LTE eNB L1 API Definition

Document Number: PC-101786

Version Number: 1.24

Product Reference: PC860x

Picochip CONFIDENTIAL

Picochip Inc, UK Headquarters: Upper Borough Court,Upper Borough Walls, Bath BA1 1RG, United Kingdom
www.picochip.com

Version History

Version Number	Approved by	Author(s)	Reviewed by	Version Date	Summary & Comments
Previous versions deleted for clarity.					
1.09		Adrian Mansell		14th August 2009	Many detail changes and editorial updates. V1.09 itself has an additional table detailing the changes.
1.10		Adrian Mansell		15th Sep 2009	Added SR.indication. CFI and DCI power offset have been moved to TLVs and other TLVs renumbered. DCI DL PDU has scrambling type field added. BCH, MCH & PCH PDU transmission power redefined. Clarification to UL subframe configuration changes to Rx.indication.
1.11		Adrian Mansell		11th March 2010	Many detail clarifications and corrections. Major changes are: reformatting of TLV tables; addition of SRS TLV; addition of shortening flags for PUSCH and PUCCH; clarification of PUCCH resources for HARQ and addition of FDD information; user indexing for SRS; addition of bundle scrambling sequence to HARQ.ind and UL measurements to CRC.ind; addition of appendices for timing estimation, Doppler estimation, message timing and measurement definitions.
1.12		Adrian Mansell		16th April 2010	Corrections and clarifications following review of v1.11. Major changes are: addition of new TLVs and clarification of applicability of all TLVs; revert ULSCH NDI to a toggle; correction of PDU sizes in 3.3.1.2.2.2 *et seq*; correction of sequencing and timing of SR.indication.
1.13		Adrian Mansell			Section 2.1.1, added reference to description of Mode A and Mode B subframe indication. Introduction clarifies that not all functionality may be supported by L1 product releases. Various minor typographical and editorial corrections. Section 3.3.1.2.1.1 clarified number of DL HARQ processes. Corrected descriptions of TLVs 85 and 86. TLV for Format 4 PRACH peak ratio has been removed from FDD configuration. Clarified timing units in RACH.ind. Added TLV53 to control RACH and SR indications; updated RACH and SR procedures to suit. Clarified that MU-MIMO operation requires that both UEs use the same resource block allocation.
1.14		Adrian Mansell		6th Jan 2011	Clarified the mapping of cyclic shift field in ULSCH PDU to a value of $n_{DMRS}^{(2)}$. Clarified resource allocation values for SPS release in section **3.3.1.3.2**. Clarified the TPC field encoding for format 3 and format 3A DCIs. Removed the transport block to codeword swap flag from DLSCH PDU. Corrected structure of PCH PDU to match the L1 implementation. Changed K=6 to K=3 for Timing Advance calculation in Appendix A. Removed appendix B relating to SRS IIR filtering; this function is now implemented in L1. Clarified the PUCCH type in UCI_xxx PDUs.
1.15		Adrian		8th Feb	Corrected SINR fields in CRC.indication.

		Mansell		2011	Clarification of applicability of TLV 86.
1.16		Adrian Mansell		8th Feb 2011	Added fields for PUCCH RSSI measurements in CQI PDU and HARQ indications.
1.17		Adrian Mansell		15th Feb 2011	Correction to byte alignment in CQI PDU
1.18		Adrian Mansell		25th Feb 2011	Added PUCCH RSSI to SR indication.
1.19		Adrian Mansell		28th Mar 2011	Clarification of HARQ size field in TDD Special Bundling. Corrected byte alignment at the end of PCH PDU.
1.20		Adrian Mansell		28th Mar 2011	Added TLV for L1 version reporting. Updated L1 message timings to reflect the PC960x external turbo decoder timings.
1.21		Adrian Mansell		8th April 2011	Ported to latest template. Removed legacy version history. Minor editorial corrections. Corrected description of TLV 17. Updated default value TLVs 82 & 83 (PRACH Peak Ratios). Clarified fields in ULSCH_CQI_RI PDU. Clarified UL Index field in DCI UL PDU. Clarifications of Downlink assignment index in DCI PDUs.
1.22		Adrian Mansell		6th May 2011	Reviewed with respect to release 9. Updated document references to release 9 versions. Fixed a number of incorrect references to external documents. Clarified the use of HARQ resource fields in UCI_HARQ when the HARQ resource is not active. Significant update to Error.indication description, error codes and dependent bytes. Removal of sub-error codes. Removal of individual sections relating error codes to specific messages. Removal of empty appendix on measurements.
1.23		Adrian Mansell		23rd May 2011	Additional change to format 4 PRACH peak ratio TLV. Clarification of DL Assignment Index field in DCI. Clarified" number of resource blocks" field for format 0 DCI. Corrected length of TLV 46.
1.24		Adrian Mansell		6th June 2011	Updated L1 API timing window for subframe request and TX request. Updated one reference in DLSCH PDU description. Clarification of SRS present field in UL config.

Related Documents

Ref.	Title	Number	Version & Date
[1]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control	TS36.331	9.3.0
[2]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control(MAC) protocol specification	TS36.321	9.3.0
[3]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Overall Description; Stage 2	TS36.300	9.4.0
[4]	3GPP Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Architecture description	TS36.401	9.2.0
[5]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode	TS36.304	9.3.0
[6]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer procedures	TS36.213	9.2.0
[7]	3GPP Base Station(BS) Radio Transmission and Reception	TS36.104	9.4.0
[8]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channel and Modulation	TS36.211	9.1.0
[9]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding	TS36.212	9.2.0
[10]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA);User Equipment(UE) radio access capabilities	TS36.306	9.2.0

Terms and Acronyms

Term / Acronym	*Definition*
3GPP	3^{rd} Generation Partnership Project
ACK	Acknowledge
API	Application Program Interface
BCH	Broadcast Channel
CC	Convolutional Code
CCE	Control Channel Element
CDD	Cyclic Delay Diversity
CFI	Control Format Indicator
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
CS	Cyclic Shift
CTC	Convolutional Turbo Codes
DCI	Downlink Control Information
DL	Downlink
DL-SCH	Downlink Shared Channel
DMA	Direct Memory Access
DMRS	Demodulation Reference Symbol
DwPTS	Downlink Pilot Time Slot
EARFCN	Evolved Absolute Radio Frequency Channel Number
eNB	evolved Node B
EPC	Evolved Packet Core
EPRE	Energy Per Resource Element
EUTRA	Evolved Universal Terrestrial Radio Access
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
FDD	Frequency Division Duplex
FEC	Forward Error Correction
FFT	Fast Fourier Transform
GP	Guard Period
HARQ	Hybrid Automatic Repeat Request
HCS	Header Check Sequence
HI	HARQ Indicator
IR	Incremental Redundancy
LTE	Long Term Evolution
MAC	Medium Access Control Layer
MBSFN	Multimedia Broadcast multicast services Single Frequency Network
MCH	Multicast Channel

MCS	Modulation and Coding Scheme
MIB	Master Information Block
MIMO	Multiple Input Multiple Output
NA	Not Applicable
NACK	Negative Acknowledge
OFDMA	Orthogonal Frequency Division Multiple Access
OS	Operating System
PCH	Paging Channel
PDCCH	Physical Downlink Control Channel
PDCP	Packet Data Convergence Protocol
PDSCH	Physical Downlink Shared Channel
PDU	Protocol Data Unit
PHICH	Physical Hybrid ARQ Indicator Channel
PHY	Physical Layer
PRACH	Physical Random Access Channel
PUCCH	Physical Uplink Control Channel
PUSCH	Physical Uplink Shared Channel
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
RA	Random Access
RACH	Random Access Channel
RB	Resource Block
RI	Rank Indicator
RRC	Radio Resource Control
RNTI	Radio Network Temporary Identifier
RSSI	Receive Signal Strength Indicator
RX	Receive
S1	The interface between the E-UTRAN and EPC
SAP	Service Access Point
SF	Subframe
SFN	System Frame Number
SISO	Single Input Single Output
SR	Scheduling Request
SRS	Sounding Reference Symbol
STC	Space Time Coding
STTD	Space Time Transmit Diversity
TB	Transport Block
TDD	Time Division Duplex
TLV	Tag Length Value

Tx	Transmit
UCI	Uplink Control Information
UE	User Equipment
UL	Uplink
UL-SCH	Uplink Shared Channel
UpPTS	Uplink Pilot Time Slot
WCDMA	Wideband Code Division Multiple Access
X2	Interface between two eNBs
ZT CC	Zero Tailed Convolutional Coding

Table of Contents

1 Introduction

This document describes the L1 API which provides the interface to the LTE PHY running on a picoArray™. The L2/L3 protocols expected to use this API are the RRC layer [1], MAC layer [2], and the scheduler. It is assumed that these run on a general purpose microprocessor or network processor.

The LTE standard [3] has been designed to support both TDD and FDD deployments. The LTE L1 API, described in this document, also supports TDD and FDD modes. Features which are specific to only TDD, or FDD, are clearly highlighted.

This document is divided into four sections. The first section provides a description of typical procedures which will occur between the L1 and L2/L3 software. The second section provides the definition of the L1 API messages. The third section, defines the encoding used in the API messages. Finally, the fourth section gives a brief overview of transport mechanisms between the L1 and L2/L3 software.

The API described in this document facilitates a broad range of LTE functionality. The user should consult the relevant L1 release note to determine the actual functionality supported by the product.

1.1 LTE

LTE is standardized by 3GPP (http://www.3gpp.org) and designed as an evolution to the current WCDMA wireless network, which is in widespread use today. A critical requirement of LTE is the capability of supporting high data rates (300Mbps), and many aspects of the LTE network have been designed specifically to support high data rates and low latency.

Figure 1 shows the architecture of an LTE network. It consists of only two elements; the Evolved Pack Core (EPC) and the E-UTRAN Node B (eNB). The LTE L1 API resides within the eNB element. The two standardized interfaces in an LTE network are called S1 and X2. The L1 is not involved in either of these interfaces, and both are out of scope for this document.

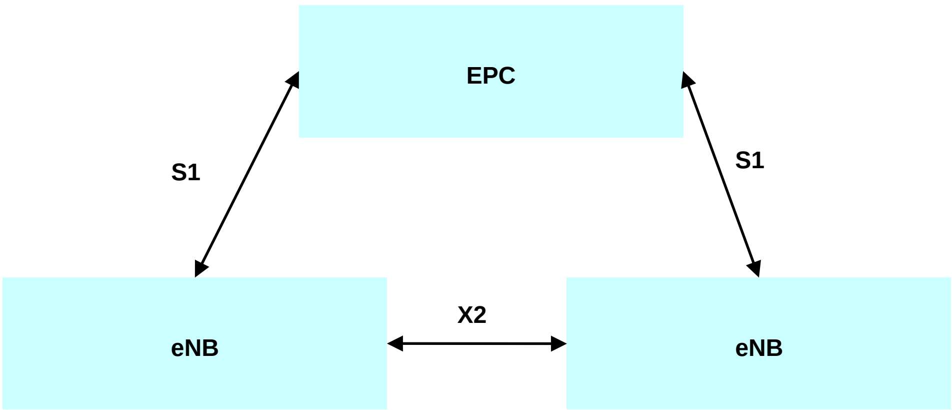

Figure 1: LTE Architecture

1.2 L1 API

The L1 API, defined in this document, resides within the eNB component. The functionality of an eNB is shown in Figure 2 and Figure 3. In both Figures the location of the L1 API is highlighted.

Figure 2 shows the protocol model for the eNB defined in the E-UTRAN architectural standard [4]. It highlights the separation of control- and data-plane information, which is maintained throughout the LTE network. Both control- and data-plane information is passed through the L1 API, however, each API message contains either control- or data-plane information, but never both.

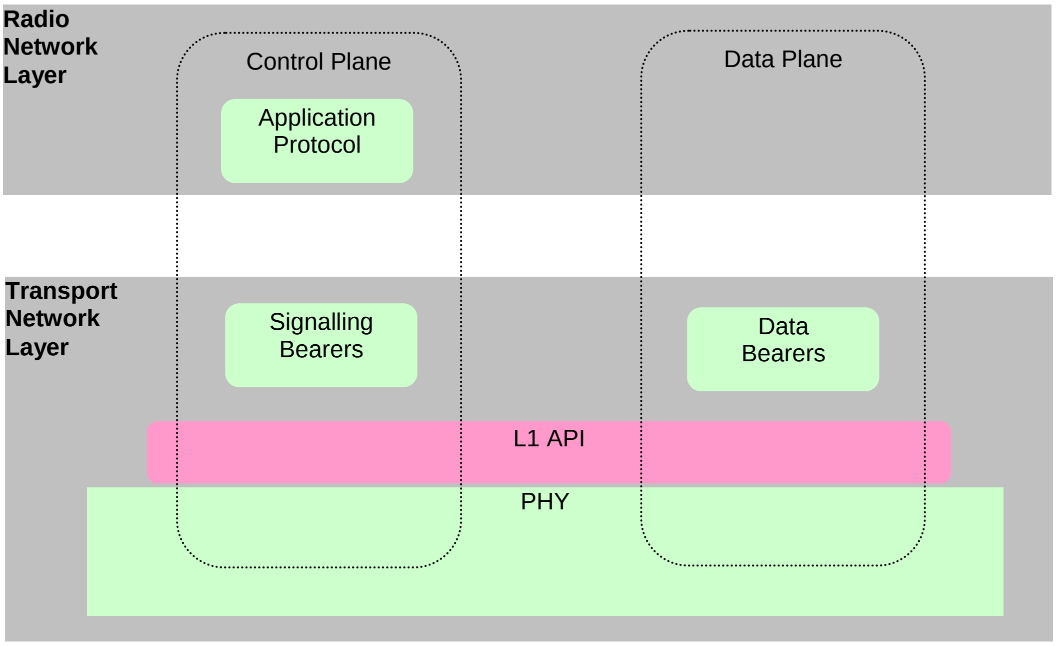

Figure 2: E-UTRAN protocol model

Figure 3 provides an example of how the different L2/L3 protocol layers will interact with the L1 API. In this example, it is assumed that the RRC layer is responsible for configuration procedures. The MAC layer is responsible for the exchange of data-plane messages with the PHY. Finally, the scheduler is responsible for deciding the subframe structure, and receiving measurement information from the PHY. It is important to note that there is no requirement for the eNB protocol to follow this example. The L1 API is a collection of messages which can be routed anywhere by the eNB software.

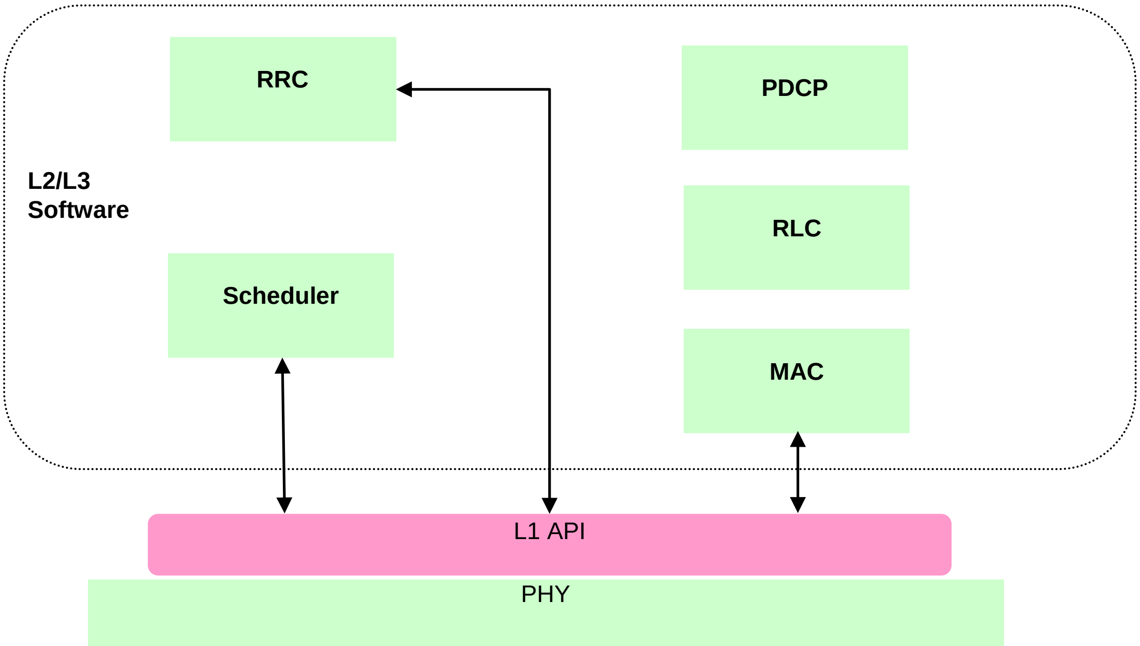

Figure 3: L1 API Interactions

2 L1 API Procedures

This section gives an overview of the procedures which use the L1 API. These procedures are split into two groups, namely, configuration procedures and subframe procedures. Configuration procedures handle the management of the PHY layer and are expected to occur infrequently. Subframe procedures determine the structure of each 1ms subframe and operate with a 1ms periodicity.

2.1 Configuration Procedures

The configuration procedures supported by the L1 API are:

- Initialization
- Termination
- Restart
- Reset
- Reconfigure
- Query
- Notification

These procedures will move the PHY layer through the IDLE, CONFIGURED and RUNNING states, as shown in Figure 4. A list of the L1 API `.request` messages which are valid in each state is given in Table 1.

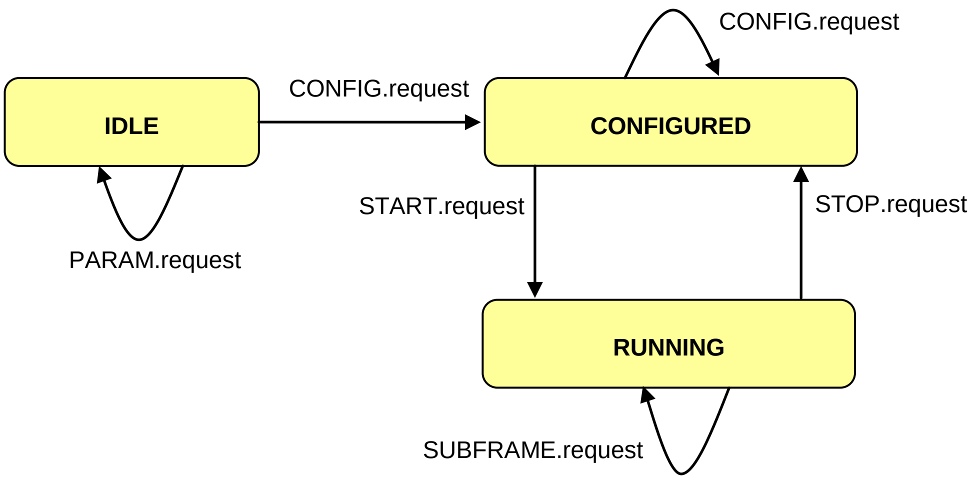

Figure 4: PHY layer state transactions on L1 API configuration messages

Idle State	*Configured State*	*Running State*
PARAM.request	PARAM.request	CONFIG.request
CONFIG.request	CONFIG.request	STOP.request
	START.request	SUBFRAME.request
		TX.request
		HI_DCI.request

Table 1: L1 API request message valid in each 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 5, the different stages are:

- L2/L3 software must load and start the PHY software on the picoArray™
- The PARAM message exchange procedure
- The CONFIG message exchange procedure
- The START message exchange procedure

The initialization procedure is completed when the PHY sends the L2/L3 software a `SUBFRAME indication`. See section 2.2.1 for a description of the `SUBFRAME indications` available.

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

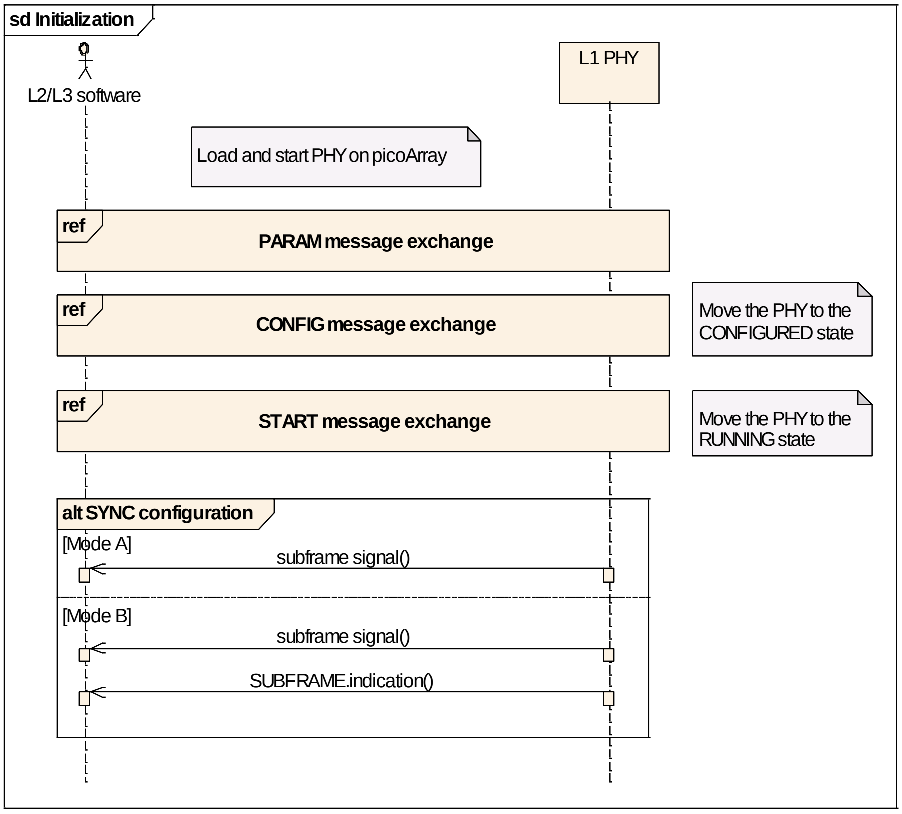

Figure 5: Initialization procedure

The PARAM message exchange procedure is shown in Figure 6. 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. 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: Information returned by the PHY during a PARAM message exchange

From Figure 6 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 loaded and running on the picoArray™ 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. 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 it should be reloaded and restarted.

