_t	<p>Length of UCI/CSI payload in bits</p> <p>Value: 1 -> 1706</p>
part2Payload	(see below)	<p>Signaling of part 2 contents, excluding any CRC will be according to the Tag and Value field, per the Tag and Value field descriptions. The actual payload length will be indicated by the <i>Expected Part2BitLen</i>. This will be raw data matching UCI/CSI payload built by UE.</p> <p>len = ceil(part2BitLen/8)</p> <p>len is derived from Expected Size only if detection status is 1 (CRC Pass), 4 (No DTX) or 5 (DTX not checked), otherwise the size is zero and this part2Payload shall be absent.</p> <p>Bit mapping for this payload shall be MSB0. If PHY signals padding capability other than 'implementation', padding bits are appended to the payload.</p> <p>This payload and its contents will be signaled as indicated by the Tag and Value fields below.</p> <p>Note:</p> <ul style="list-style-type: none">- MSB0 is interpreted as in 3GPP TS 38.212 [3], section 6.2.1: "lowest order information bit [position 0] is mapped to the most significant bit"- FAPIv4 is the first release that spells out bit mapping and padding for this field.
Tag	uint8	<p>Value: 0 -> 3</p> <ul style="list-style-type: none">• 0: payload is carried directly in the Value:• 1: 32-bit pointer to payload is in the value field• 2: 32-bit offset from first address• 3: offset from the end of control portion• 4: 32-bit pointer to payload is in the value field• 5: 32-bit offset from first address
Value	Variable of uint32_t	<p>32 bits, or a multiple of the number of bits.</p> <p>Tag=0: Only the most significant bytes of the size described 'len' are valid. Remaining bytes are Zero padded to the nearest bit 32-bit boundary</p> <p>Tag=1/4: Pointer to payload cumulating to length 'len'. Occupies 32-bits for tag 1 and 64 bits for tag 4.</p> <p>Tag=2/5: Offset from the "first address" to the payload of size 'len' is in the Value field. Where the first address is a predefined value. Occupies 32-bits for tag 2 and 64 bits for tag 5.</p> <p>Tag=3: Offset from the end of control portion to the payload of size 'len' is in the value field, where the end of control portion is indicated by Control Length field in</p>



Field	Type	Description
		UCI.indication. The payload occupies 32-bits

Table 3-138 UCI/CSI TLV transport PDU (for Part2)

3.4.10 SRS.indication

The format of the SRS.indication message is given in Table 3-139.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
Control Length	uint16_t	Length (in bytes) of control plane portion, including any padding bytes, if tag value 3 is used for TBS-TLV (i.e. CP/UP separation), may be zero otherwise. See Table 3-30 for the related capability. It is measured from the first byte of the message until then end of the control portion.
Number of PDUs	uint8_t	Number of PDUs included in this message. Value: 0-> 255
For each SRS PDU {		
Handle	uint32_t	The handle passed to the PHY in the the UL_TTI.request SRS PDU.
RNTI	uint16_t	The RNTI passed to the PHY in the UL_TTI.request SRS PDU. Value: 1 → 65535.
Timing Advance offset	uint16_t	Timing advance T_A measured for the UE in multiples of $16 * 64 * T_c / 2^u$ [3GPP TS 38.213 [4], Section 4.2] Value: 0 → 63 0xffff will be set if this field is invalid
Timing advance offset in nanoseconds	int16_t	Timing advance measured for the UE between the reference uplink time and the observed arrival time for the UE Value: - 16800 ... +16800 nanoseconds. 0xffff should be set if this field is invalid
SRS usage	uint8_t	0 – beamManagement 1 – codebook 2 – nonCodebook 3 – antennaSwitching 4 – 255: reserved Note: This field matches the SRS usage field of the SRS PDU for the Handle.
Report Type	uint8_t	The type of report included in or pointed to by Report TLV corresponds to the requested <i>Report Type</i> in the SRS PDU for the Handle:



Field	Type	Description
		<p>Values:</p> <ul style="list-style-type: none">• 0: Per-PRG and Symbol SNR (Table 3-141)• 1: Normalized Channel I/Q Matrix (Table 3-142)• 2: Channel SVD Representation (Table 3-143)• 3: Positioning (Table 3-144)• 255: No report (the length field of Report TLV will be set to 0). <p>Any values not listed above are reserved</p> <p>Note: The values above correspond to the bit positions of the Report Type field in the SRS PDU for the Handle.</p>
Report TLV		See Table 3-140

Table 3-139 SRS.indication message body

Field	Type	Description
Tag	uint16_t	<p>Value: 0, 3</p> <p>0: Report is carried directly in the value field</p> <p>1: 32-bit Pointer to payload is in the value field</p> <p>2: 32-bit offset from first address</p> <p>3: The offset from the end of the control portion of the message to the beginning of the report.</p> <p>4: 64-bit Pointer to payload is in the value field</p> <p>5: 64-bit offset from first address</p> <p>Other values are reserved.</p>
length	uint32_t	Length of the actual report in bytes, without the padding bytes
value	Variable or uint32_t	<p>32 bits, or a multiple of the number of bits indicated by the pdschMacPduBitsAlignment capability (default 32-bits).</p> <p>Tag=0: Only the most significant bytes of the size indicated by 'length' field are valid. Remaining bytes are zero padded to the nearest 32-bit bit boundary</p> <p>Tag=1/4: Pointer to the payload. Occupies 32-bits for tag 1 and 64 bits for tag 4.</p> <p>Tag=2/5: Offset from the "first address" to the payload is in the value field. Where the first address is a predefined value. Occupies 32-bits for tag 2 and 64 bits for tag 5.</p> <p>Tag=3 Offset from the end of the control portion of the message to the payload is in the value field. Occupies 32-bits.</p>

Table 3-140 SrsReport-TLV structure



The structure below represents the SRS report for beamforming-based SRS resources, as supported in FAPIv3. The only enhancement is the ability to report at PRG resolution, based on the MAC-signaled PRG size in the corresponding SRS PDU.

Field	Type	Description	
prgSize	uint16_t	Size in RBs of a precoding resource block group (PRG) – to which the same digital beamforming gets applied. Value: 1->275	
numSymbols	uint8_t	Number of symbols for SRS Value: 1 -> 4 If a PHY does not report for individual symbols, then this parameter should be set to 1.	
wideBandSNR	uint8_t	SNR value in dB measured within configured SRS bandwidth on each symbol. Value: 0 → 255 representing -64 dB to 63 dB with a step size 0.5 dB 0xff will be set if this field is invalid	
numReportedSymbols	uint8_t	Number of symbols reported in this message. This allows PHY to report individual symbols or aggregated symbols where this field will be set to 1. Value: 1->4	
For each reported symbol {			
	numPRGs	uint16_t	Number of PRGs to be reported for this SRS PDU. Value: 0-> 272
	For each PRG {		
	rbSNR	uint8_t	SNR value in dB. Value: 0 → 255 representing -64 dB to 63 dB with a step size 0.5 dB 0xff will be set if this field is invalid
	}		
}			

Table 3-141 FAPIv3 Beamforming report, with PRG-level resolution

The structure below is used to represent the complex channel $Nu \times Ng$ matrix H estimated between the UE's Nu antenna ports and gNBs Ng antenna ports. $H[uI, gI]$ represents the flat fading approximation of the channel between the uI -th antenna at the UE and the gI -th antenna at the gNB.