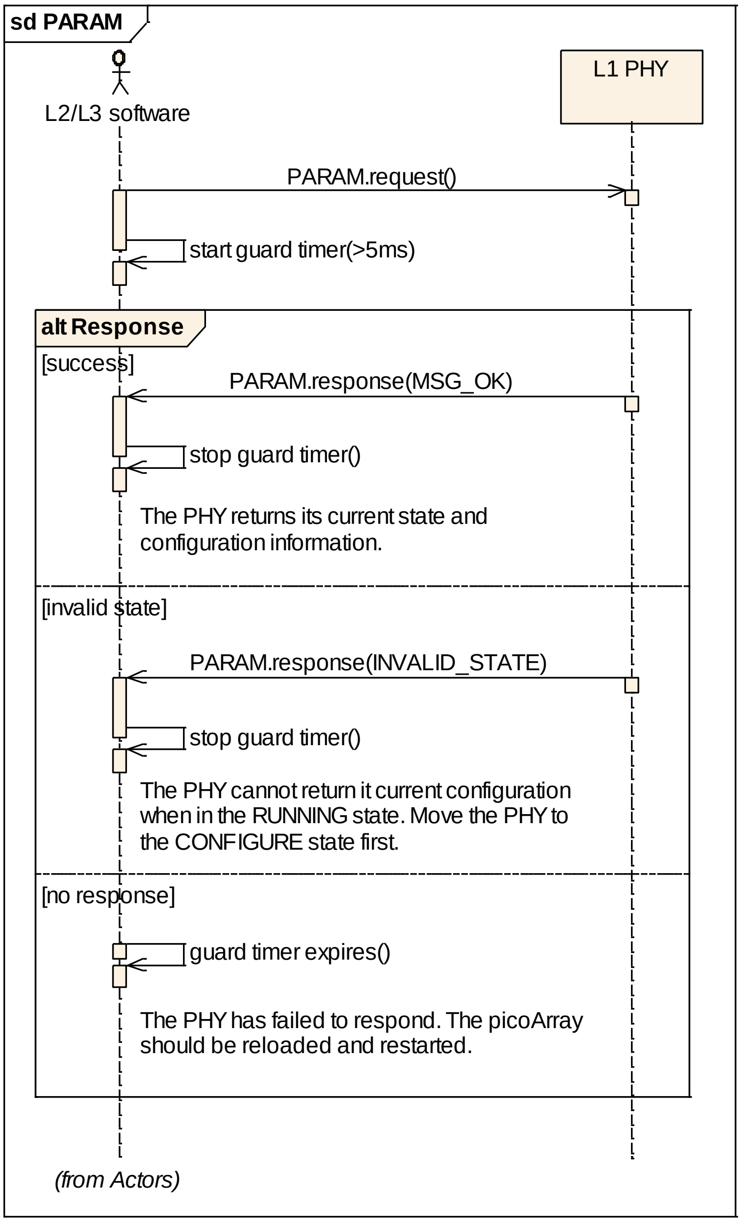

Figure 6: PARAM message exchange

The CONFIG message exchange procedure is shown in Figure 7. 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 highlighted later in Section 3.2.2.1. 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 highlighted later in Section 3.2.2.1. 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.

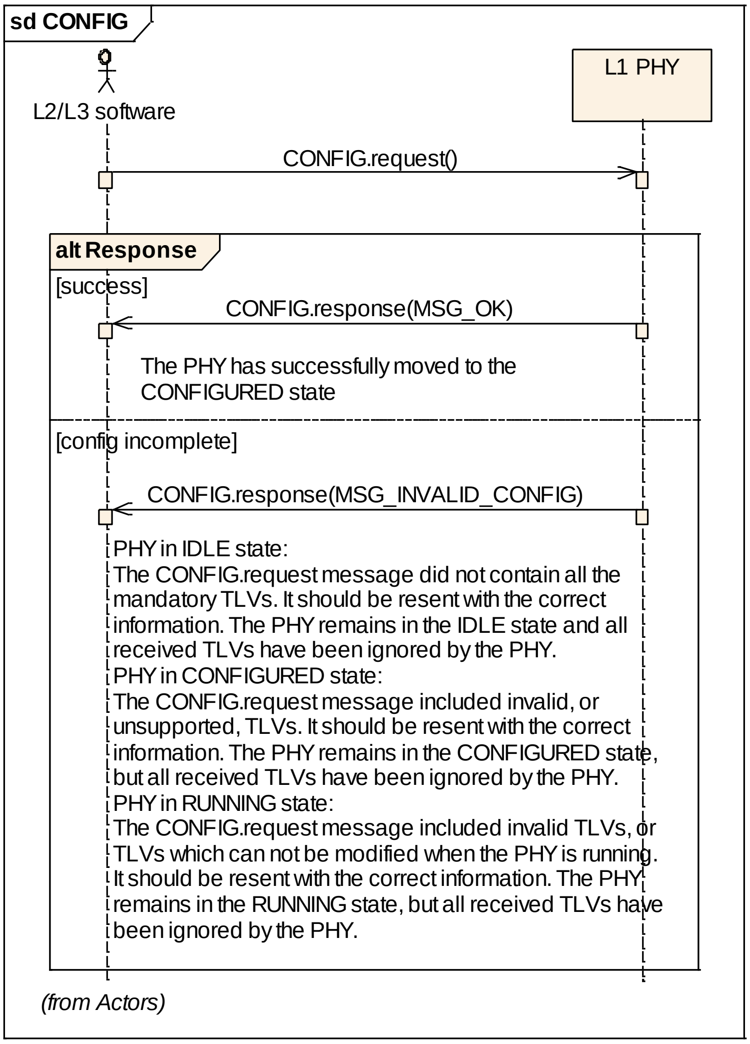

Figure 7: CONFIG message exchange

The START message exchange procedure is shown in Figure 8. Its purpose is to instruct a configured PHY to start transmitting as an eNB. The L2/L3 software initiates this procedure by sending a `START.request` message to the PHY. If the PHY is in the CONFIGURED state, it starts an internal 1ms timer and waits for an internal command to issue a SUBFRAME indication. After the PHY has sent its first SUBFRAME indication 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.

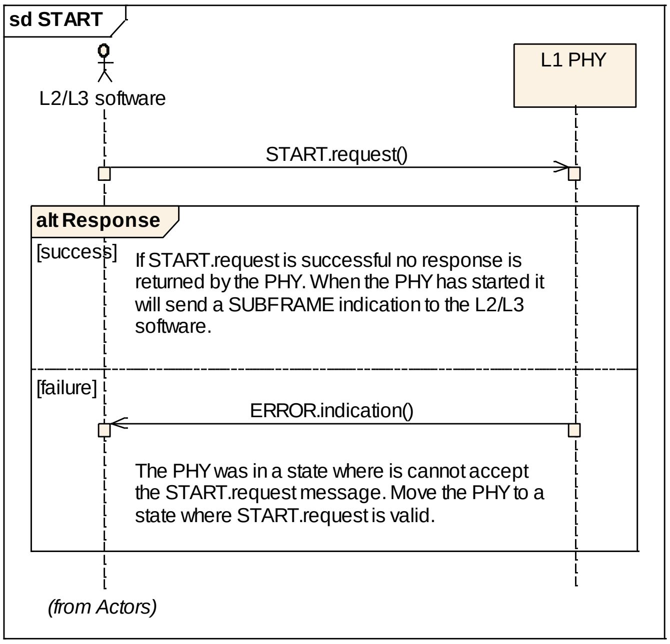

Figure 8: START message exchange

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 an eNB. The termination procedure is shown in Figure 9 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.

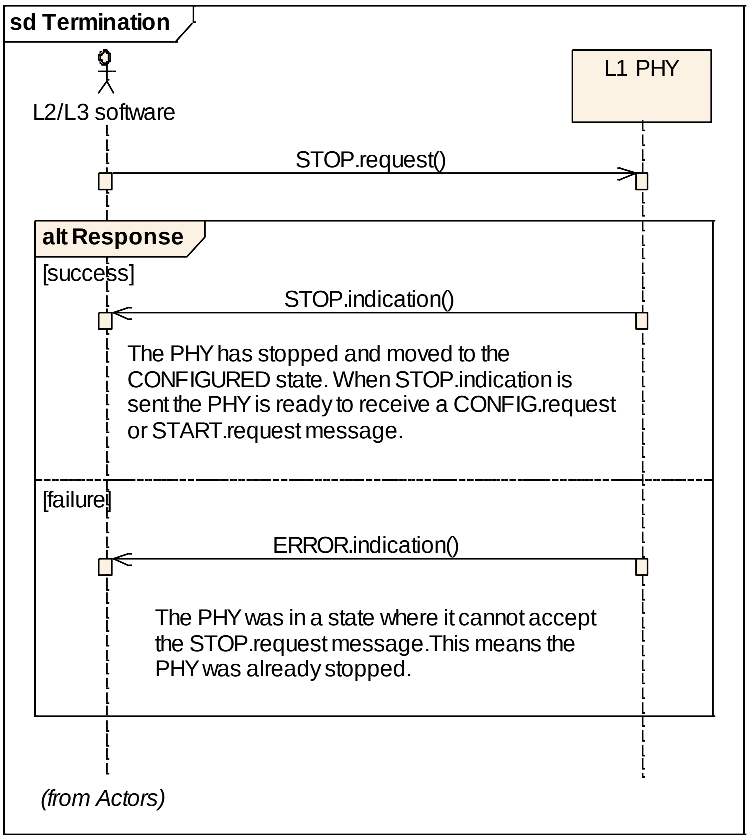

Figure 9: Termination procedure

2.1.3 Restart

The restart procedure is shown in Figure 10. 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 9. This moves the PHY to the CONFIGURED state. To restart transmission it should follow the START message exchange, shown in Figure 8, moving the PHY back to the RUNNING state.

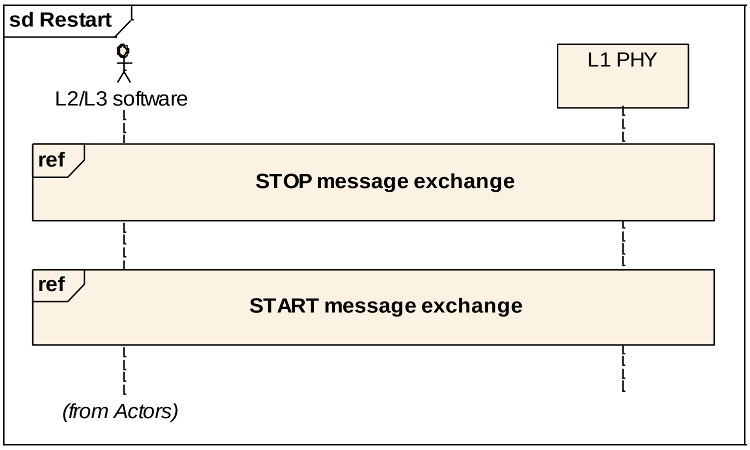

Figure 10: Restart procedure

2.1.4 Reset

The reset procedure is shown in Figure 11. This procedure is used when the L2/L3 software wants to return the PHY to the IDLE state. This can only be achieved by terminating the PHY (as shown in Figure 9) and then reloading and restarting the PHY on the picoArray™.

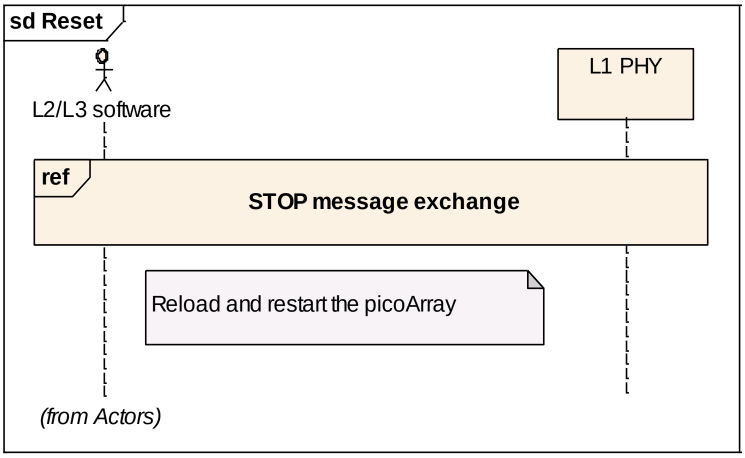

Figure 11: Reset procedure

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 12. 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 9, is followed to halt the PHY and move it to the CONFIGURED state. The CONFIG message exchange, shown in Figure 7, is used to reconfigure the PHY. Finally, the START message exchange, shown in Figure 8, is followed to start the PHY and return it to the RUNNING state.

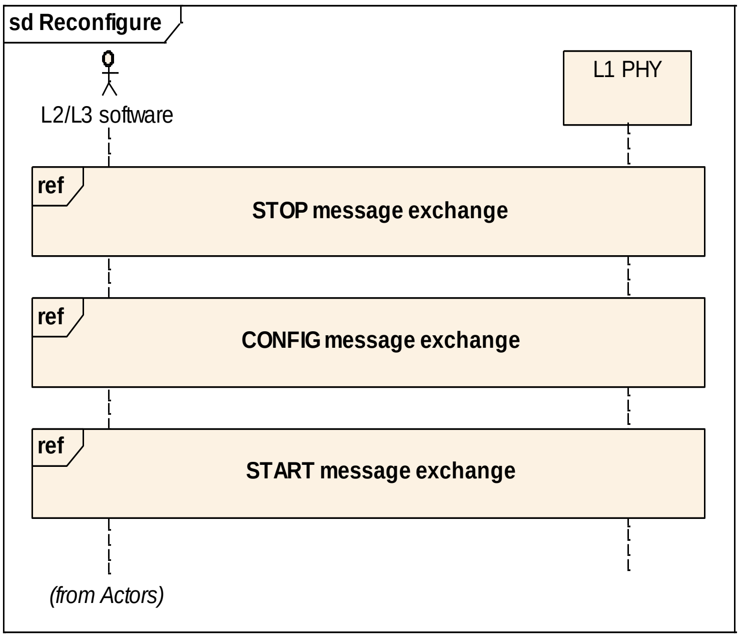

Figure 12: Major reconfigure procedure

The minor reconfiguration procedure is shown in Figure 13. It is typically used in conjunction with a RRC system information update.

In the subframe 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 highlighted later in Section 3.2.2.1. TLVs included in the `CONFIG.request` message for subframe N will be applied immediately, and used when processing the subsequent message relating to subframe N, for example, `SUBFRAME.request`. Reconfiguring the PHY while in the RUNNING state has a further restriction, the `CONFIG.request` message must be sent before the `SUBFRAME.request` message.

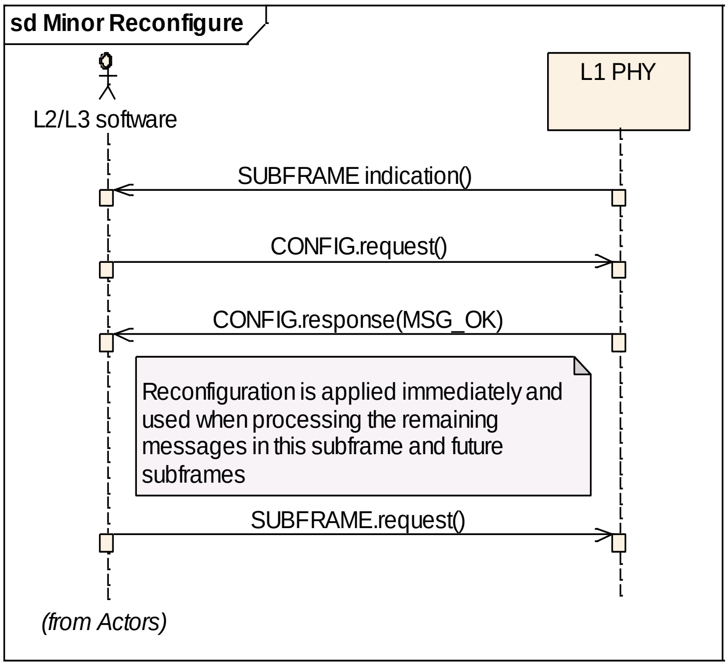

Figure 13: Minor reconfigure procedure

2.1.6 Query

The query procedure is shown in Figure 14. 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 6, is used. This signalling sequence can be followed when the PHY is stopped, in either the IDLE or CONFIGURED state.

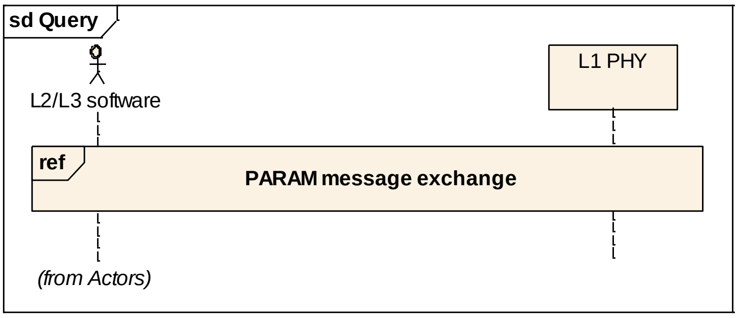

Figure 14: Query procedure

2.1.7 Notification

The notification procedure is shown in Figure 15. 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.

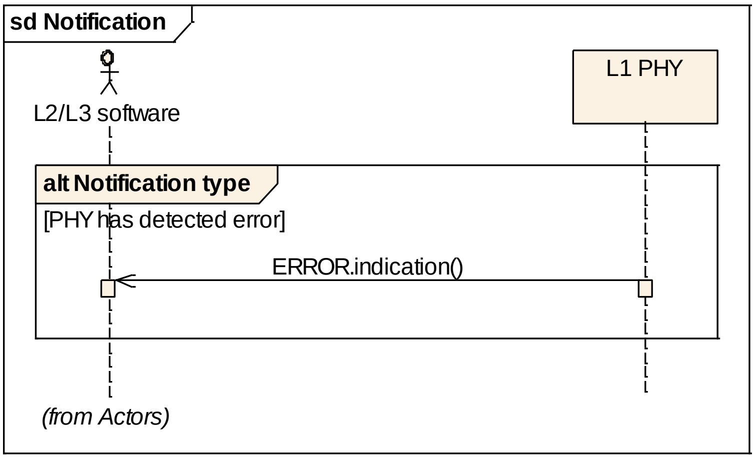

Figure 15: Notification procedures

2.2 Subframe Procedures

The subframe procedures have two purposes. Firstly, they are used to control the DL and UL frame structures. Secondly, they are used to transfer the subframe data between the L2/L3 software and PHY. The subframe procedures supported by the L1 API are:

- Transmission of a 1ms SUBFRAME signal
- Synchronization of SFN/SF 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 and reception of ACK/NACK response
- Transmission of the MCH transport channel
- Reception of the RACH transport channel
- Reception of the UL-SCH transport channel and transmission of ACK/NACK response
- Reception of the sounding reference signal
- Reception of CQI and RI reporting
- Reception of scheduling request information

2.2.1 SUBFRAME Signal

A SUBFRAME signal is sent from the PHY, to the L2/L3 software, indicating the start of a 1ms frame. The type of SUBFRAME signal is determined by the PHY Sync Mode TLV, sent in the `CONFIG.request` message. If Mode A was configured, the SUBFRAME indication is given by an interrupt mechanism sent from the picoArray^{TM} to the host processor. However, if Mode B was configured, the indication is given by both a L1 API `SUBFRAME.indication` message and the interrupt mechanism.

The location of the SUBFRAME signal for TDD (frame structure 2) is shown in Figure 16 and Figure 17. In TDD two frame structures are possible, one with 5ms switch points and one with 10ms switch points [3]. The subframe indication is generated at the start of every subframe, and its timing is based on the DL.

In TDD the DL and UL subframes are offset, with the UL subframe first. Triggering, using the SUBFRAME signal on the DL, ensures the SUBFRAME indications are always 1ms apart. This ensures the DL and UL offset behaviour does not affect the L1 API and, therefore, is out of scope of this document.

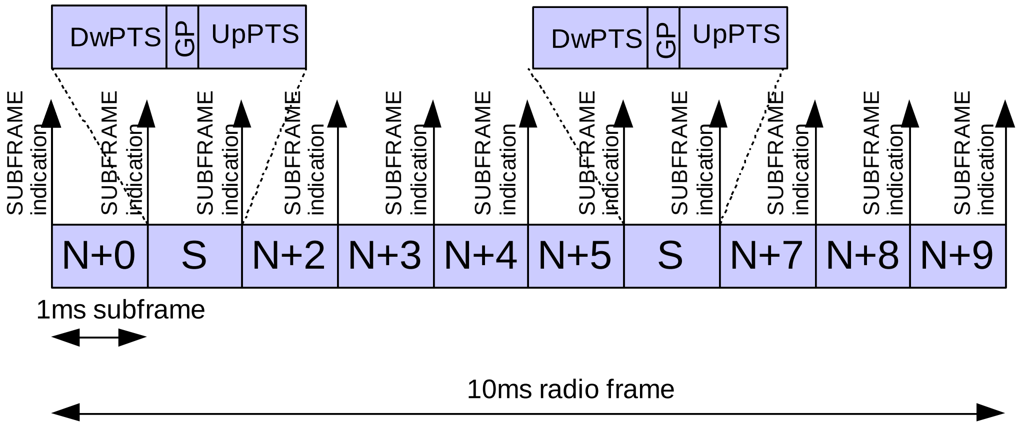

Figure 16: SUBFRAME signal for TDD using 5ms switch points

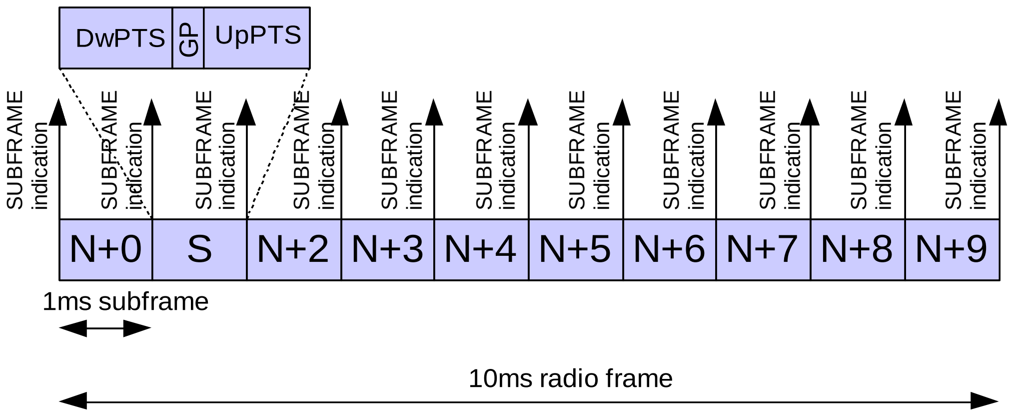

Figure 17: SUBFRAME signal for TDD using 10ms switch point

The location of the SUBFRAME signal for FDD (frame structure 1) is shown in Figure 18. The subframe indication is generated at the start of every DL subframe.

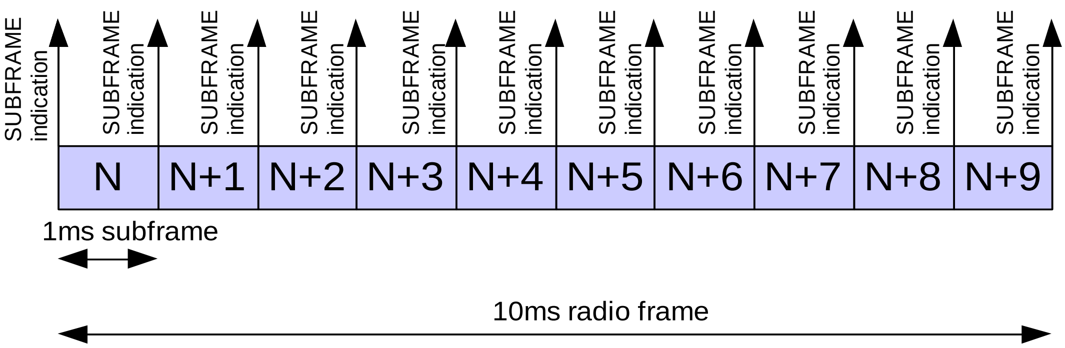

Figure 18: SUBFRAME signal for FDD

2.2.2 SFN/SF Synchronization

The SFN/SF synchronization procedure is used to maintain a consistent SFN/SF value between the L2/L3 software and PHY. Maintaining this synchronization is important since different subframes have different structures, and in TDD subframes are either downlink or uplink.

Two options are provided by the L1 API; the first option configures the PHY to use the SFN/SF value provided by the L2/L3 software. The second option configures the PHY to initialize the SFN/SF and ensure the L2/L3 software remains synchronous. The synchronisation 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/SF synchronization start-up procedure, where the L2/L3 software is master, is given in Figure 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/SF = 0. The PHY sends a SUBFRAME signal to the L2/L3 software. If a `SUBFRAME.indication` message is sent, SFN/SF = 0
- The L2/L3 software sends a `SUBFRAME.request` message to the PHY containing the correct SFN/SF = N
- The PHY uses the SFN/SF received from the L2/L3 software. It changes its internal SFN/SF to match the value provided by the L2/L3 software

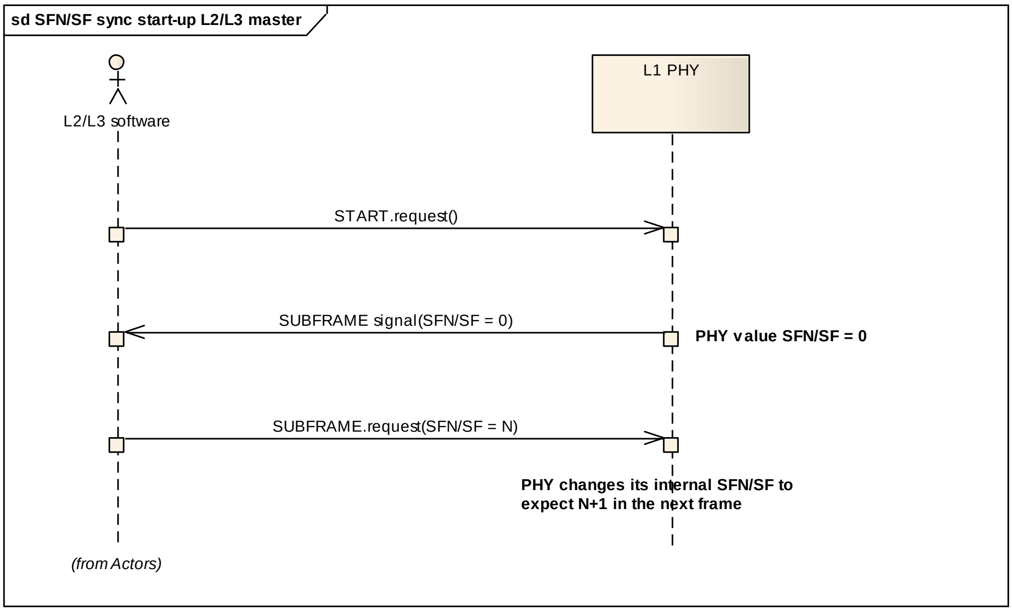

Figure 19: SFN/SF synchronization start-up with L2/L3 master

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

- The PHY sends a SUBFRAME signal to the L2/L3 software. If a `SUBFRAME.indication` message is sent, SFN/SF = M. If the PHY is not configured to send the `SUBFRAME.indication` message its internal SFN/SF count is M.
- The L2/L3 software sends a `SUBFRAME.request` message to the PHY containing SFN/SF = N
- If SFN/SF M = N
    - The PHY received the SFN/SF it was expecting. No SFN/SF synchronization is required
- If SFN/SF M ≠ N
    - The PHY received a different SFN/SF from the expected value. SFN/SF synchronization is required

- The PHY uses the SFN/SF received from the L2/L3 software. It changes its internal SFN/SF to match the value provided by the L2/L3 software
- The PHY returns an `ERROR.indication` message indicating the mismatch

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

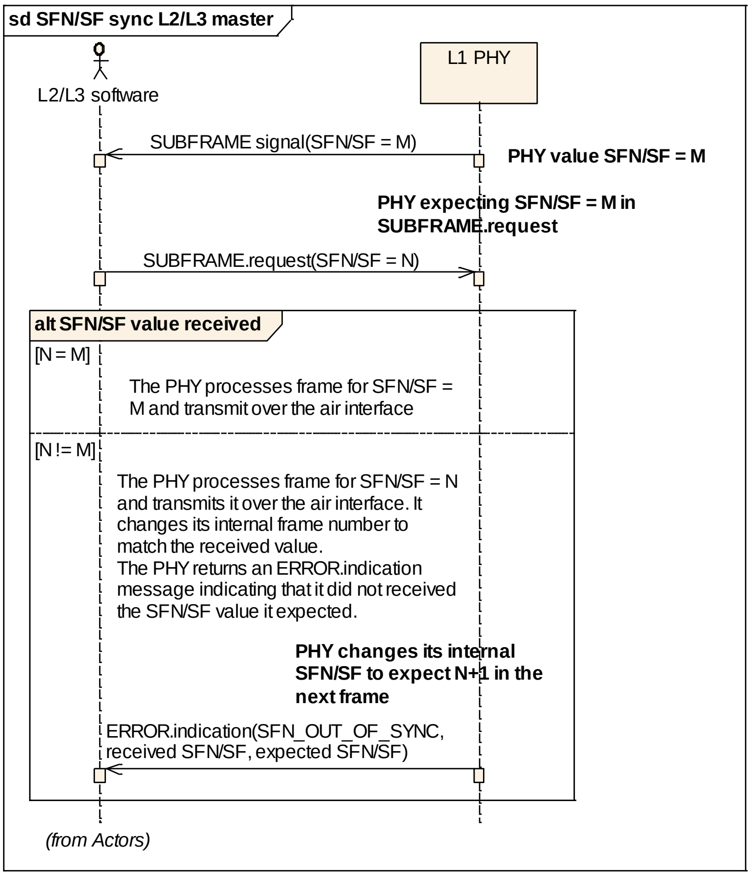

Figure 20: SFN/SF synchronization with L2/L3 master

2.2.2.2 L1 PHY is Master

The SFN/SF synchronization start-up procedure, where the L1 software is master, is given in Figure 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/SF = M. The value of M is not deterministic, and could have been set by an external mechanism, such as GPS. The PHY sends a SUBFRAME signal to the L2/L3 software. To inform the L2/L3 software of the SFN/SF value on synchronization start-up, the `SUBFRAME.indication` message is always sent by the PHY
- The L2/L3 software uses the SFN/SF received from the PHY. It changes its internal SFN/SF to match the value provided by the PHY
- The L2/L3 software sends a `SUBFRAME.request` message to the PHY containing SFN/SF = M
- When sending future SUBFRAME signals the PHY uses the mode (A or B) configured by the L2/L3 software with PHY Sync Mode TLV in the `CONFIG.request` message

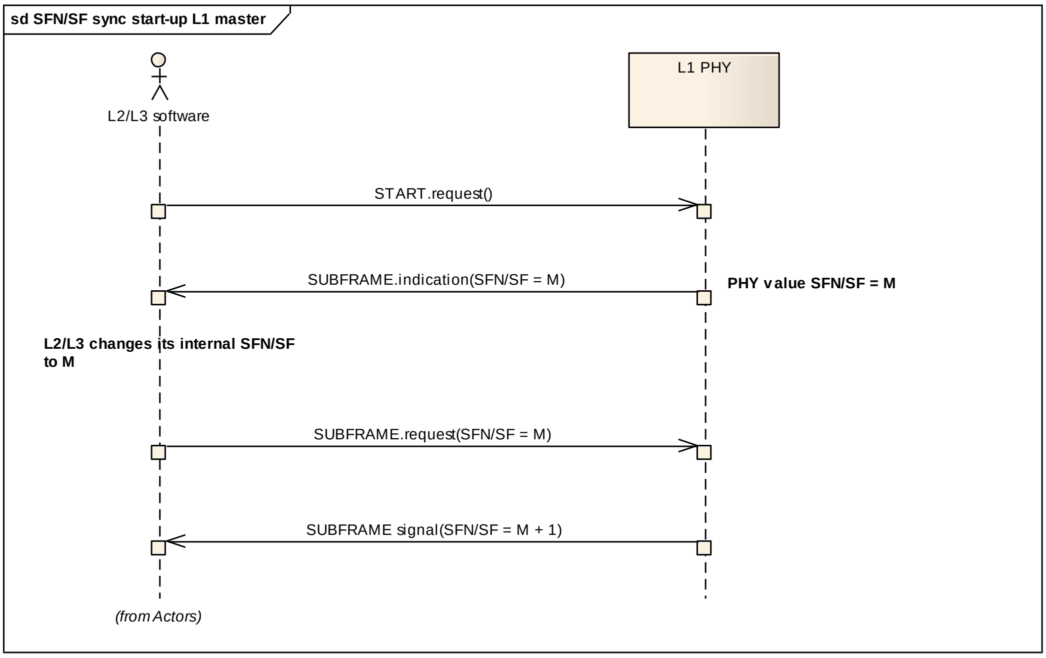

Figure 21: SFN/SF synchronization start-up with L1 master

The SFN/SF synchronization maintenance procedure, is shown in Figure 22. In this example, the L1 PHY is expecting the next `SUBFRAME.request` to contain information regarding frame M. The procedure followed is:

- The PHY sends a SUBFRAME signal to the L2/L3 software. If a `SUBFRAME.indication` message is sent, SFN/SF = M. If the PHY is not configured to send the `SUBFRAME.indication` message its internal SFN/SF count is M.
- The L2/L3 software sends a `SUBFRAME.request` message to the PHY containing SFN/SF = N
- If SFN/SF M = N
    - The PHY received the SFN/SF it was expecting. No SFN/SF synchronization is required
- If SFN/SF M ≠ N
    - The PHY received a different SFN/SF from the expected value. SFN/SF synchronization is required
    - The PHY discards the received `SUBFRAME.request` message
    - The PHY returns an `ERROR.indication` message indicating the mismatch

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

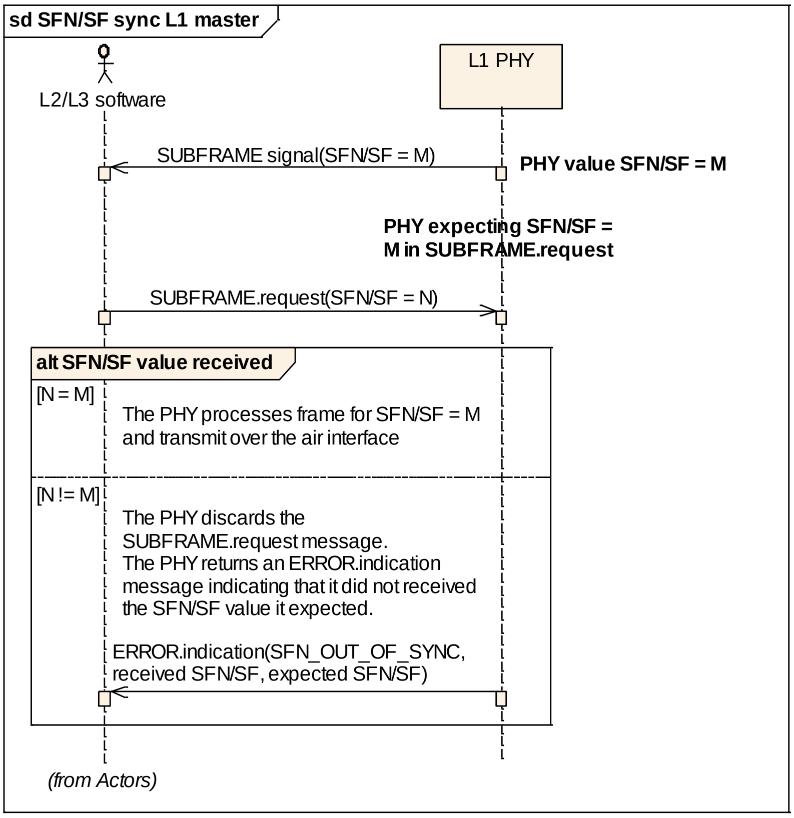

Figure 22: SFN/SF synchronization with L1 master

2.2.3 API Message Order

The L1 API has constraints of when certain subframe messages can be sent, or will be received, by the L2/L3 software.

The downlink API message constraints are shown in Figure 23:

- If the PHY is configured to send a `SUBFRAME.indication` message the included SFN/SF is expected in the corresponding `SUBFRAME.request`
- If the PHY is being reconfigured using the `CONFIG.request` message, this must be the first message for the subframe.
- The `SUBFRAME.request` must be sent for every subframe and must be the next message.
- If required, `TX.request` must be the next message.
- If present, the `HI_DCI.request` must be the next message for the subframe.
- There must be only 1 `SUBFRAME.request`, 1 `TX.request` and 1 `HI_DCI.request` for a subframe.

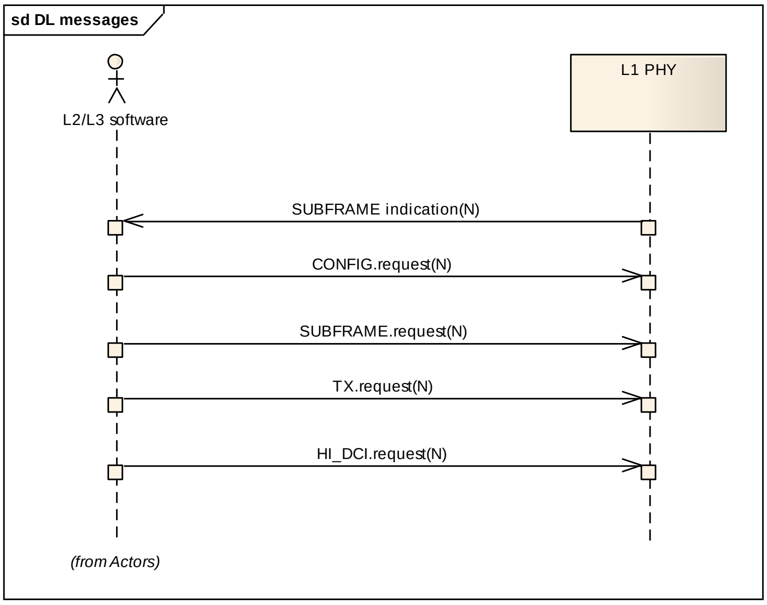

Figure 23: DL message order

The uplink API message constraints are shown in Figure 24:

- If present, the first data message for subframe N will be `HARQ.indication`
- If present, the next message will be a `SR.indication`
- The next message will be a `CRC.indication`, If configured (see section 0), this message is present in every subframe and can be used as a trigger for UL scheduling by L2/L3.
- If present, the remaining messages can be in any order
  - The `RX.indication` message includes any uplink data PDUs, CQI PDUs and SR PDUs
  - The `RACH.indication` is included if any RACH preambles were detected in the subframe
  - The `SRS.indication` includes any sounding reference symbol information
- There will be only 1 `HARQ.indication`, 1 `CRC.indication`, 1 `RX.indication`, 1 `RACH.indication` 1 `SRS.indication` message per subframe

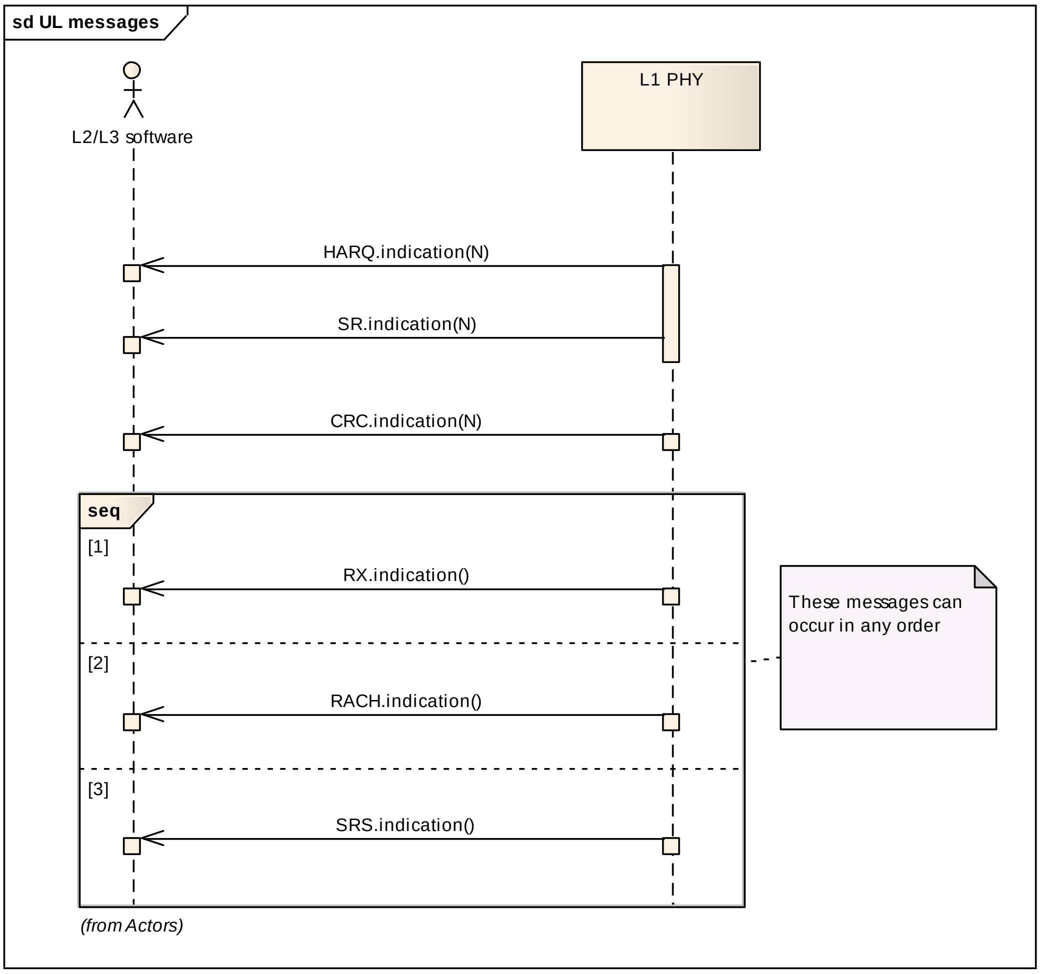

Figure 24: UL message order

2.2.4 API Message Timing

Understanding the L1 API message timing in LTE is complex. The DL-SCH and UL-SCH transport channels both use HARQ. This means, that a DL-SCH has an uplink component to return the ACK/NACK response, while the UL-SCH requires a downlink component. In addition, the scheduling of UL-SCH is performed by transmitting control information on the downlink PDCCH physical channel.