Field	Type	Description
Normalized iq Representation	uint8_t	0: 16-bit normalized complex number (iqSize = 2) 1: 32-bit normalized complex number (iqSize = 4) See section 3.4.10.1
numGnbAntennaElements	uint16_t	Ng: Number of gNB antenna elements Value: 0→511
numUeSrsPorts	uint16_t	Nu: Number of sampled UE SRS ports Value: 0→7
prgSize	uint16_t	Size in RBs of a precoding resource block group (PRG) – to which the same digital beamforming gets applied. Value: 1->272
numPRGs	uint16_t	Number of PRGs Np to be reported for this SRS PDU. Value: 0-> 272
Array representing channel matrix H	uint8_t [Np * Nu * Ng *iqSize]	Array of (numPRGs*Nu*Ng) entries of the type denoted by iqRepresentation $H\{PRG pI\} [ueAntenna uI, gNB antenna gI] = array[uI*Ng*Np + gI*Np + pI],$ - uI: 0...Nu-1 // UE antenna index - gI: 0...Ng-1 // gNB antenna index - pI: 0...Np-1 // PRG index

Table 3-142 Normalized Channel I/Q Matrix

The structure below is used to represent the SVD decomposition complex channel $Nu \times Ng$ matrix H estimated between the gNBs Ng antenna ports and UE's Nu antenna ports in the downlink direction, under reciprocity assumptions. $H[uI, gI]$ represents the flat fading approximation of the channel between the the gI -th antenna at the gNB and the uI -th antenna at the UE.

In particular, the table represents $H = U \Sigma V^H$, using the notation in section 2.4.4 of [Golub's Matrix computations] with:

- U : $Nu \times Nu$ = matrix of Nu orthonormal left eigenvectors.
- Σ : $Nu \times Ng$ = rectangular matrix of singular values. Since only at most Nu diagonal entries may be non-zero, the table represents Σ as the diagonal entries of the Σ_F matrix of size $Nu \times Nu$
- V^H : $Ng \times Ng$ = complex conjugate matrix of right eigenvectors. Only those right eigenvectors corresponding to entries in Σ_F are signaled in matrix V_F^H of size $Nu \times Ng$

Note: $H = U \Sigma V^H = U \Sigma_F V_F^H$



Field	Type	Description
Normalized iq Representation	uint8_t	0: 16-bit normalized complex number (iqSize = 2) 1: 32-bit normalized complex number (iqSize = 4) See section 3.4.10.1
Normalized singular value representation	uint8_t	0: 8-bit linear representation (sSize = 1) (0 ... 1] in steps of size 1/256 1: 16-bit linear representation (sSize = 2). (0 ... 1] in steps of size 1/65536 See section 3.4.10.2
Singular value scaling	int8_t	dB-domain representation of singular value scaling. Range: -10 ... 50 in 0.25 dB steps. = (val/4) + 54 Values outside this range are undefined. See section 3.4.10.2
numGnbAntennaElements	uint16_t	Ng: Number of gNB antenna elements Value: 0->511
numUeSrsPorts	uint8_t	Nu: Number of sampled UE SRS ports Value: 0->7
prgSize	uint16_t	Size in RBs of a precoding resource block group (PRG) – to which the same digital beamforming gets applied. Value: 1->272
numPRGs	uint16_t	Number of PRGs to be reported for this SRS PDU. Value: 0-> 272
For each PRG {		
Array representing matrix of left eigenvectors \mathbf{U}	uint8_t [Nu *Nu *iqSize]	Array of (Nu*Nu) entries of the type denoted by 'iq Representation' $U[ueAntenna uI, \text{left eigenvector leI}] = \text{array}[uI*Nu + leI],$ <ul style="list-style-type: none">- $uI: 0...Nu-1$- $leI: 0...Nu-1$
Array representing diagonal entries of the singular matrix Σ_F	uint8_t [Nu *iqSize]	Array of (Nu) entries of the type denoted by 'singular Values Representation' $\Sigma[\text{left eigenvector leI}, \text{right eigenvector rel}] =$ <ul style="list-style-type: none">• 0 if $leI \neq rel$• 0 if $\min(leI, rel) \geq Nu$• $\text{array}[leI]$ if $leI = rel$



	Array representing the complex conjugate of matrix of right eigenvectors V_F^H	uint8_t [Nu *Ng *iqSize]	Array of (Nu*Ng) entries of the type denoted by 'iq Representation' $V_F^H[\text{right eigenvector reI}, \text{gNB antenna gI}] = \text{array}[reI*Ng + gI],$ <ul style="list-style-type: none">- reI: 0...Nu-1- gI: 0...Ng-1
}			

Table 3-143 Channel SVD Representation

The structure below is used to represent the Positioning report.