The L1 API message timing is dependent on the duplexing mode of the PHY. In TDD mode, the frame structure is made more complex by the configurable ratio of uplink and downlink subframes. In both TDD and FDD modes the ACK/NACK response is required in subframe N+K, and the UE applies the uplink PDCCH control in subframe $N+K_{PUSCH}$. For FDD K=4 and $K_{PUSCH}=4$. However, for TDD the value of K depends on the subframe structure in use [6]. In the worst case, TDD also requires K=4 and $K_{PUSCH}=4$.

LTE defines a maximum of 15 HARQ processes for the DL-SCH and 8 HARQ processes for the UL-SCH. To achieve a maximum throughput, a HARQ process must be scheduled every subframe. Therefore, the transmission of data and reception of ACK/NACK response must be completed in 8 DL, or UL, subframes.

An example L1 API message timing is shown in Figure 25 for FDD and K=4. The L1 API timing is:

- Following subframe indication N:
  - A `SUBFRAME.request` is sent to the PHY. This message includes downlink transport channels and DCI format 1 → 2 information. The DCI information relates to the downlink
  - A `TX.request` is sent to the PHY. This message includes the data for the downlink transport channels
  - A `HI_DCI.request` is sent to the PHY. This message includes DCI format 0 and 3 information, and relates to the uplink
  - The eNodeB transmits during air subframe N.

- Following subframe indication (N+4):
  - A `SUBFRAME.request` is sent to the PHY. This message includes uplink transport channels. It also includes SR and SRS information used to maintain the uplink connection. Also, it includes CQI and HARQ information which relate to the downlink connection
  - The UE transmits during air subframe (N+4).
- The `HARQ.indication`, including ACK/NACK for downlink transmission in subframe N, is sent to the L2/L3 software after subframe indication for (N+6).
- The `CRC.indication` and `RX.indication` for the uplink transmission in air subframe (N+4) is sent to the L2/L3 software after subframe indication (N+8). This uplink data corresponds to the DCI format 0 information sent in subframe N, and uplink PDUs sent in subframe N+4
- To maintain maximum data throughput, a `SUBFRAME.request` is sent to the PHY in subframe N+8 for the same HARQ process as subframe N
- A `HI_DCI.request` is sent to the PHY after subframe indication N+8, containing the ACK/NACK information for the uplink data received in subframe N+4.

More information about the physical timing of API messages is given in Appendix L1 API Message Timing.

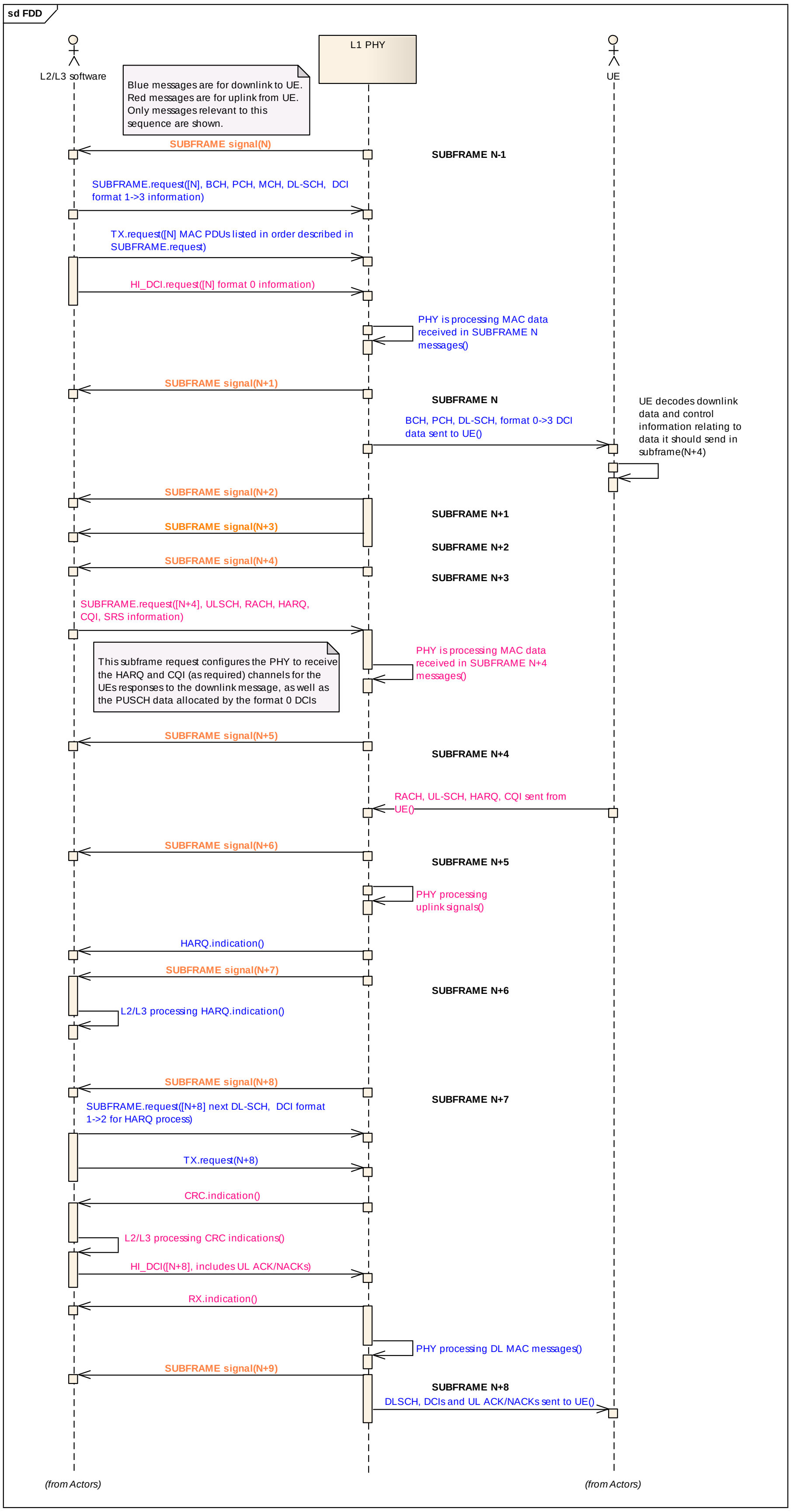

Figure 25: API Message Timing

2.2.5 Downlink

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

2.2.5.1 BCH

The BCH transport channel is used to transmit the Master Information Block (MIB) information to the UE. The location of the MIB is defined in the LTE standards [1], and shown in Figure 26. It is transmitted in subframe 0 of each radio frame. When the radio frame (SFN mod 4) = 0 an updated MIB is transmitted in subframe 0. When the radio frame (SFN mod 4) ≠ 0 the MIB is repeated.

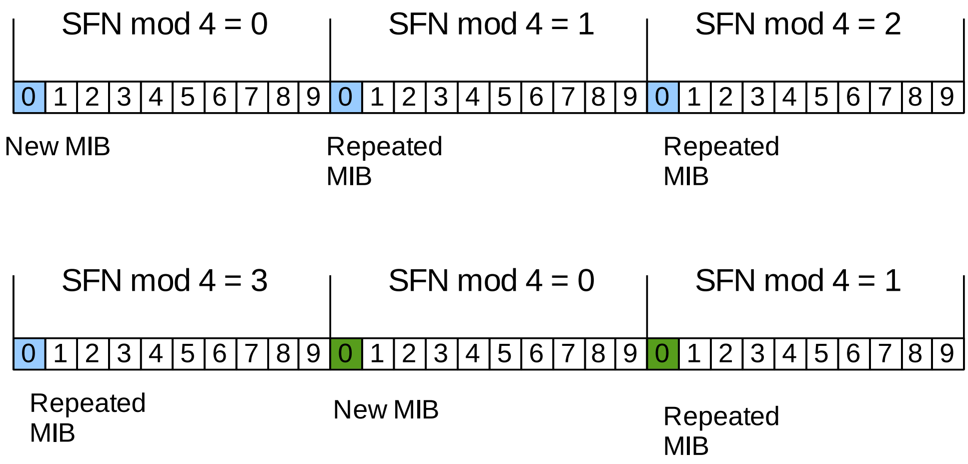

Figure 26: MIB scheduling on the BCH transport channel

The BCH procedure is shown in Figure 27. The L2/L3 software should provide a BCH PDU to the PHY in subframe SF=0, for each radio frame (SFN mod 4) = 0. This is once every 40ms. The L2/L3 software provides the following information:

- In `SUBFRAME.request` a BCH PDU is included.
- In `TX.request` a MAC PDU containing the MIB is included.

If the PHY does not receive a BCH PDU in subframe SF=0, where radio frame (SFN mod 4) = 0, it will return an `ERROR.indication` with MSG_BCH_MISSING.

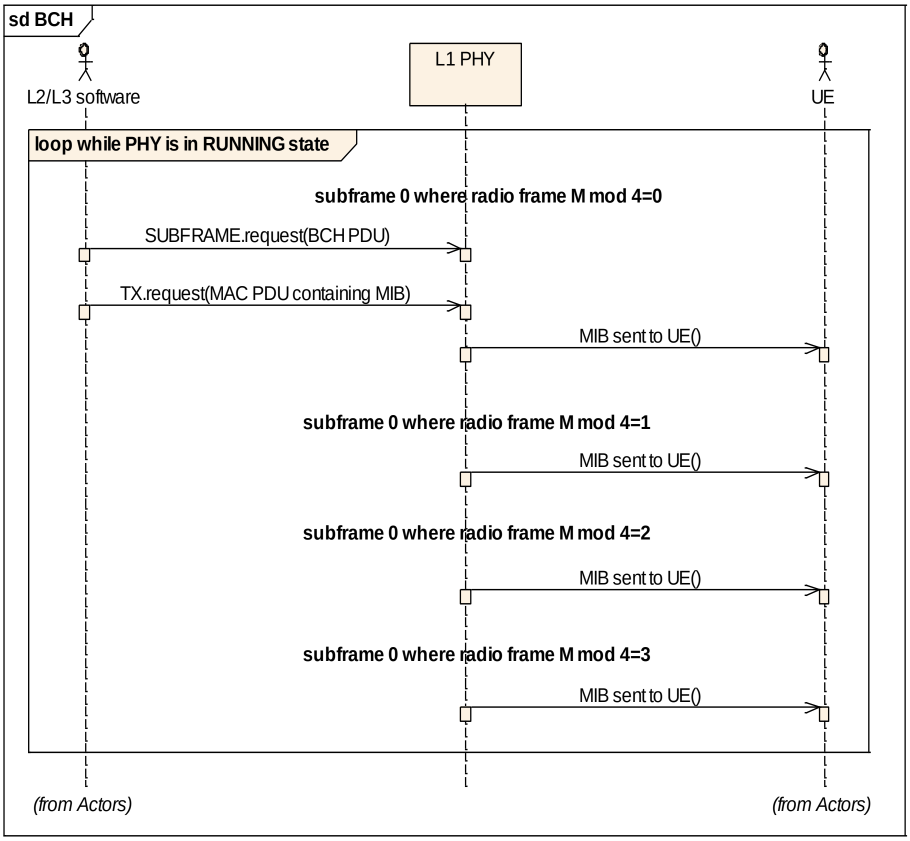

Figure 27: BCH procedure

2.2.5.2 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 [5]. 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 `SUBFRAME.request` message.

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

- In `SUBFRAME.request` a PCH PDU and DCI PDU are included.
- In `TX.request` a MAC PDU containing the paging message is included.

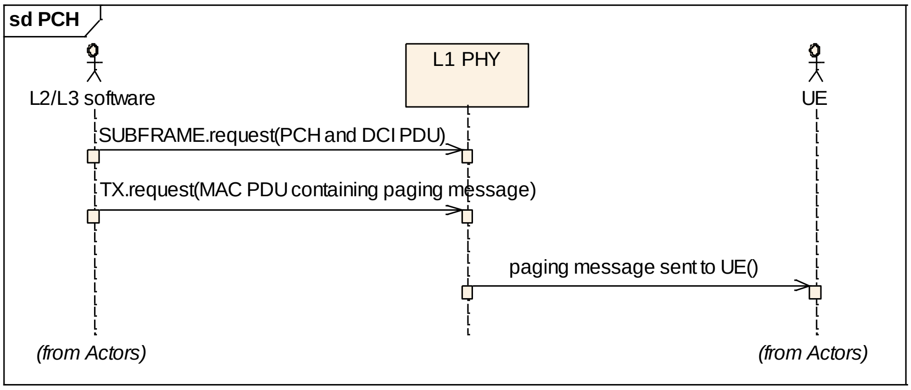

Figure 28: PCH procedure

2.2.5.3 DL-SCH

The DL-SCH transport channel is used to send data from the eNB to a single UE. This distinguishes the DL-SCH from the MCH, where data is sent to multiple UEs. HARQ is always applied on the DL-SCH transport channel. Therefore, together with scheduling downlink transmissions the L2/L3 software must schedule uplink bandwidth for the UE to return an ACK/NACK response.

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

- In `SUBFRAME.request` a DLSCH PDU and DCI Format 1 or 2 PDU are included. The DCI PDU contains control regarding the DL frame transmission. If 2 transport blocks are to be sent on two layers, then two DLSCH PDUs must be provided and they must be in sequence in `SUBFRAME.request`; these should be in layer order i.e., layer 1 then layer 2.
- In `TX.request` a MAC PDU containing the data is included. If two layers are in use, two MAC PDUs must be provided in sequence in `TX.request;` these should be in layer order i.e., layer 1 then layer 2.
- In a later `SUBFRAME.request` a HARQ PDU is included. The timing of this subframe was explained in Section 2.2.4. There are 5 possible HARQ PDUs that can be used to indicate reception of the HARQ response on the uplink:
  - ULSCH_HARQ – is used if the UE is scheduled to transmit data and the ACK/NACK response
  - ULSCH_CQI_HARQ_RI – is used if the UE is scheduled to transmit data, a CQI report and the ACK/NACK response
  - UCI_HARQ – is used if the UE is just scheduled to transmit the ACK/NACK response
  - UCI_SR_HARQ – is used if the UE is scheduled to transmit a scheduling request and the ACK/NACK response
  - UCI_CQI_HARQ – is used if the UE is scheduled to transmit a CQI report and the ACK/NACK response
- The PHY will return the ACK/NACK response information in the `HARQ.indication` message

Section 2.2.4 explained the strict timing requirements in LTE. For FDD, and worst-case TDD, to maintain maximum downlink throughput the L2/L3 software must analyse the `HARQ.indication` message within 1ms. This allows the L2/L3 software to generate the next `SUBFRAME.request` message, using the same HARQ process, in subframe N+8.

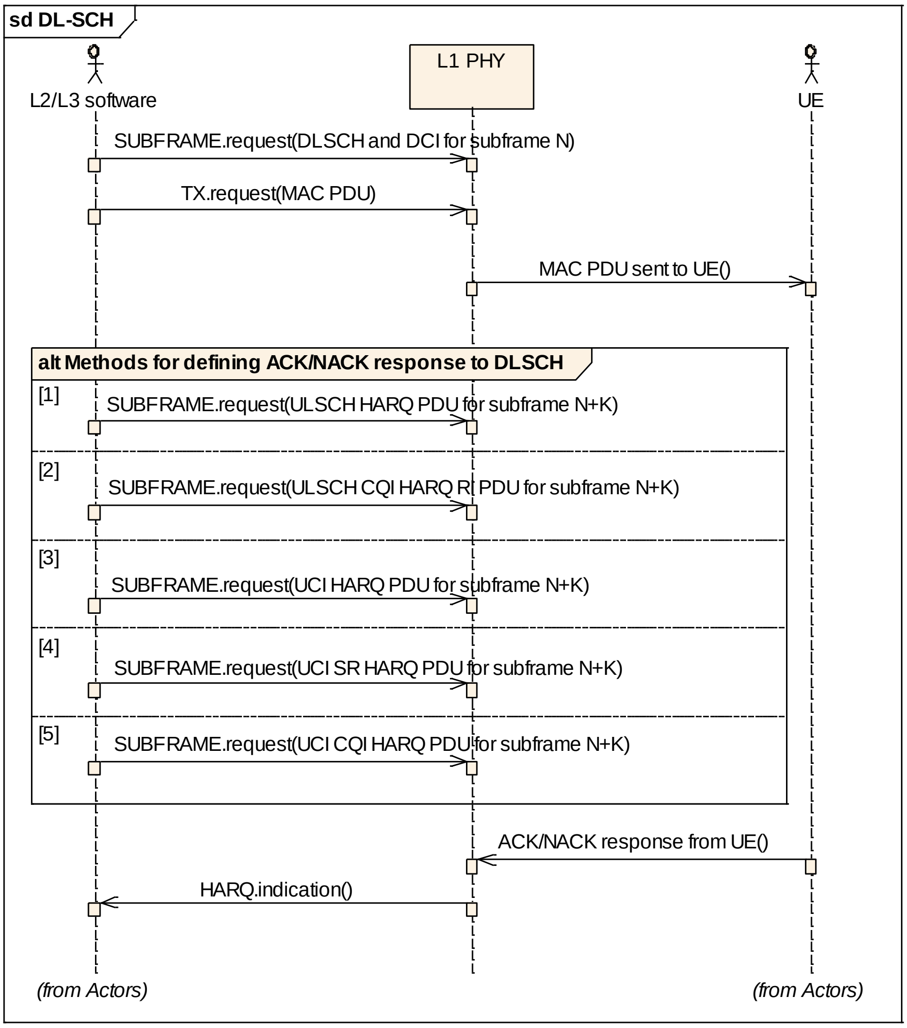

Figure 29: DL SCH procedure

2.2.5.4 MCH

The MCH transport channel is used to send data, simultaneously, to multiple UEs. This means that HARQ is not used on this transport channel.

The MCH procedure is shown in Figure 30. To transmit a MCH PDU the L2/L3 software must provide the following information:

- In `SUBFRAME.request` a MCH PDU is included
- In `TX.request` a MAC PDU containing the multicast data is included

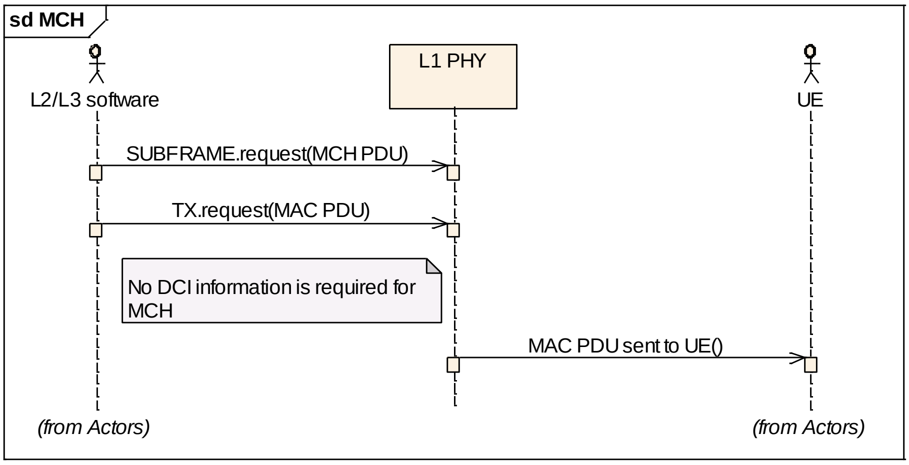

Figure 30: MCH procedure

2.2.6 Uplink

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

2.2.6.1 RACH

The RACH transport channel is used by the UE to send data to the eNB 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 L1 API, the RACH procedure begins when the PHY receives a `SUBFRAME.request` message indicating the presence of a RACH.

The RACH procedure is shown in Figure 31. To configure a RACH procedure the L2/L3 software must provide the following information:

- In `SUBFRAME.request` the RACH present field must be set
- If a UE decides to RACH, and a preamble is detected by the PHY:
    - The PHY will include 1 RACH PDU in the `RACH.indication` message. This RACH PDU includes all detected preambles
- If no RACH preamble is detected by the PHY, then the behaviour depends upon the RACH & SR Report Mode TLV (see Table 9)
    - If the TLV is set to 0, a `RACH.indication` message is sent containing zero preambles
    - If the TLV is set to 1, no indication message is produced.

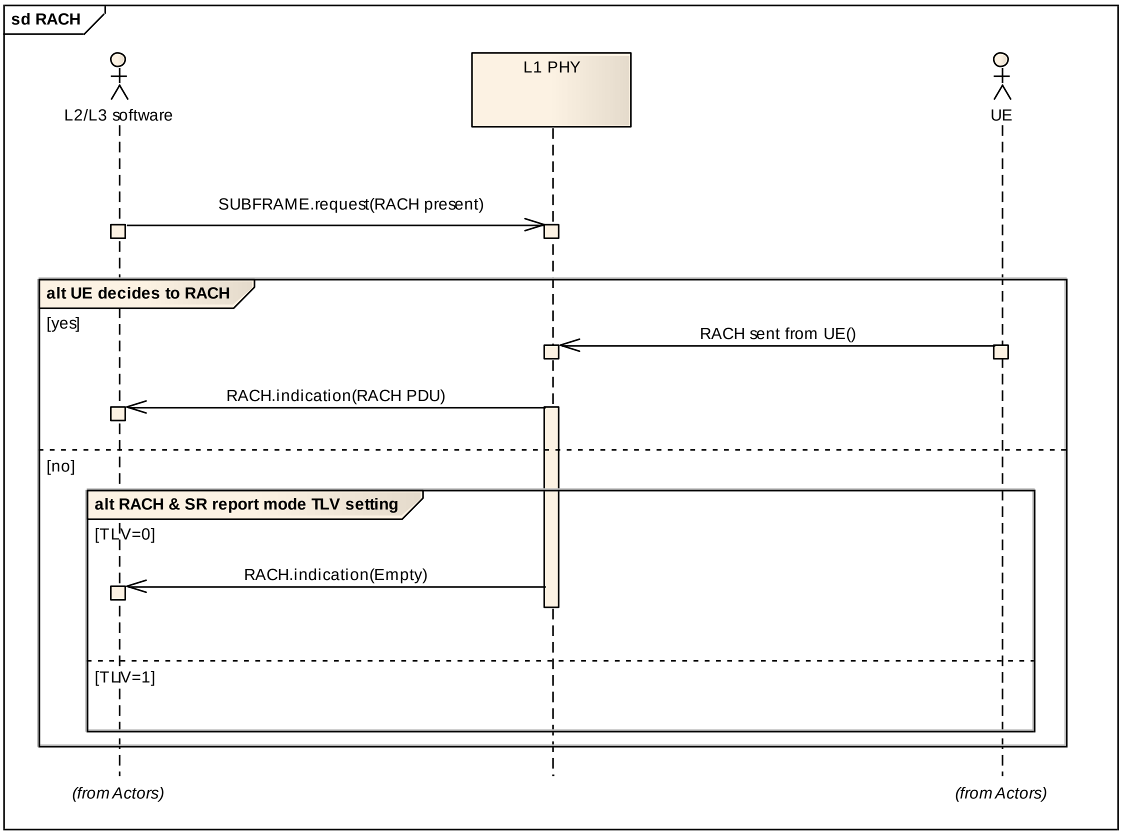

Figure 31: RACH procedure

2.2.6.2 UL-SCH

The UL-SCH transport channel is used to send data from the UE to the eNB. HARQ is always applied on the UL-SCH transport channel. Therefore, together with scheduling uplink transmissions the L2/L3 software must schedule downlink ACK/NACK responses.

The procedure for the UL-SCH transport channel is shown in Figure 32. To transmit an UL-SCH PDU the L2/L3 software must provide the following information:

- Within the `HI_DCI.request` for subframe N a DCI PDU is included. The DCI Format 0 PDU contains control information regarding the UL frame transmission being scheduled.
- In `SUBFRAME.request` for subframe N+K1 an ULSCH PDU is included. The timing of this subframe, and value of K1, are explained in Section 2.2.4. There are 4 possible ULSCH PDUs that can be used to schedule ULSCH data on the uplink:
    - ULSCH – is used if the UE is scheduled to only transmit data
    - ULSCH_CQI_RI – is used if the UE is scheduled to transmit data and a CQI report
    - ULSCH_HARQ – is used if the UE is scheduled to transmit data and an ACK/NACK response
    - ULSCH_CQI_HARQ_RI – is used if the UE is scheduled to transmit data, a CQI report and an ACK/NACK response
- If MU-MIMO is in use, two sequential ULSCH PDUs must be provided in `SUBFRAME.request`. The two ULSCH allocations must occupy the same resource blocks.
- If the Data Report Mode TLV = 0 in the `CONFIG.request` message, then:
    - 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.indication` message. The `RX.indication` message repeats the CRC information given in the `CRC.indication` message

- The ACK/NACK response must be submitted to the PHY using a `HI_DCI.request` message in subframe N+K1+K2. The timing of this subframe, and value of K2, was explained in Section 2.2.4.

Section 2.2.4 explained the strict timing requirements in LTE. For FDD and worst-case TDD, the L2/L3 software must analyse the CRC information and generate the `HI_DCI.request` message within 250us.

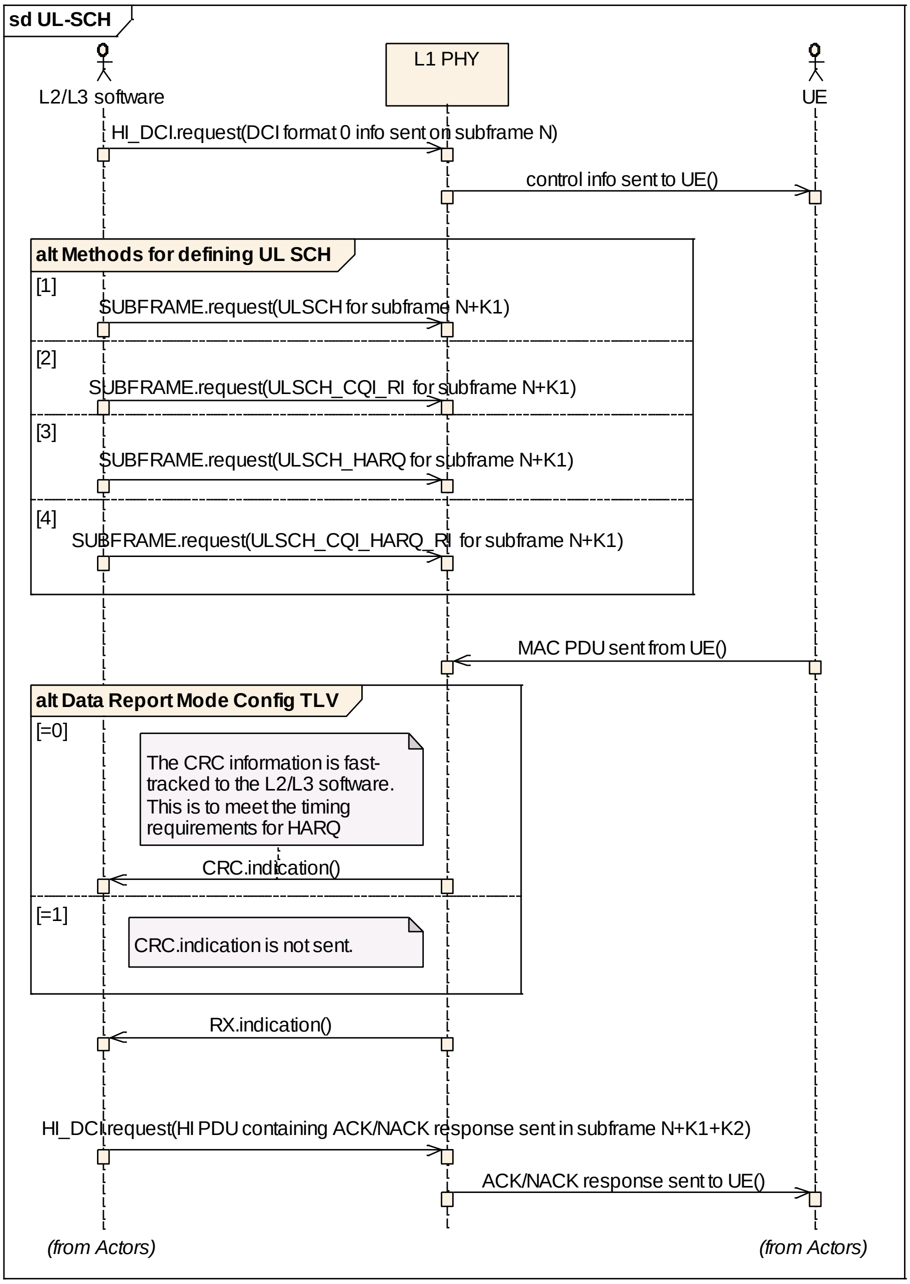

Figure 32: ULSCH procedure

2.2.6.3 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 33. To schedule a SRS the L2/L3 software must provide the following information:

- In `SUBFRAME.request` the SRS present field must be set and one SRS PDU per sounding UE is included
- The PHY will return the SRS response to the L2/L3 software in the `SRS.indication` message

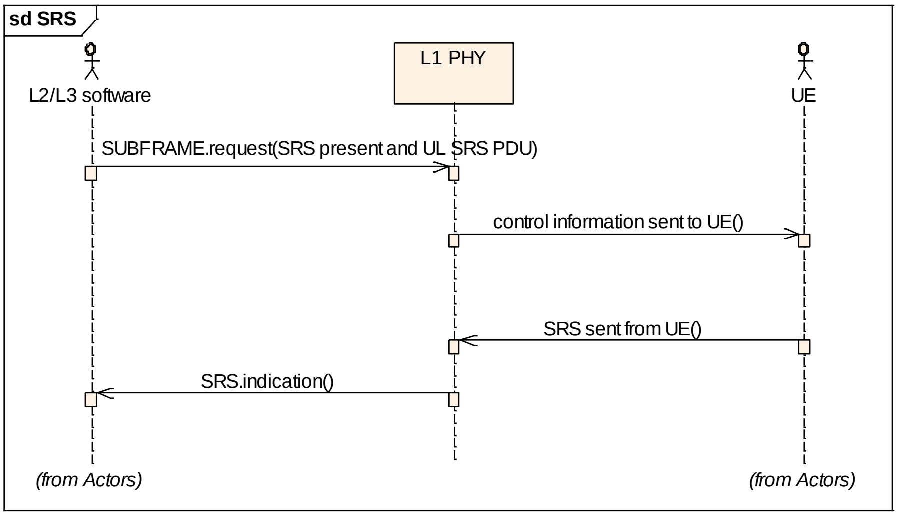

Figure 33: SRS procedure

2.2.6.4 CQI

The CQI reporting mechanism is used by the L2/L3 software to determine the quality of the downlink channel. CQI reporting is initiated through two methods. Firstly, during the RRC connection procedure the L2/L3 software will instruct the UE to transmit periodic CQI reports. Secondly, the L2/L3 software can use the PDCCH to instruct the UE to transmit an aperiodic CQI report.

In both cases the PHY is only responsible for receiving CQI reports when instructed by the L2/L3 software.

The CQI reporting procedure is shown in Figure 34. To schedule a CQI report the L2/L3 software must provide the following information:

- For an aperiodic report the DCI format 0 PDU is included in the `HI_DCI.request`. This instructs the UE to send a CQI report. For periodic CQI reports no explicit DCI information is sent.
- In the `SUBFRAME.request`, where the L2/L3 software is expecting a CQI report, a CQI PDU is included. There are 4 possible CQI PDUs that can be used to indicate reception of the CQI report on the uplink:
  - ULSCH_CQI_RI – is used if the UE is scheduled to transmit data and a CQI report
  - ULSCH_CQI_HARQ_RI – is used if the UE is scheduled to transmit data, a CQI report and the ACK/NACK response
  - UCI_CQI – is used if the UE is just scheduled to transmit a CQI report
  - UCI_CQI_HARQ – is used if the UE is scheduled to transmit a CQI report and the ACK/NACK response
- The PHY will return the CQI report to the L2/L3 software in the `RX.indication` message

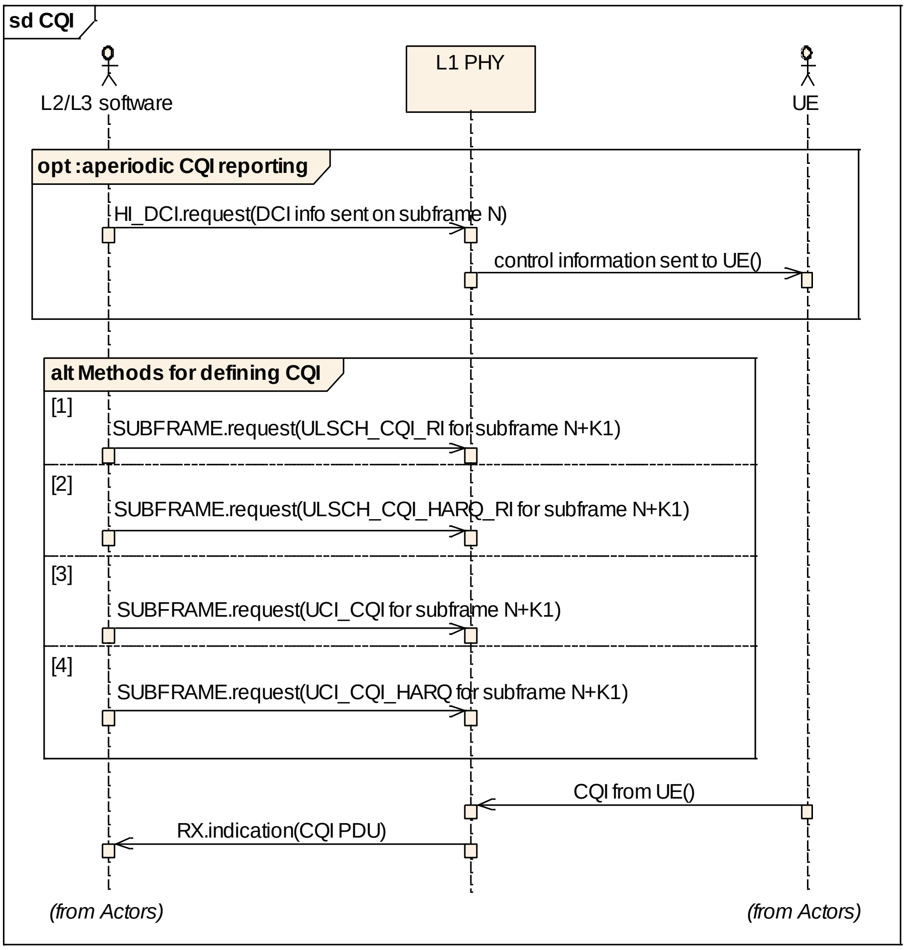

Figure 34: CQI procedure

2.2.6.5 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 PHY is only responsible for receiving SR information when instructed by the L2/L3 software.

The SR procedure is shown in Figure 35. To schedule a SR the L2/L3 software must provide the following information:

- In the `SUBFRAME.request` a SR PDU is included. There are 2 possible SR PDUs that can be used to indicate reception of the SR on the uplink:
  - UCI_SR – is used if the UE is just scheduled to transmit a SR
  - UCI_SR_HARQ – is used if the UE is scheduled to transmit a SR and the ACK/NACK response
- If the UE transmits a positive SR and it is received successfully, the PHY will return the SR to the L2/L3 software in the `SR.indication` message
- If no positive SR are received, the behaviour depends on the RACH & SR Report Mode TLV (see Table 9)
  - If the TLV is set to 0, an empty `SR.indication` will be generated
  - If the TLV is set to 1, no message is generated

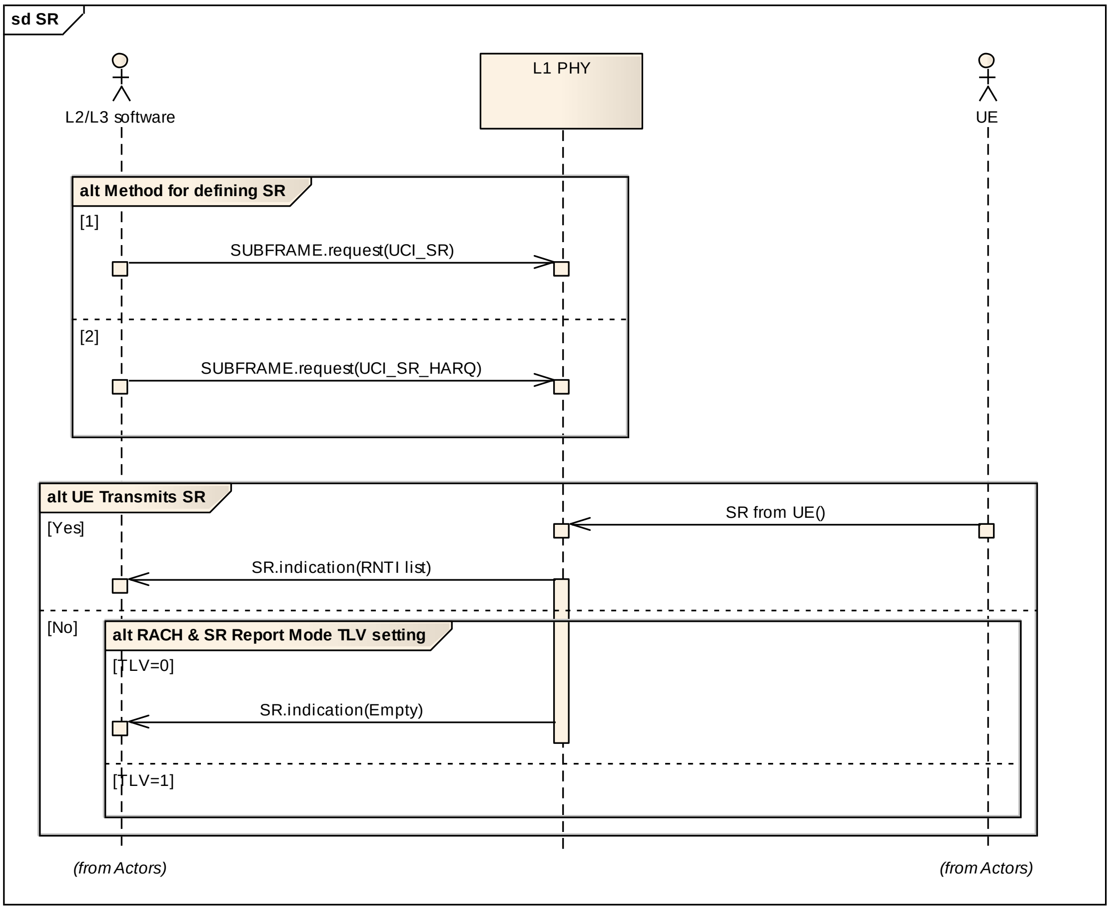

Figure 35: SR procedure

2.2.7 Error Sequences

The error sequences used for each subframe procedure are shown in Figure 36 to Figure 38. In all subframe procedures errors that are detected by the PHY are reported using the `ERROR.indication` message. In section 3.2.6.1, the L1 API message definitions include a list of error codes applicable for each message.

The `SUBFRAME.request`, `HI_DCI.request` and `TX.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 `SUBFRAME.request`, `HI_DCI.request` or `TX.request`, it should assume that the UE did not receive data and control sent in this subframe. This is similar to the UE experiencing interference on the air-interface and LTE mechanisms, such as, HARQ and ARQ, will enable the L2/L3 software to recover.