Field	Type	Description
UL Relative Time of Arrival	uint32_t	T _{UL-RTOA} , as recorded in [3GPP TS 38.215 [19], sec 5.1] Value: 0 ... 1970049, per the mapping in [3GPP TS 38.133 [29], Table 13.1.1-1] (i.e. -985024 T _c ... 985024 T _c) 0xFFFFFFFF: not available
gNB Rx – Tx time difference	uint32_t	gNB Rx – Tx time difference, as recorded in [3GPP TS 38.215 [19], sec 5.1] Value: 0 ... 1970049, per the mapping in [3GPP TS 38.133, [29], Table 13.2.1-1] (i.e. -985024 T _c ... 985024 T _c) 0xFFFFFFFF: not available
Coordinate system for UL AoA	uint8_t	0: Local 1: Global
UL Angle of Arrival (UL AoA)	uint16_t	UL AoA, as recorded in [3GPP TS 38.215 [19], sec 5.1] Value: 0 ... 3599, , per the mapping in [3GPP TS 38.133 [29], Table 13.4.1-1] (i.e. -180 ... 180 degrees in step of 0.1 degrees) 0xFFFF: not available
UL SRS-RSRP	uint16_t	UL SRS-RSRP, as recorded in [3GPP TS 38.215 [19], sec 5.1] See Table 3-23 for RSRP definition. If RSRP is reported in dBFS (see PARAM and CONFIG measurement TLVs) then this value represents -144dBFS to 0dBFS with a step size of 0.1dB If RSRP is reported in dBm (see PARAM and CONFIG measurement TLVs) then this value represents -156dBm to -12dBm, with a step size of 0.1dB Value: 0-1440 0xffff should be set if this field is invalid

Table 3-144 SRS-based positioning report

3.4.10.1 I-Q Representation

Based on PHY capability signalled in [TBD Capability TLV], MAC may request SRS.indication reports represent I-Q samples in 16-bit or 32-bit representations.

16-bit representation



An array **a** of N 16-bit I-Q samples represents the kth normalized complex value entry (k = 0...N-1) in 8-bit array entries 2*k and 2*k+1 as follows:

- Normalized I-sample: (int8_t)a[2*k]/128.0
- Normalized Q-sample: (int8_t)a[2*k+1]/128.0

Note: array **a** has size N*2, and the k-th entry is illustrated below:

a[2k+1]	a[2k]
<i>Q-sample</i>	<i>I-sample</i>

32-bit representation

An array **a** of N 32-bit I-Q samples represents the kth normalized complex value entry (k = 0...N-1) in 8-bit array entries 4*k, 4*k+1, 4*k+2, 24*k+3 as follows:

- Normalized I-sample: (int16_t)(a[4*k + M] << 8 + a[4*k + 1-M]) / 32768.0
- Normalized Q-sample: (int16_t)(a[4*k + 2 + M] << 8 + a[4*k + 3-M]) / 32768.0

Where M = 1 if FAPI uses little-endian notation, otherwise M=0

Note: array **a** has size N*4, and the k-th entry is illustrated below:

a[4k+3]	a[4k+2]	a[4k+1]	a[4k]
<i>Q-sample</i>		<i>I-sample</i>	

3.4.10.2 Singular Value Representation

Singular values in the diagonal matrix Σ of the channel SVD decomposition are positive values. Post normalization, these values range between 0...1. Each actual singular value is obtained after accounting for normalized representation described in this section, and the singular value scaling (see Table 3-143):

$$\text{Singular value } V_{\text{linear}} = V_{\text{linear,normalized}} * 10^{(\text{singular value scaling})/20}$$

This section describes two normalized singular value representations supported in this specification:

8-bit dB representation

An array **a** of N 8-bit dB values represents the kth normalized entry (k = 0...N-1) in 8-bit array entry k as follows:

- $V_{\text{linear,normalized}} = (a[k] + 1)/256$

Notes:

- the range covers the following values: 1/256, 2/256, ... 255/256, 1
- array **a** has size N bytes and the k-th entry is illustrated below:



a[k]
V _{linear,normalized}

16-bit linear representation

An array **a** of N 16-bit $V_{\text{linear,normalized}}$ samples represents the k^{th} normalized complex value entry ($k = 0 \dots N-1$) in 8-bit array entries $2*k$, $2*k+1$ as follows:

- $V_{\text{linear,normalized}} = (\text{uint16_t})(a[2*k + M] << 8 + a[2*k + 1-M] + 1) / 65536.0$

Where $M = 1$ if FAPI uses little-endian notation, otherwise $M=0$

Notes:

- the range covers the following values: $1/65536, 2/65536, \dots 65535/65536, 1$
- the uint16_t 16-bit representation follows the same byte endianness as FAPI
- array **a** has size $N*2$, and the k -th entry is illustrated below:

a[2k+1]	a[2k]
$V_{\text{linear,normalized}}$	

3.4.11 RACH.indication

The format of the RACH.indication message is given in Table 3-145. There can be more than one RACH.indication message per slot.

Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 -> 159
Number of PDUs	uint8_t	Number of Measurement PDUs included in this message. Value: 0-> 255
For each Measurement:		
handle	uint32_t	Handle for the <code>UL_TTI.request</code> RACH PDU to which this report is linked
SymbolIndex	uint8_t	The index of first symbol of the PRACH TD occasion. [3GPP TS 38.211 [2], sec 6.3.3.2 and Tables 6.3.3.2-2 to 6.3.3.2-4] Value: 0->13
SlotIndex	uint8_t	The index of first slot of the PRACH TD occasion in a system frame Value: 0->79



Field		Type	Description
	raIndex	uint8_t	The index of the received PRACH frequency domain occasion. [3GPP TS 38.211 [2], sec 6.3.3.2] Value: 0->7
	avgRssi	uint16_t	Average value of RSSI in dB Value: 0->170000 representing -140dBm to 30dBm with a step size of 0.001dB 0xffffffff should be set if this field is invalid.
	avgSnr	uint8_t	Average value of SNR in dB Value: 0 → 254 representing -64 dB to 63 dB with a step size 0.5 dB. 0xff should be set if this field is invalid.
	numPreambles	uint8_t	Number of detected preambles in the PRACH occasion. Value: 0->64
For each preamble			
	preambleIndex	uint8_t	Preamble Index. Value: 0-> 63
	Timing advance offset	uint16_t	Timing advance for PRACH. [3GPP TS 38.213 [4], sec 4.2] Value:0 -> 3846 in units of in multiples of $16 * 64 * T_c / 2^u$ 0xffff should be set if this field is invalid.
	Timing advance offset in nanoseconds	uint32_t	Timing advance measured for the UE between the observed arrival and the reference uplink time for the UE Value: 0 ... 2005000 nanoseconds. 0xffffffff should be set if this field is invalid
	preamblePwr	uint32_t	Preamble Received power in dBm or dBFS, depending on the P5 capability (preamblePowerUnit) and configuration (preamblePowerUnitSelection) Value: 0->170000 representing Reporting range in dBm: -140dBm to 30dBm with a step size of 0.001dB Reporting range in dBFS: -170 dBFS to 0 dBFS with a step size of 0.001dB 0xffffffff should be set if this field is invalid Notes: <ul style="list-style-type: none">- step size should not be used as an expected accuracy guideline.- implementations are not expected to report preamblePwr > 15 dBm
	preambleSnr	uint8_t	Preamble SNR in dB



Field	Type	Description
		Value: 0 → 254 representing -64 dB to 63 dB with a step size 0.5 dB. 0xff should be set if this field is invalid.
	preamblePwrRSRP	uint16_t RSRP. See Table 3-23 for RSRP definition. If RSRP is reported in dBFS (see PARAM and CONFIG measurement TLVs) then this value represents -128dBFS to 0dBFS with a step size of 0.1dB If RSRP is reported in dBm (see PARAM and CONFIG measurement TLVs) then this value represents -140dBm to -12dBm, with a step size of 0.1dB Value: 0-1280 0xffff should be set if this field is invalid

Table 3-145 RACH.indication message body

3.4.12 DL Node Sync

The DL Node Sync message is supported when using the nFAPI message format for a PHY that supports Delay Management. The body of the DL Node Sync message is as documented in section 4.1.1 of SCF-225 [10].