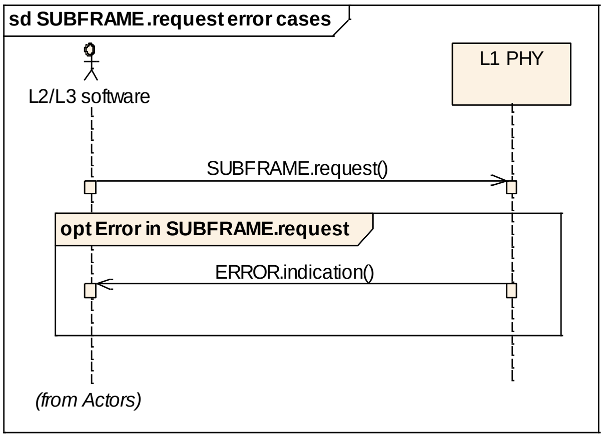

Figure 36: `SUBFRAME.request` error sequence

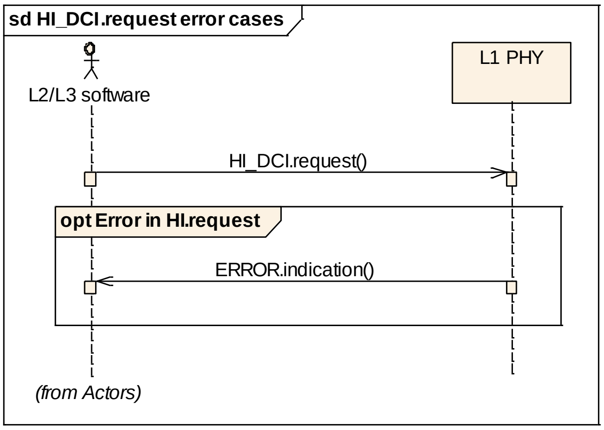

Figure 37: `HI_DCI.request` error sequence

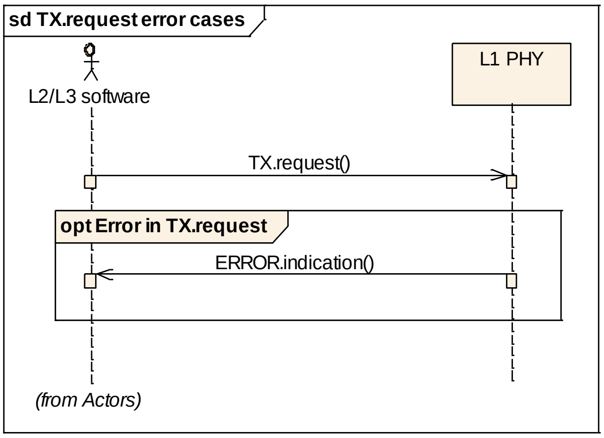

Figure 38: `TX.request` error sequence

3 L1 API Messages

This section provides a description of the L1 API message formats. It defines the L1 API message header, the message bodies and the error codes associated with the L1 API.

3.1 General Message Format

The general message format of the L1 API is shown in Table 3, where it can be seen that each L1 API message consists of a header followed by a message body.

The generic header consists of 4-bytes, including a message type ID and message body length. The current list of message types is given in Table 4. The L1 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 3.2.6.1.
- All multi-byte parameters (e.g. 16 bit, 32 bit) shall be aligned on boundaries of the same size (e.g. 2 byte and 4 byte respectively).

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. More information on endianess, together with a description of the byte encoding, is given in Section 4.

Size(bytes)	***Description***
1	Message type ID
1	Reserved
2	Length of message body (bytes)
Message body	

Table 3: General L1 API message format

Message	***Value***	***Message Body Definition***
`PARAM.request`	0x00	See Section 3.2.1.1
`PARAM.response`	0x01	See Section 3.2.1.2
`CONFIG.request`	0x02	See Section 3.2.2.1
`CONFIG.response`	0x03	See Section 3.2.2.2
`START.request`	0x04	See Section 3.2.4.1
`STOP.request`	0x05	See Section 3.2.5.1
`STOP.indication`	0x06	See Section 3.2.5.2

Message	*Value*	*Message Body Definition*
`ERROR.indication`	0x07	See Section 3.2.6.1
RESERVED	0x08-0x7f	
`SUBFRAME.request`	0x80	See Section 3.3.1.2
`SUBFRAME.indication`	0x81	See Section 3.3.1.1
`HI_DCI.request`	0x82	See Section 3.3.1.3
`TX.request`	0x83	See Section 3.3.2.1
`HARQ.indication`	0x84	See Section 3.3.3.2
`CRC.indication`	0x85	See Section 3.3.3.3
`RX.indication`	0x86	See Section 3.3.3.1
`RACH.indication`	0x87	See Section 3.3.3.5
`SRS.indication`	0x88	See Section 3.3.3.6
`SR.indication`	0x89	See Section 3.3.3.4
RESERVED	0x8a-0xff	

Table 4: L1 API Message Types

3.2 Configuration Messages

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

3.2.1 PARAM

The PARAM message exchange was described in Figure 6.

3.2.1.1 PARAM.request

This message can be sent by the L2/L3 when the PHY is in the IDLE and CONFIGURED states. If it is sent when the PHY is in the RUNNING state, a MSG_INVALID_STATE error is returned in `PARAM.response`. No message body is defined for `PARAM.request`. The message length in the generic header is, therefore, zero.

3.2.1.2 PARAM.response

The `PARAM.response` is given in Table 5. 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 TLVs is given in Table 9, along with their applicability to `PARAM.response` in various configurations and states. Note: There is no requirement for the PHY to return the TLV's in the order specified in the table.

Field	*Size (bytes)*	*Description*
Error Code	1	See 3.2.6.1.1
Number of TLVs	1	Number of TLVs contained in the message body.

Field	*Size (bytes)*	*Description*
Reserved	2	Align to 32-bit boundary
TLVs	Variable	See section 0

Table 5: `PARAM.response` message body

3.2.1.3 PARAM Errors

Section removed. Please see section 3.2.6.1.

3.2.2 CONFIG

The CONFIG message exchange was described in Figure 7.

3.2.2.1 CONFIG.request

The `CONFIG.request` message is given in Table 6. 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 TLVs is given in **Table 9**. There is no requirement for the L2/L3 software to provide the TLVs in the order specified in the Tables.

Field	*Size (bytes)*	*Description*
Number of TLVs	1	Number of TLVs contained in the message body.
Reserved	3	Align to 32-bit boundary
TLVs	Variable	See section 0

Table 6: `CONFIG.request` message body

3.2.2.2 CONFIG.response

The `CONFIG.response` message is given in Table 7. 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.

Field	*Size (bytes)*	*Description*
Error Code	1	See 3.2.6.1.1
Number of Invalid or Unsupported TLVs	1	Number of invalid or unsupported TLVs contained in the message body.
Number of Missing TLVs	1	Number of missing TLVs contained in the message body. If the PHY is in the CONFIGURED state this will always be 0.
Reserved	1	Align to 32-bit boundary
A list of invalid or unsupported TLVs – each TLV is presented in its entirety. Each TLV is, by definition, aligned to a 32-bit boundary, so no additional padding is required.		

Field	*Size (bytes)*	*Description*
TLV	Variable	Complete TLVs
A list of missing TLVs – each TLV is presented in its entirety. Each TLV is, by definition, aligned to a 32-bit boundary, so no additional padding is required.		
TLV	Variable	Complete TLVs

Table 7: `CONFIG.response` message body

3.2.2.3 CONFIG Errors

Section removed. Please see section 3.2.6.1.

3.2.3 Configuration TLVs

The configuration TLVs that are used in the PARAM and CONFIG message exchanges follow the format given in Table 8. Each TLV consists of; a Tag parameter of 1 byte, a Length parameter of 1 byte and a Value parameter. The length of the Value parameter ensures the complete TLV is a multiple of 4-bytes (32-bits).

Size(bytes)	*Description*
Byte 0	Tag
Byte 1	Length (in bytes)
Byte 2 to (Length+2)	Value

Table 8: TLV format

The individual TLVs are defined in Table 9. In this table, columns headed by "I" refer to behaviour in the IDLE state, whilst columns headed by "C" refer to behaviour in the configured state and columns headed by "R" refer to the running state. For `PARAM.response` messages, a "✓" indicates that this TLV will be included in the response. For `CONFIG.request` messages, a "M" indicates that this is a mandatory TLV and must be included in the request; an "O" indicates that this TLV is optional and may be included in the request.

Description	Tag	Len	Value	TDD PARAM.RES		FDD PARAM.RES		TDD CONFIG.REQ			FDD CONFIG.REQ		
				I	C	I	C	I	C	R	I	C	R
These TLVs are used by the L2/L3 software to configure a physical parameter in L1													
Duplexing Mode	1	2	Type of duplexing mode Value : 0 : TDD, 1 : FDD, 2 : HD_FDD		✓		✓	M	O		M	O	
Tx Antenna Ports	2	2	The number of cell specific transmit antenna ports. See [8] section 6.2.2. Value:1,2,4		✓		✓	M	O		M	O	
Rx Antenna Ports	3	2	The number of cell specific receive antenna ports. See [8] section 6.2.2. Value: 1, 2, 4		✓		✓	M	O		M	O	
Physical Cell ID	4	2	The Cell ID sent with the synchronization signal. See [8] section 6.11. Value: 0 → 503		✓		✓	M	O		M	O	
Downlink Channel Bandwidth	5	2	Downlink channel bandwidth in resource blocks. See [7] section 5.6. Value: 6,15, 25, 50, 75, 100		✓		✓	M	O		M	O	
Uplink Channel Bandwidth	6	2	Uplink channel bandwidth in resource blocks. See [7] section 5.6 Value: 6,15, 25, 50, 75,100		✓		✓	M	O		M	O	
Reference Signal Power	7	2	Sets the level of the reference signal at the output of the PHY transmit path. Value: 0 → 110. If the PHY is transmitting a fully loaded downlink with all channels at 0dB with respect to the reference signal, then this represents -125dBFS to -15dBFS average power in 1dB steps.		✓		✓	M	O	O	M	O	O
Synchronization signal power offset	8	2	The power of primary and secondary synchronization signals with respect to the reference signal, Value: 0 → 10000 represents -6dB to 4dB in step 0.001dB		✓		✓	M	O	O	M	O	O

Description	Tag	Len	Value	TDD PARAM.RES		FDD PARAM.RES		TDD CONFIG.REQ			FDD CONFIG.REQ		
				I	C	I	C	I	C	R	I	C	R
CFI Power Offset	9	2	The power of CFI with respect to the reference signal, Value: 0 → 10000 represents -6dB to 4dB in step 0.001dB		✓		✓	M	O	O	M	O	O
DCI power offset	10	2	The power of DCI with respect to the reference signal, Value: 0 → 10000 represents -6dB to 4dB in step 0.001dB		✓		✓	M	O	O	M	O	O
PRACH Configuration Index	11	2	Provides information about the location and format of the PRACH. See [8] section 5.7. Table 5.7.1-2 for FDD, Table 5.7.1-3 for TDD. The value is an index into the referenced tables. Value: 0 → 63		✓		✓	M	O		M	O	
PRACH Root Sequence Index	12	2	PRACH Root sequence index. See [8] section 5.7.2. Value: 0 → 837		✓		✓	M	O		M	O	
PRACH Zero Correlation Zone Configuration	13	2	Equivalent to N_{cs}, see [8] section 5.7.2. TDD: 0 → 6 FDD: 0 → 15		✓		✓	M	O		M	O	
PRACH High Speed Flag	14	2	Indicates if unrestricted, or restricted, set of preambles is used. See [8] section 5.7.2. 0: HS_UNRESTRICTED_SET 1: HS_RESTRICTED_SET		✓		✓	M	O		M	O	
PRACH Frequency Offset	15	2	The first physical resource block available for PRACH. see [8] section 5.7.1 Value: 0 → UL_channel_bandwidth – 6		✓		✓	M	O		M	O	
PUSCH Uplink RS Hopping	16	2	Indicates the type of hopping to use. See [8] section 5.5.1. 0: RS_NO_HOPPING 1: RS_GROUP_HOPPING 2: RS_SEQUENCE_HOPPING		✓		✓	M	O		M	O	

Description	*Tag*	*Len*	*Value*	*TDD PARAM.RES*		*FDD PARAM.RES*		*TDD CONFIG.REQ*			*FDD CONFIG.REQ*		
				I	C	I	C	I	C	R	I	C	R
PUSCH Group Assignment (Delta sequence-shift pattern)	17	2	The sequence shift pattern for PUSCH. See [8] section 5.5.1 Values: 0 → 29		✓		✓	M	O		M	O	
PUSCH Cyclic Shift 1 for DMRS	18	2	Specifies the cyclic shift for the reference signal used in the cell. See [8] section 5.5.1. The value is an index into the referenced table. Value: 0 → 7		✓		✓	M	O		M	O	
PUSCH Hopping Mode	19	2	If hopping is enabled indicates the type of hopping used. See [8] section 5.3.4 0: HM_INTER_SF 1: HM_INTRA_INTER_SF		✓		✓	M	O		M	O	
PUSCH Hopping Offset	20	2	The offset used if hopping is enabled. See [8] section 5.3.4 Value: 0 → 98		✓		✓	M	O		M	O	
PUSCH Number of Sub-bands	21	2	The number of sub-bands used for hopping. See [8] section 5.3.4. Value: 1 → 4		✓		✓	M	O		M	O	
PUCCH Delta Shift	22	2	The cyclic shift difference. See [8] section 5.4.1. Value: 1 → 3		✓		✓	M	O		M	O	
PUCCH N_CQI RB	23	2	The bandwidth, in units of resource blocks, that is available for use by PUCCH formats 2/2a/2b transmission in each slot. See Section 5.4 in [8]. Value: 0 → 98		✓		✓	M	O		M	O	

Description	***Tag***	***Len***	***Value***	***TDD PARAM.RES***		***FDD PARAM.RES***		***TDD CONFIG.REQ***			***FDD CONFIG.REQ***		
				I	**C**	**I**	**C**	**I**	**C**	**R**	**I**	**C**	**R**
PUCCH N_AN CS	24	2	The number of cyclic shifts used for PUCCH formats 1/1a/1b in a resource block with a mix of formats 1/a/1/ab and 2/2a/2b. See Section 5.4 in [8]. Value: 0 → 7		✓		✓	**M**	**O**		**M**	**O**	
PUCCH N1Pucch-AN	25	2	$N^{(1)}_{\mathrm{PUCCH}}$, see [6] section 10.1 Value: 0 → 2047		✓		✓	**M**	**O**		**M**	**O**	
SRS Bandwidth Configuration	26	2	The available SRS bandwidth of the cell. See [8] section 5.5.3. The value is an index into the referenced table. Value: 0 → 7		✓		✓	**M**	**O**		**M**	**O**	
SRS MaxUpPTS	27	2	Used for TDD only and indicates how SRS operates in UpPTS subframes. See [8] section 5.5.3.2 and [6] section 8.2 0: Disabled 1: Enabled		✓			**M**	**O**				
PHICH Resource	28	2	The number of resource element groups used for PHICH. See [8] section 6.9. 0: PHICH_R_ONE_SIXTH 1: PHICH_R_HALF 2: PHICH_R_ONE 3: PHICH_R_TWO		✓		✓	**M**	**O**		**M**	**O**	
PHICH Duration	29	2	The PHICH duration for MBSFN and non-MBSFN sub-frames. See [8] section 6.9 0: PHICH_D_NORMAL 1: PHICH_D_EXTENDED		✓		✓	**M**	**O**		**M**	**O**	
Subframe Assignment	30	2	For TDD mode only, indicates the DL/UL subframe structure. See [8] section 4.2. Value: 0 → 6		✓			**M**	**O**				

Description	***Tag***	***Len***	***Value***	***TDD PARAM.RES***		***FDD PARAM.RES***		***TDD CONFIG.REQ***			***FDD CONFIG.REQ***		
				I	**C**	**I**	**C**	**I**	**C**	**R**	**I**	**C**	**R**
Special Subframe Patterns	31	2	For TDD mode only. Length of fields DwPTS, GP and UpPTS. See [8] section 4.2. Value: 0 → 8		✓			**M**	**O**				
P-B	32	2	Refers to downlink power allocation. See [6] section 5.2 Value is an index into the referenced table. Value: 0 → 3		✓		✓	**M**	**O**		**M**	**O**	
Cyclic Prefix Type	33	2	Cyclic prefix type, used for DL and UL. See [8] section 5.2.1 0: CP_NORMAL, 1: CP_EXTENDED.		✓		✓	**M**	**O**		**M**	**O**	
These TLVs are used by L1 to report its physical capabilities to the L2/L3 software													
Downlink Bandwidth Support	40	2	The PHY downlink channel bandwidth capability (in resource blocks). See [7] section 5.6 Value: bitX :0 = no support, 1= support. Bit0: 6 Bit1: 15 Bit2: 25 Bit3: 50 Bit4: 75 Bit5: 100	✓	✓	✓	✓						

Description	*Tag*	*Len*	*Value*	*TDD PARAM.RES*		*FDD PARAM.RES*		*TDD CONFIG.REQ*			*FDD CONFIG.REQ*		
				I	C	I	C	I	C	R	I	C	R
Uplink Bandwidth Support	41	2	The PHY uplink channel bandwidth capability (in resource blocks). See [7] section 5.6 Value: bitX :0 = no support, 1= support. Bit0: 6 Bit1: 15 Bit2: 25 Bit3: 50 Bit4: 75 Bit5: 100	✓	✓	✓	✓						
Downlink Modulation Support	42	2	The PHY downlink modulation capability. Value: bitX :0 = no support, 1= support. Bit0: QPSK Bit1: 16QAM Bit2: 64QAM	✓	✓	✓	✓						
Uplink Modulation Support	43	2	The PHY uplink modulation capability. Value: bitX :0 = no support, 1= support. Bit0: QPSK Bit1: 16QAM Bit2: 64QAM	✓	✓	✓	✓						
PHY Antenna Capability	44	2	Number of antennas supported. Value: 0, 1, 2, 4	✓	✓	✓	✓						
SRS configuration support	45	2	Bit-map defining which of the SRS configurations defined in 36.211 5.5.3.3-1 (FDD) or 5.5.3.3-2 (TDD) are supported. Configuration 0 maps to bit 0 and a 1 in the bit-map indicates that the configuration is supported by the PHY. For example 0x0005 would indicate support for configurations 0 and 2.	✓	✓	✓	✓						

Description	*Tag*	*Len*	*Value*	*TDD PARAM.RES*		*FDD PARAM.RES*		*TDD CONFIG.REQ*			*FDD CONFIG.REQ*		
				I	C	I	C	I	C	R	I	C	R
PHY Version	46	6	Reports the release version of this PHY. Byte 0: 0 for a formal (customer) release, otherwise for internal use. Byte 1: Patch release number Byte 2: Minor release number Byte 3: Major release number Bytes 4 & 5: Reserved	✓	✓	✓	✓						
These TLVs are used by the L2/L3 software to configure the interaction between L2/L3 and L1													
PHY SYNC Mode	51	2	The method used by the PHY to signal the 1ms subframe 1: MODE A `SUBFRAME.indication` is not sent from the PHY. Only the subframe interrupt signal is sent. 2: MODE B Both `SUBFRAME.indication` and the subframe interrupt signal are sent from the PHY.		✓		✓	**M**	**O**		**M**	**O**	
Data Report Mode	52	2	The data report mode for the uplink data. 0: A `CRC.indication` message is sent in every subframe. If ULSCH data has been processed, the CRC.indication contains CRC results for the subframe. The CRC results are, also, given in the `RX.indication` message. 1: The `CRC.indication` message is not sent. The CRC results are given in the `RX.indication` message.		✓		✓	**M**	**O**		**M**	**O**	
RACH and SR Report Mode	53	2	The reporting mode for `RACH.indication` and `SR.indcation`. For subframes where a RACH and/or SR opportunity was scheduled: 0: An indication message will always be produced. In the case of RACH, if no preamble was received then the indication will report zero preambles. In the case of SR, if no SR was received then the indication will report zero SR's. 1: A RACH indication will only be generated if one ore more preambles was received. A SR indication will only be generated if one or more SR's was received.		✓		✓	**M**	**O**		**M**	**O**	
These TLVs are used by L1 to report its status													

Description	*Tag*	*Len*	*Value*	*TDD PARAM.RES*		*FDD PARAM.RES*		*TDD CONFIG.REQ*			*FDD CONFIG.REQ*		
				I	C	I	C	I	C	R	I	C	R
PHY State	60	2	Indicates the current operational state of the PHY. 0 = IDLE 1 = CONFIGURED 2 = RUNNING	✓	✓	✓	✓						
			These TLVs are used by L2/L3 to configure internal PHY modules										
Extraction Window Margin	80	2	A margin between the end of the extraction window and the expected time of the last sample of the SC-FDMA symbol. Value: 0 to 63. Default value: 5		✓		✓	M	O	O	M	O	O
PUCCH Noise Estimation Gamma	81	2	This is the forgetting factor used in the IIR filtering for PUCCH noise power estimation. Value: 0 to 65535. The 16-bit value is a 16-bit unsigned number with 16 fractional bits. Default value: 0x2000 (corresponding to 1/8)		✓		✓	M	O	O	M	O	O
PRACH Format 4 Peak Ratio	82	2	The ratio between the threshold for peak detection in PRACH to the average power of the correlator output Value: 0 to 65535. The 16-bit value is a 16-bit unsigned number with 10 fractional bits. Default value: 0x38cc (corresponding to 14.2) for single antenna 0x22cc (corresponding to 8.7) for two antennas		✓			M	O	O			
PRACH Format 0 Peak Ratio	83	2	The ratio between the threshold for peak detection in PRACH to the average power of the correlator output. Value: 0 to 65535. The 16-bit value is a 16-bit unsigned number with 10 fractional bits. Default value: 0x38cc (corresponding to 14.2) for single antenna 0x2266 (corresponding to 8.6) for two antennas		✓		✓	M	O	O	M	O	O

Description	***Tag***	***Len***	***Value***	***TDD PARAM.RES***		***FDD PARAM.RES***		***TDD CONFIG.REQ***			***FDD CONFIG.REQ***		
				I	**C**	**I**	**C**	**I**	**C**	**R**	**I**	**C**	**R**
Doppler Estimation Compensation Factor	84	2	A compensation factor in the Doppler estimation, which tries to account for the effect of the SNR of the channel estimate based on SRS. If the estimates of the signal power and noise power are perfect or accurate, the compensation factor of unity is preferred theoretically. However, in case of environments that these power estimates can hardly be accurate, smaller weighting imposed in the SNR of the channel estimate may improve the performance of Doppler estimation. Value: 0 to 65535. The 16-bit value is a 16-bit unsigned number with 16 fractional bits. Default value: 0xffff (corresponding to 1-(2^-16))		✓		✓	**M**	**O**	**O**	**M**	**O**	**O**
Probability DTX-ACK PUSCH	85	2	The value should be set to the target DTX-to-ACK probability. PHY will use this value to estimate the threshold in ACK-NACK detection algorithm in PUSCH to meet the target DTX-to-ACK probability. See Table 63 for a description of the values. Value: 0 → 31. Default value: 12		✓		✓	**M**	**O**	**O**	**M**	**O**	**O**
Probability DTX-ACK PUCCH Format 1	86	2	The value should be set to the target DTX-to-ACK probability. PHY will use this value to estimate the threshold in ACK-NACK detection algorithm in PUCCH format 1,1a &1b to meet the target DTX-to-ACK probability. See Table 63 for a description of the values. Value: 0 → 31. Default value: 12		✓		✓	**M**	**O**	**O**	**M**	**O**	**O**

Table 9: Configuration TLVs and their applicability

3.2.4 START

The START message exchange was described in Figure 8.

3.2.4.1 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 is, therefore, zero.

3.2.4.2 START Errors

Section removed. Please see section 3.2.6.1.

3.2.5 STOP

The STOP message exchange was described in Figure 9.

3.2.5.1 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 is, therefore, zero.

3.2.5.2 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 is, therefore, zero.

3.2.5.3 STOP Errors

Section removed. Please see section 3.2.6.1.

3.2.6 PHY Notifications

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

3.2.6.1 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 10.

Field	***Size (bytes)***	***Description***
Message ID	1	Indicate which message received by the PHY has an error. Values taken from Table 4.
Error Code	1	The error code, see section 3.2.6.1.1 for more information.
Error code dependent bytes	6	6 bytes of information are provided for each error message. The format of these bytes is dependent on the error code. See Table 12 to Table 20.

Table 10: ERROR`.indication` message body

3.2.6.1.1 Error Codes

The list of possible error codes returned in either `.response` messages, or the `ERROR.indication` message are given in Table 11.

Value	***Error Code***	***Description***
0	MSG_OK	Message is OK.
1	MSG_INVALID_STATE	The received message is not valid in the PHY's current state. See Table 12.
2	MSG_INVALID_CONFIG	The configuration provided in the `CONFIG.request` was invalid and/or incomplete.
3	SFN_OUT_OF_SYNC	The `SUBFRAME.request`, `TX.request`, and/or `HI_DCI.request` was received with different SFN/SF from that expected by the PHY. See Table 13.
4	MSG_SUBFRAME_ERR	An error was received in `SUBFRAME.request`. See Table 14.
5	MSG_BCH_MISSING	A BCH PDU was expected in the `SUBFRAME.request` message for this subframe. However, it was not present. See Table 15.
7	MSG_HI_ERR	An error was received in `HI_DCI.request`. See Table 16.
8	MSG_TX_ERR	An error was received in `TX.request`. See Table 17.
9	MSG_SYNTAX_ERR	The received API message has one or more syntax error. See Table 18. For `SUBFRAME.request`, `HI_DCI.request` and `TX.request` this will be reported if the message has a general syntax error, e.g., a header format error. If the message is generally syntactically correct but has an error in detail, MSG_SUBFRAME_ERR, MSG_HI_ERR or MSG_TX_ERR will be reported as appropriate.
10	MSG_INVALID_TIMING	The received API message starts before and/or completes after the expected window but before the next TTI interrupt. See Table 19.
11	MSG_INVALID_SEQUENCE	The received API messages were in wrong sequence. See Table 20.

Table 11: Error codes

3.2.6.1.2 Error Code Dependent Bytes

For error codes reported in `ERROR.indication` there are six error code dependent bytes that follow. These are detailed in the following tables.

Field	***Size (bytes)***	***Description***
phyState	2	0x29: BLANK 0x30: IDLE 0x31: RUNNING 0x40: CONFIGURED
Reserved	4	Not used.

Table 12: Error code dependent bytes for MSG_INVALID_STATE

Field	*Size (bytes)*	*Description*
Received SFN/SF	2	[15:4]: The system frame number of the received API message [3:0]:The subframe number of the received API message
Expected SFN/SF	2	[15:4]: The system frame number of the PHY [3:0]:The subframe number of the PHY
Reserved	2	Not used.

Table 13: Error code dependent bytes for SFN_OUT_OF_SYNC

Field	*Size (bytes)*	*Description*
Received SFN/SF	2	[15:4]: The system frame number of the received API message [3:0]:The subframe number of the received API message
Error Type	1	Indicates the error type. 0x01: The UL/DL present field conflicts with the SFN/SF of the API message and the subfarme assignment in the TLV. 0x02: The RACH present field conflicts with the TLV. 0x03: Any of the Header fields is out of range. 0x04: Any of the Header fields conflicts with other PDU/TLV fields. 0x05: Any of the PDU fields is out of range. 0x06: Any of the PDU fields conflicts with other PDU/TLV fields.
Direction and PDU Types	1	[7:4]: Indicates if this error was in a DL subframe configuration or an UL subframe configuration. 0 = DL, 1 = UL. [3:0]: The PDU Type parameter specified in both DL subframe configuration and UL subframe configuration. For DL subframe configuration: 0: DCI DL PDU 1: BCH PDU 2: MCH PDU 3: DLSCH PDU 4: PCH PDU For UL subframe configuration: 0: ULSCH PDU 1: ULSCH_CQI_RI PDU 2: ULSCH_HARQ PDU 3: ULSCH_CQI_HARQ_RI PDU 4: UCI_CQI PDU 5: UCI_SR PDU 6: UCI_HARQ PDU 7: UCI_SR_HARQ PDU 8: UCI_CQI_HARQ PDU 9: SRS PDU If the error type is 0x01 or 0x02, this value is set to 0.

Field	*Size (bytes)*	*Description*
PDU index for this type of PDU	2	For example, a subframe request is received with 7 PDUs, 4 of which are DLSCH PDUs. The fifth PDU in the request is the third of the DLSCH PDUs and contains a syntax error. This field will contain 2, the "index" of the faulty PDU in the four DLSCH PDUs. If the error type is 0x01 or 0x02, this value is set to 0. If the error occurred in a MCH or BCH PDU, this value is set to 0.

Table 14: Error code dependent bytes for MSG_SUBFRAME_ERR

Field	*Size (bytes)*	*Description*
Expected SFN/SF	2	[15:4]: The system frame number of the PHY [3:0]:The subframe number of the PHY
Error Type	1	Indicates the error type: 0x01: no SUBFRAME.req is received 0x02: no BCH PDU found in the SUBFRAME.request
Reserved	3	Not used.

Table 15: Error code dependent bytes for MSG_BCH_MISSING

Field	*Size (bytes)*	*Description*
Received SFN/SF	2	[15:4]: The system frame number of the received API message [3:0]:The subframe number of the received API message
Error Type	1	Indicates the error type. 0x03: Any of the Header fields is out of range. 0x04: Any of the Header fields conflicts with other PDU/TLV fields. 0x05: Any of the PDU fields is out of range. 0x06: Any of the PDU fields conflicts with other PDU/TLV fields.
PDU Type	1	The PDU Type parameter specified in both HI DCI subframe configuration If the error type is 0x01 or 0x02, this value is set to 0. 0: HI PDU 1: DCI UL PDU
Position inside the type of PDU	2	For example, a HIDCI request is received with 5 PDUs, 3 of which are HI PDUs. The fifth PDU in the request is the third of the HI PDUs and contains a syntax error. This field will contain 2, the "index" of the faulty PDU in the three HI PDUs. If the error type is 0x01 or 0x02, this value is set to 0.

Table 16: Error code dependent bytes for MSG_HI_ERR

Field	Size (bytes)	Description
Received SFN/SF	2	[15:4]: The system frame number of the received API message [3:0]:The subframe number of the received API message
Error Type	1	Indicates the error type. 0x01: PDU index and length in TX.request is not consistent with SUBFRAME.request. 0x02: The PDU(s) are received too late.
PDU index	1	The first PDU that is not consistent with SUBFRAME.request or is received to late.
reserved	2	Not used

Table 17: Error code dependent bytes for MSG_TX_ERR

Field	Size (bytes)	Description
Received SFN/SF	2	If the received message is SUBFRAME.request, TX.request, or HI_DCI.request: [15:4]: The system frame number of received API message [3:0]:The subframe number of the received API message Otherwise: [15:0]: 0
Error Type	1	Indicates the error type. 0x01: invalid message type 0x02: invalid message length 0x03: invalid SFN/SF 0x04: invalid dlPresent field 0x05: invalid ulPresent field 0x06: invalid SRS present field 0x07: invalid RACH present field 0x08: invalid number of PDU 0x09: invalid PDU type 0x0a: invalid PDU size 0x0b: invalid PDU index
reserved	3	Not used

Table 18: Error code dependent bytes for MSG_SYNTAX_ERR

Field	Size (bytes)	Description
Received SFN/SF	2	[15:4]: The system frame number of the received API message [3:0]:The subframe number of the received API message
Error Type	2	Indicates the error type. 0x01: The time when receiving the first transfer of the API message

Field	*Size (bytes)*	*Description*
		exceeds the expected window 0x02: The time when receiving the last transfer of the API message exceeds the expected window
Starting Samplecnt	2	Samplecnt is an L1 internal counter used to time events. Please contact Picochip support for more information.

Table 19: Error code dependent bytes for MSG_INVALID_TIMING

Field	*Size (bytes)*	*Description*
Received SFN/SF	2	[15:4]: The system frame number of the received API message [3:0]:The subframe number of the received API message
Error Type	1	Indicates the error type. 0x00: SUBFRAME.request, TX.request, and/or HIDCI.request are received for more than once in the same subframe. 0x01: Missing expected API message(s). 0x02: The received API message is detected after unexpected API message(s).
reserved	3	not used

Table 20: Error code dependent bytes for MSG_INVALID_SEQUENCE

3.3 Subframe Messages

The subframe messages are used by the L2/L3 software to control the data transmitted, or received, every 1ms subframe.

3.3.1 SUBFRAME

3.3.1.1 SUBFRAME.indication

The `SUBFRAME.indication` message is given in Table 21. This message is only sent from the PHY if the PHY Sync Mode TLV, sent during the CONFIG message exchange procedure, indicated Mode B.

Field	*Size (bytes)*	*Description*
SFN/SF	2	A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Reserved	2	Align to 32-bit boundary

Table 21: `SUBFRAME.indication` message body

3.3.1.2 SUBFRAME.request

The format of the `SUBFRAME.request` message is shown in Table 22. A `SUBFRAME.request` message indicates the SFN/SF of the subframe for which it contains information. This control information can be for a downlink subframe, an uplink subframe, or both. The details of the downlink and uplink subframe control are described in Sections 3.3.1.2.1 and 3.3.1.2.2.

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	***Size (bytes)***	***Description***
SFN/SF	2	A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Downlink subframe configuration present	1	0 = DL subframe configuration is not present in message 1 = DL subframe configuration is present in message
Uplink subframe configuration present	1	0 = UL subframe configuration is not present in message 1 = UL subframe configuration is present in message
Downlink Subframe configuration	Variable (Always aligned to 32-bits)	See Section 3.3.1.2.1
Uplink Subframe Configuration	Variable (Always aligned to 32-bits)	See Section 3.3.1.2.2

Table 22: `SUBFRAME.request` message body

3.3.1.2.1 Downlink Subframe Configuration

The format of the downlink subframe configuration is given in Table 23.

The following combinations of PDUs are required:

- A BCH PDU does not have an associated DCI PDU
- A PCH PDU requires an associated DCI PDU
- A MCH PDU requires an associated DCI PDU
- A DLSCH allocated with Semi-Persistent Scheduling may not have an associated DCI PDU
- A DLSCH for a unique RNTI requires an associated DCI PDU. Therefore, 2 DLSCH for the same RNTI only require 1 DCI PDU

The PDUs included in this structure have no ordering requirements, except where DLSCH PDUs are provided for two-layer transmission of two transport blocks. In this case, the DLSCH PDU for the first transport block must be followed immediately by the DLSCH PDU for the second.

Field	***Size (bytes)***	***Description***
Length	2	The length of the downlink subframe configuration. Range 0 → 65535.
CFI	1	The number of OFDM symbols for the PDCCH. See [8] section 6.7. Value:0 → 4
Number of DCIs	1	The number of DCI PDUs included in this message. Range: 0 → 255
Number of PDUs	2	Total number of PDUs that are included in this message (DCI, BCH, MCH, DLSCH, PCH). Range 0 → 514

Field	Size (bytes)	Description
Number of PDSCH RNTIs	1	Number of unique RNTIs sent on the PDSCH. - a PCH PDU will have an unique RNTI and should be included in this value - a DLSCH PDU can be one transport block sent to a UE with an unique RNTI. This RNTI should be included in this value - a DLSCH PDU can be one of two transport blocks sent to a UE. In this case the two DLSCH PDUs will share the same RNTI. Only one RNTI should be included in this value.
Reserved	1	Align to 32-bit boundary
For Number of PDUs		
PDU Type	1	0: DCI DL PDU, see Section 3.3.1.2.1.1. 1: BCH PDU, see Section 3.3.1.2.1.2. 2: MCH PDU, see Section 3.3.1.2.1.3. 3: DLSCH PDU, see Section 3.3.1.2.1.4. 4: PCH PDU, see Section 3.3.1.2.1.5.
PDU Size	1	Size of the PDU control information (in bytes). This length value includes the 2 bytes required for the PDU type and PDU size parameters and the two reserved bytes that follow. The size will always be a multiple of 4 bytes.
Reserved	2	Align to 32-bit boundary
DL PDU Configuration	Variable	See Sections 3.3.1.2.1.1 to 3.3.1.2.1.4.

Table 23: Downlink subframe configuration message body

3.3.1.2.1.1 DCI DL PDU

The format of a DCI DL PDU is shown in Table 24. The DCI DL PDU contains the information which the L2/L3 software must provide the PHY so it can create the DCI formats, related to the downlink, described in [9] section 5.3.3.1.

Field	Size (bytes)	Description
DCI Format	1	Format of the DCI 0 = 1 1 = 1A 2 = 1B 3 = 1C 4 = 1D 5 = 2 6 = 2A
CCE Index	1	CCE index used to send the DCI. Value: 0 → 88
Aggregation Level	1	The aggregation level used Value: 1,2,4,8

Field	***Size (bytes)***	***Description***
Reserved	1	Align to 32-bit boundary
RNTI	2	The RNTI used for the receiving UE Valid for all DCI formats Value: 0 → 65535.
Resource Allocation Type	1	Resource allocation type Valid for DCI formats: 1,2,2A 0=type 0 1=type 1
Virtual resource block assignment flag	1	Type of virtual resource block used Valid for DCI formats: 1A,1B,1D 0 = localized 1 = distributed
Resource Block Coding	4	The encoding for the resource blocks. Its coding is dependent on whether resource allocation type 0,1,2 is in use. Resource allocation type 0 is explicitly signalled for DCI formats 1, 2, 2A Resource allocation type 1 is explicitly signalled for DCI formats 1, 2, 2A Resource allocation type 2 is implicit for DCI formats 1A, 1B, 1C, 1D See [6] section 7.1.6 for the encoding used for each format. Valid for DCI formats: 1,1A,1B,1C,1D ,2,2A
MCS [1]	1	The modulation and coding scheme for 1st transport block Valid for DCI formats: 1,1A,1B,1C,1D ,2,2A Value: 0 → 31
Redundancy Version [1]	1	The redundancy version for 1st transport block. Valid for DCI formats: 1,1A,1B,1C,1D ,2,2A Value: 0 → 3
New Data Indicator[1]	1	The new data indicator for 1st transport block. Valid for DCI formats: 1,1A,1B,1C,1D ,2,2A NB: for format 1A, this field is invalid if CRC is not scrambled by RA-RNTI, P-RNTI or SI-RNTI Value: 0..1, If the value is the same as previous value, then it is defined as Not toggled, otherwise it is defined as toggled. Toggled: new data, Not toggled: repeated data.
Transport block to codeword swap flag	1	Indicates the mapping of transport block to codewords Valid for DCI formats: 2,2A 0 = no swapping 1 = swapped
MCS[2]	1	The modulation and coding scheme for 2nd transport block. Valid for DCI formats: 2,2A Value: 0 → 31

Field	***Size (bytes)***	***Description***
Redundancy Version [2]	1	The redundancy version for 2nd transport block. Valid for DCI formats: 2,2A Value: 0 → 3
New Data Indicator[2]	1	The new data indicator for 2nd transport block. Valid for DCI formats: 2,2A Value: 0..1, If the value is the same as previous value, then it is defined as Not toggled, otherwise it is defined as toggled. Toggled: new data, Not toggled: repeated data.
HARQ Process	1	HARQ process number Valid for DCI formats: 1,1A,1B,1D,2,2A NB: for format 1A, this field is invalid if CRC is scrambled by RA-RNTI, P-RNTI or SI-RNTI Value: 0 → 15 See [6] for more detail on the correct number of HARQ processes for each configuration.
TPMI	1	The codebook index to be used for precoding Valid for DCI formats: 1B,1D 2 antenna_ports: 0 → 3 4 antenna_ports: 0 → 15
PMI	1	Confirmation for precoding Valid for DCI formats: 1B 0 = use precoding indicated in TPMI field 1 = use precoding indicated in last PMI report on PUSCH
Precoding Information	1	Precoding information Valid for DCI formats: 2,2A 2 antenna_ports: 0 → 7 4 antenna_ports: 0 → 63
TPC	1	Tx power control command for PUCCH. Valid for DCI formats: 1,1A,1B,1D,2,2A Value 0-3 If the format 1A CRC is scrambled by RA-RNTI, P-RNTI, or SI-RNTI: Bit 1 of the TPC command is reserved. Bit 0 of the TPC command indicates column N_{PRB}^{1A} of the TBS table defined in [6]; If bit 0 is 0 then N_{PRB}^{1A} =2 else N_{PRB}^{1A} =3.
Downlink Assignment Index	1	The downlink assignment index. Valid only for TDD uplink-downlink configurations 1-6. Valid for DCI formats: 1,1A,1B,1D,2,2A For format 1A, this field is invalid if CRC is scrambled by RA-RNTI, P-RNTI or SI-RNTI. Value: 1,2,3,4

Field	***Size (bytes)***	***Description***
N_{GAP}	1	Indicates which gap value to use for distributed virtual resource blocks. Valid for DCI formats:1A,1B,1C,1D For DCI format 1A, this field is valid only if CRC is scrambled by RA-RNTI, P-RNTI or SI-RNTI and number of DL resource blocks is 50 or more. 0= N_{GAP1} 1= N_{GAP2}
Transport block size index	1	The transport block size Valid for DCI formats: 1C Value: 0 → 31
Downlink power offset	1	Indicates the DL power offset type for multi-user MIMO transmission Valid for DCI formats: 1D Value: 0 → 1
Allocate PRACH flag	1	Indicates that PRACH procedure is initiated Valid for DCI formats: 1A 0 = false 1= true
Preamble Index	1	The preamble index to be used on the PRACH Valid for DCI formats: 1A Value: 0 → 63
PRACH Mask Index	1	The mask index to be used on the PRACH Valid for DCI formats: 1A Value: 0 → 15
Scramble type	1	Scramble type. See Section 5.3.3.2 in [9]. Valid for DCI format 1A. For Random Access Procedure initiated by PDCCH order, this field will be 1 (C-RNTI) and the Allocate PRACH flag (see above) will be TRUE. Value: 1 = CRC scrambled with C-RNTI 2 = CRC scrambled by RA-RNTI, P-RNTI, or SI-RNTI
TRAILING RESERVED BYTES HAVE BEEN REMOVED SINCE VERSION 1.11		

Table 24: DCI DL PDU

3.3.1.2.1.2 BCH PDU

The format of the BCH PDU is shown in Table 25.

Field	***Size (bytes)***	***Description***
Length	2	The length (in bytes) of the associated MAC PDU, delivered in `TX.request`. This should be the actual length of the MAC PDU, which may not be a multiple of 32-bits.

Field	*Size (bytes)*	*Description*
PDU index	2	This is a count value which is incremented following the inclusion of a BCH, MCH, PCH or DLSCH PDU in the `SUBFRAME.request` message. This value is repeated in `TX.request` and associates the control information to the data. It is reset to 0 at the beginning of each downlink configuration Range 0 → 65535
Transmission power	2	Offset to the reference signal power. Value: 0 → 10000, representing -6 dB to 4 dB in 0.001 dB steps.
Reserved	2	Align to 32-bit boundary

Table 25: BCH PDU

3.3.1.2.1.3 MCH PDU

The format of the MCH PDU is shown in Table 26. The contents of the MCH PDU are preliminary.