3.4.13 UL Node Sync

The UL Node Sync message is supported when using the nFAPI message format for a PHY that supports Delay Management. The body of the UL Node Sync message is as documented in section 4.1.2 of SCF-225 [10].

3.4.14 Timing Info

The Timing Info message is supported when using the nFAPI message format for a PHY that supports Delay Management. The body of the Timing Info message is as documented in section 4.1.3 of SCF-225 [10].



References

- [1] 3GPP TS 38.401 NG-RAN Architecture Description v16.9.0
- [2] 3GPP TS 38.211 NR Physical Channel and Modulation v15.10.0
- [3] 3GPP TS 38.212 NR Multiplexing and Channel Coding v15.13.0
- [4] 3GPP TS 38.213 NR Physical Layer Procedures for Control v15.14.0
- [5] 3GPP TS 38.214 NR Physical Layer Procedures for Data v15.16.0
- [6] 3GPP TS 38.331 NR Radio Resource Control (RRC) Protocol Specification v15.15.0
- [7] 3GPP TS 38.104 NR Base-station (BS) Radio Transmission and Reception v15.16.0
- [8] 3GPP TS 38.300 NR Overall Description Stage-2 v15.13.0
- [9] SCF 223 5G FAPI P19 RF and Digital Front End Control API v4.0
- [10] SCF 225 5G nFAPI Specification v3
- [11] 3GPP TS 38.300 NR Overall Description Stage-2 v16.8.0
- [12] 3GPP TS 38.211 NR Physical Channel and Modulation v16.9.0
- [13] 3GPP TS 38.331 NR Radio Resource Control (RRC) Protocol Specification v16.7.0
- [14] 3GPP TS 29.274 3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3 v15.9.0
- [15] 3GPP TS 38.212 NR Multiplexing and Channel Coding v16.9.0
- [16] 3GPP TS 38.214 NR Physical Layer Procedures for Data v16.9.0
- [17] 3GPP TS 38.104 NR Base-station (BS) Radio Transmission and Reception v16.11.0
- [18] 3GPP TS 38.215 NR; Physical layer measurements v15.7.0
- [19] 3GPP TS 38.215 NR; Physical layer measurements v16.5.0
- [20] 3GPP TS 38.321 NR; Medium Access Control (MAC) protocol specification v15.12.0
- [21] 3GPP TS 38.321 NR; Medium Access Control (MAC) protocol specification v16.7.0
- [22] 3GPP TS 38.213 NR Physical Layer Procedures for Control v16.9.0
- [23] Samsung, "Summary for WI: Enhancement on MIMO for NR", [RP-202803](#), 3GPP TSG RAN Meeting #90-e, December 7-11, 2020
- [24] Moderator(ZTE), "FL summary on the maintenance of 2-step RACH channel structure", [R1-2002728](#), 3GPP TSG-RAN WG1 Meeting #100bis-e, April 20th – 30th, 2020
- [25] Moderator(ZTE), "FL summary on the maintenance of 2-step RACH procedures", [R1-2002729](#), 3GPP TSG-RAN WG1 Meeting #100bis-e, April 20th – 30th, 2020
- [26] Sasha Sirotkin, *5G Radio Access Network Architecture: The Dark Side of 5G*, Wiley-IEEE Press, 2020
- [27] Intel Corporation, Ericsson, "New WID: NR Positioning Support" , [RP-190752](#), 3GPP TSG RAN Meeting #83, March 18-21, 2019
- [28] 3GPP TS 37.355, LTE Positioning Protocol (LPP) v16.7.0
- [29] 3GPP TS 38.133, NR; Requirements for support of radio resource management v16.11.0