Field	*Size (bytes)*	*Description*
Length	2	The length (in bytes) of the associated MAC PDU, delivered in `TX.request`. This should be the actual length of the MAC PDU, which may not be a multiple of 32-bits
PDU index	2	This is a count value which is incremented following the inclusion of a BCH, MCH, PCH or DLSCH PDU in the `SUBFRAME.request` message. This value is repeated in `TX.request` and associates the control information to the data. It is reset to 0 at the beginning of each downlink configuration Range 0 → 65535
RNTI	2	The RNTI associated with the MCH See [6] section 7.1. Value: 0 → 65535.
Resource Allocation Type	1	Resource allocation type See [6] section 7.1.6 0 = type 0 1 = type 1 2 = type 2
Reserved	1	Align to 32-bit boundary
Resource Block Coding	4	The encoding for the resource blocks. Its coding is dependent on whether resource allocation type 0,1,2 is in use. See [6] section 7.1.6 for the encoding used for each format.
Modulation Type	1	The modulation type used in the transport channel 0: QPSK 1: QAM16 2: QAM64

Field	Size (bytes)	Description
Reserved	1	Align to 16-bit boundary
Transmission power	2	Offset to the reference signal power. Value: 0 → 10000, representing -6 dB to 4 dB in 0.001 dB steps.

Table 26: MCH PDU

3.3.1.2.1.4 DLSCH PDU

The format of the DLSCH PDU is shown in Table 27.

Field	Size (bytes)	Description
Length	2	The length (in bytes) of the associated MAC PDU, delivered in `TX.request`. This should be the actual length of the MAC PDU, which may not be a multiple of 32-bits. It should also match the transport block size sent in the corresponding DL DCI since layer 2 is responsible for PDU padding.
PDU index	2	This is a count value which is incremented following the inclusion of a BCH, MCH, PCH or DLSCH PDU in the `SUBFRAME.request` message. This value is repeated in `TX.request` and associates the control information to the data. It is reset to 0 at the beginning of each downlink configuration Range 0 → 65535
RNTI	2	The RNTI associated with the UE See [3] section 5.1.4 Value: 0 → 65535.
Resource Allocation Type	1	Resource allocation type See [3] section 7.1.6 0 = type 0 1 = type 1 2 = type 2 (allocated by format 1A or 1B or 1D DCI) 3 = type 2 (allocated by format 1C DCI)
Virtual resource block assignment flag	1	Type of virtual resource block used. This should match the value sent in the DCI Format 1A, 1B, 1D PDU which allocated this grant. See [6] section 7.1.6.3 0 = localized 1 = distributed
Resource Block Coding	4	The encoding for the resource blocks. Its coding is dependent on whether resource allocation type 0,1,2 is in use. This should match the value sent in the DCI Format PDU which allocated this grant. See [6] section 7.1.6 for the encoding used for each format.

Field	Size (bytes)	Description
N_{GAP}	1	Indicates which gap value to use for distributed virtual resource blocks. Valid for DCI formats:1A,1B,1C,1D For DCI format 1A, this field is valid only if CRC is scrambled by RA-RNTI, P-RNTI or SI-RNTI and number of DL resource blocks is 50 or more. 0= N_{GAP1} 1= N_{GAP2}
MCS	1	0: QPSK 1: 16QAM 2: 64QAM
Redundancy Version	1	HARQ redundancy version. This should match the value sent in the DCI PDU which allocated this grant. Value: 0 → 3.
Number Of Transport Blocks	1	The number of transport blocks transmitted to this RNTI. Value: 1 → 2
Reserved	1	This was previously the transport block to codeword swap flag; however, this feature is not supported by the L1 implementation and therefore this field is reserved.
Transmission Scheme	1	The transmission scheme used in this DLSCH See [6] section 7.1. 0: SINGLE_ANTENNA_PORT_0, 1: TX_DIVERSITY, 2: LARGE_DELAY_CDD, 3: CLOSED_LOOP_SPATIAL_MULTIPLEXING, 4: MULTI_USER_MIMO, 5: CLOSED_LOOP_RANK_1_PRECODING, 6: SINGLE_ANTENNA_PORT_5.
Number Of Layers	1	The number of layers used in this transmission See [8] section 6.3.3 Value: 1 → 4
Codebook Index	1	Defines the codebook used. Only valid when transmission scheme = 3, 4, or 5. When antenna port = 1: NA When antenna port = 2: 0..3, When antenna port = 4: 0..15.
UE Category Capacity	1	The UE capabilities category See [10] section 4.1. Value:1 → 5

Field	Size (bytes)	Description
P-A	1	The ratio of PDSCH EPRE to cell-specific RS EPRE among PDSCH REs in all the OFDM symbols not containing cell-specific RS in dB. See [6], section 5.2. 0: -6dB 1: -4.77dB 2: -3dB 3: -1.77dB 4: 0dB 5: 1dB 6: 2dB 7: 3dB
Delta power offset index	1	Delta power offset, value: 0..1, Refer to: Table 7.1.5-1 in [6] for Multi-user MIMO mode. Takes value zero for all other modes.
Reserved	1	

Table 27: DLSCH PDU

3.3.1.2.1.5 PCH PDU

The format of the PCH PDU is shown in Table 28.

Field	Size (bytes)	Description
Length	2	The length (in bytes) of the associated MAC PDU, delivered in `TX.request`. This should be the actual length of the MAC PDU, which may not be a multiple of 32-bits. It should also match the transport block size sent in the corresponding DL DCI since layer 2 is responsible for PDU padding.
PDU index	2	This is a count value which is incremented following the inclusion of a BCH, MCH, PCH or DLSCH PDU in the `SUBFRAME.request` message. This value is repeated in `TX.request` and associates the control information to the data. It is reset to 0 at the beginning of each downlink configuration Range 0 → 65535
P-RNTI	2	The P-RNTI associated with the paging See [6] section 7.1. Value: 0xFFFE
Resource Allocation Type	1	Resource allocation type See [6] section 7.1.6 2 = type 2 (allocated by format 1A) 3 = type 2 (allocated by format 1C)

Field	***Size (bytes)***	***Description***
Virtual resource block assignment flag	1	Type of virtual resource block used. This should match the value sent in the DCI PDU which allocated this grant. See [6] section 7.1.6.3 0 = localized 1 = distributed
Resource Block Coding	4	The encoding for the resource blocks. Its coding is dependent on whether resource allocation type 0,1,2 is in use. This should match the value sent in the DCI PDU which allocated this grant. See [6] section 7.1.6 for the encoding used for each format.
N_{GAP}	1	Indicates which gap value to use for distributed virtual resource blocks. Valid for DCI formats:1A, 1C For DCI format 1A, this field is valid only if the number of DL resource blocks is 50 or more. 0= N_{GAP1} 1= N_{GAP2}
MCS	1	For PCH PDU only QPSK modulation is allowed. 0: QPSK
Redundancy Version	1	For PCH PDU only redundancy version 0 is allowed Value: 0
Number Of Transport Blocks	1	The number of transport blocks transmitted to this RNTI. Only 1 transport block is sent on the PCH per subframe Value: 1
Reserved	1	(Transport block to codeword swap flag is not used on PCH.)
Transmission Scheme	1	The transmission scheme to be used. See [6] section 7.1. 0: SINGLE_ANTENNA_PORT_0, 1: TX_DIVERSITY, 6: SINGLE_ANTENNA_PORT_5.
Number Of Layers	1	The number of layers used in transmission See [8] section 6.3.3 Value: 1 → 4
Reserved	1	(Codebook Index is not used on PCH.)
Reserved	1	(UE Category Capacity Is not used on PCH.)

Field	Size (bytes)	Description
P-A	1	The ratio of PDSCH EPRE to cell-specific RS EPRE among PDSCH REs in all the OFDM symbols not containing cell-specific RS in dB. See [6], section 5.2. 0: -6dB 1: -4.77dB 2: -3dB 3: -1.77dB 4: 0dB 5: 1dB 6: 2dB 7: 3dB
Reserved	1	Align to 16-bit boundary
Reserved	1	(Was previously transmission power offset which does not apply to PCH)

Table 28: PCH PDU

3.3.1.2.2 Uplink Subframe Configuration

The format of the uplink subframe configuration is given in Table 29.

The following combinations of PDUs are required:

- In order to support RACH in the subframe, the RACH present field must be true
- In order to support SRS in the subframe, the SRS present field must be true.
- If the SRS present field is true, there can be zero, one or more SRS PDUs present.
- The ULSCH PDU is used when a UE has been instructed to only send uplink data.
- The ULSCH_CQI_RI, ULSCH_HARQ and ULSCH_CQI_HARQ_RI PDUs are present when a UE has been instructed to send uplink data and control.
- The UCI_CQI, UCI_SR, UCI_SR_HARQ and UCI_CQI_HARQ PDUs are present when a UE has been only been instructed to transmit control
- The following combinations can have the same RNTI values:
  - UCI_HARQ + SRS
  - ULSCH_x + SRS

Field	Size (bytes)	Description
Length	2	The length of the uplink subframe configuration. Range 0 → 65535.
Number of PDUs	1	The number of PDUs included in this message.
RACH present	1	0: No RACH in this subframe 1: RACH present in this subframe
SRS present	1	0: No cell-specific SRS in this subframe 1: Cell-specific SRS present in this subframe

Field	*Size (bytes)*	*Description*
Reserved	3	Aligned to 32-bits boundary
For Number of PDUs		
PDU Type	1	0: ULSCH, see Section 3.3.1.2.2.1. 1: ULSCH_CQI_RI, see Section 3.3.1.2.2.2. 2: ULSCH_HARQ, see Section 3.3.1.2.2.3. 3: ULSCH_CQI_HARQ_RI, see Section 3.3.1.2.2.4. 4: UCI_CQI, see Section 3.3.1.2.2.5. 5: UCI_SR, see Section 3.3.1.2.2.6. 6: UCI_HARQ, see Section 3.3.1.2.2.7. 7: UCI_SR_HARQ, see Section 3.3.1.2.2.8. 8: UCI_CQI_HARQ, see Section 3.3.1.2.2.9. 9: SRS, see Section 3.3.1.2.2.10. Note : ULSCH_CQI_RI represents two types : ULSCH_CQI and UL_RI, ULSCH_CQI_HARQ_RI represents two types : ULSCH_CQI_HARQ and ULSC_RI_HARQ.
PDU Size	1	Size of the PDU control information (in bytes). This length value includes the 2 bytes required for the PDU type and PDU size parameters and the two reserved bytes that follow. The length will always be a multiple of 4 bytes.
Reserved	2	Align to 32-bit boundary
UL PDU Configuration	Variable	See Sections 3.3.1.2.2.1 to 3.3.1.2.2.10.

Table 29: Uplink subframe configuration message body

3.3.1.2.2.1 ULSCH PDU

The format of the ULSCH PDU is given in Table 30.

Field	*Size (bytes)*	*Description*
Size	2	The size of the ULSCH PDU in bytes as defined by the relevant UL grant. The size can be 0 if UCI over ULSCH without data is configured. The sizes of CQI/RI/HARQ are not added to this element.
Reserved	2	Align to 32-bit boundary
RNTI	2	The RNTI used for identifying the UE when receiving the PDU See [6] section 8. Value: 0 → 65535.
Resource Block Start	1	The starting resource block for this ULSCH allocation. This should match the value sent in the DCI Format 0 PDU which allocated this grant. Value: 0 → 99

Field	Size (bytes)	Description
Number of Resource Blocks	1	The number of resource blocks allocated to this ULSCH grant. This should match the value sent in the DCI Format 0 PDU which allocated this grant. Value: 1 → 100
Modulation Type	1	0: QPSK 1: 16QAM 2: 64QAM
Cyclic Shift 2 for DMRS	1	The 2nd cyclic shift for DMRS assigned to the UE in the ULSCH grant. This should match the value sent in the DCI Format 0 PDU which allocated this grant. The mapping of this field to a value of $n_{DMRS}^{(2)}$ is defined in [8]. Value: 0 → 7
Frequency enabled flag	1	Indicates if hopping is being used. This should match the value sent in the DCI Format 0 PDU which allocated this grant. See [6] Section 8.4. 0 = no hopping, 1= hopping enabled
Frequency hopping bits	1	The frequency hopping bits. This should match the value sent in the DCI Format 0 PDU which allocated this grant. See [6] Section 8.4 Value: 0 → 3
New Data Indicator	1	Specify whether this received PUSCH is a new transmission from UE. This indicator controls the L1 HARQ combining process. Value: 0, 1 If the value has not toggled, data is a retransmission If the value has toggled, data is a new transmission
Redundancy Version	1	Redundancy version Value: 0 → 3
HARQ Process Number	1	HARQ Process number. TDD 0 → 15 FDD 0 → 7
UL Tx Mode	1	0 = SISO/SIMO 1 = MIMO
Shorten Flag	1	ULSCH should be shortened in following cases: (1) ULSCH conflicts with cell-specific SRS transmission. (2) ULSCH and SRS are transmitted in the same subframe for the same UE. 0 = ULSCH is normal 1 = ULSCH is shortened
Reserved	3	

Table 30: ULSCH PDU

3.3.1.2.2.2 ULSCH_CQI_RI PDU

The format of the ULSCH_CQI_RI PDU is given in Table 31.

Field	Size (bytes)	Description
ULSCH PDU	20	Description of contents given in Table 30
CQI_RI information	4	Description of contents given in Table 32

Table 31: ULSCH_CQI_RI PDU

The format of the CQI_RI information is given in Table 32.

Field	Size (bytes)	Description
DL CQI/PMI Size	1	The size of the DL CQI/PMI in bits. Value: 0 → 255, 0 means that there is no CQI/PMI.
RI Size	1	The size of RI in bits Value:0 → 2, 0 means that there is no RI.
Delta Offset CQI	1	Delta offset for CQI. This value is fixed for a UE and allocated in RRC connection setup. Valid when DLCQI/PMI size is non-zero, otherwise reserved. See [6] section 8.6.3 Value: 0 → 15
Delta Offset RI	1	Delta offset for RI. This value is fixed for a UE and allocated in RRC connection setup. Valid when RI Size is non-zero, otherwise reserved. See [6] section 8.6.3 Value: 0 → 15

Table 32: CQI_RI Information

3.3.1.2.2.3 ULSCH_HARQ PDU

The format of the ULSCH_HARQ PDU is given in Table 33.

Field	Size (bytes)	Description
ULSCH PDU	20	Description of contents given in Table 30
HARQ information	4	Description of contents given in Table 34

Table 33: ULSCH_HARQ PDU

The format of the HARQ information is given in Table 34.

Field	Size (bytes)	Description
HARQ Size	1	For ACK/NACK bundling and ACK/NACK multiplexing of one subframe (M=1 in [6] section 10.1), this is the number of spatial layers being reported Value: 1 → 2 Otherwise for ACK/NACK multiplexing this is the number of multiplexed subframes (M) Value: 1 → 4
Delta Offset HARQ	1	Delta offset for HARQ. This value is fixed for a UE and allocated in RRC connection setup. See [6] section 8.6.3 Value: 0 → 15
ACK_NACK mode	1	The format of the ACK/NACK response expected. Valid for TDD only. 0 = BUNDLING 1 = MULTIPLEXING
Reserved	1	Align to 32-bit boundary

Table 34: HARQ Information

3.3.1.2.2.4 ULSCH_CQI_HARQ_RI PDU

The format of the ULSCH_CQI_HARQ_RI PDU is given in Table 35.

Field	Size (bytes)	Description
ULSCH PDU	20	Description of contents given in Table 30
CQI_RI information	4	Description of contents given in Table 32
HARQ information	4	Description of contents given in Table 34

Table 35: ULSCH_CQI_HARQ_RI PDU

3.3.1.2.2.5 UCI_CQI PDU

The format of the UCI_CQI PDU is given in Table 35.

Field	Size (bytes)	Description
Common UCI information	8	Description of contents given in Table 37. For this PDU the PUCCH Index Type must be 1 (Type 2/2a/2b).
CQI Information	4	Description of contents given in Table 38

Table 36: UCI_CQI PDU

The common information for UCI_x PDUs is given in Table 37.

Field	Size (bytes)	Description
RNTI	2	The RNTI used for identifying the UE when receiving the PDU See [6] section 8 Value: 0 → 65535.
PUCCH Index Type	1	The type of PUCCH Index 0 = PUCCH Index Type 1/1a/1b, see [8] section 5.4.1 1 = PUCCH Index Type 2/2a/2b, see [8] section 5.4.1
Reserved	1	Align to 32 bits
PUCCH index	2	For PUCCH Index Type = 0 (i.e. 1/1a/1b), The PUCCH Index value $n_{\mathrm{PUCCH}}^{(1)}$ For PUCCH Index Type = 1 (i.e. 2/2a/2b), The PUCCH Index value $n_{PUCCH}^{(2)}$ Value: 0 → 2047
Shorten flag	1	The length of PUCCH should be shortened due to the concurrent transmission of PUCCH and SRS. 0 = PUCCH is normal 1 = PUCCH is shortened
Reserved	1	Align to 32 bits

Table 37: Common UCI information

The CQI information is given in Table 38.

Field	Size (bytes)	Description
DL CQI/PMI Size	1	The size of the DL CQI/PMI in bits. Value: 0 → 255
Reserved	3	Align to 32-bit boundary

Table 38: CQI information

3.3.1.2.2.6 UCI_SR PDU

The format of the UCI_SR PDU is given in Table 39.

Field	Size (bytes)	Description
Common UCI information	8	Description of contents given in Table 37. For this PDU the PUCCH Index Type must be 0 (type 1/1a/1b).

Table 39: UCI_SR PDU

3.3.1.2.2.7 UCI_HARQ PDU

The format of the UCI_HARQ PDU is given in Table 40.

Field	***Size (bytes)***	***Description***
Common UCI information	8	Description of contents given in Table 37. For this PDU the PUCCH Index Type must be 0 (type 1/1a/1b).
HARQ Information	12 for TDD 4 for FDD	Description of contents given in Table 41 for TDD and Table 42 for FDD.

Table 40: UCI_HARQ PDU

The HARQ information is given in Table 41 and Table 42. Note that for UCI_CQI_HARQ and UCI_SR_HARQ PDUs in TDD mode, the data is in format described here as "Special Bundling" (see [6]).

Field	***Size (bytes)***	***Description***
HARQ Size	1	For ACK/NACK bundling and ACK/NACK multiplexing of one subframe (M=1 in [6] section 10.1), this is the number of spatial layers being reported Value: 1 → 2 Otherwise for ACK/NACK multiplexing this is the number of multiplexed subframes (M) Value: 1 → 4 In the case of SPECIAL BUNDLING for UCI_CQI_HARQ and UCI_SR_HARQ PDUs (see section 3.3.3.2.1) this field is reserved.
ACK_NACK mode	1	The format of the ACK/NACK response expected. 0 = BUNDLING 1 = MULTIPLEXING
Number of PUCCH Resources	1	For ACK/NACK bundling and ACK/NACK multiplexing of one subframe (M=1 in [6] section 10.1), one PUCCH resource is used. Otherwise for ACK/NACK multiplexing the number of resources is equal to the number of subframes being reported (M) A value of 0 is valid for UCI_CQI_HARQ PDU. Value: 0 → 4
Reserved	1	
n_PUCCH_1_0	2	HARQ resource 0, value: 0 → 2047 If this HARQ resource is unused, this value should be set to 0xFFFF.
n_PUCCH_1_1	2	HARQ resource 1, value: 0 → 2047 If this HARQ resource is unused, this value should be set to 0xFFFF.
n_PUCCH_1_2	2	HARQ resource 2, value: 0 → 2047 If this HARQ resource is unused, this value should be set to 0xFFFF.
n_PUCCH_1_3	2	HARQ resource 3, value: 0 → 2047 If this HARQ resource is unused, this value should be set to 0xFFFF.

Table 41: UCI HARQ information for TDD

Field	Size (bytes)	Description
HARQ Size	1	The size of the ACK/NACK in bits. Value: 1 → 2.
Reserved	1	
n_PUCCH_1	2	The PUCCH Index value $n_{\mathrm{PUCCH}}^{(1)}$ Value for type 0: 0 → 2047

Table 42: UCI HARQ information for FDD

3.3.1.2.2.8 UCI_SR_HARQ PDU

The format of the UCI_SR_HARQ PDU is given in Table 43.

Field	Size (bytes)	Description
Common UCI information	8	Description of contents given in Table 37. For this PDU the PUCCH Index Type must be 0 (type 1/1a/1b). “PUCCH resource index for SR” should be stored in the corresponding "PUCCH Index" field in this "Common UCI information” field.
HARQ Information	12 for TDD 4 for FDD	Description of contents given in Table 41 for TDD and Table 42 for FDD. "PUCCH resource index for HARQ” should be stored in the corresponding “n_PUCCH_1_X" field in this "HARQ information" field.

Table 43: UCI_SR_HARQ PDU

3.3.1.2.2.9 UCI_CQI_HARQ PDU

The format of the UCI_CQI_HARQ PDU is given in Table 44.

For TDD when both HARQ and CQI or SR are transmitted on PUCCH, multiple HARQ ACK/NACK responses are bundled according to table 7.3-1 of [6]. This is referred to in Table 41 as “Special Bundling” and implies a unique interpretation of the message fields.

Field	Size (bytes)	Description
Common UCI information	8	Description of contents is given in Table 37. For this PDU the PUCCH Index Type must be 1 (type 2/2a/2b). The PUCCH resource for format 2a or 2b should be stored in the corresponding field in this “Common UCI information” field.
CQI Information	4	Description of contents given in Table 38
HARQ Information	12 for TDD 4 for FDD	Description of contents given in Table 41 for TDD and Table 42 for FDD. "Number of PUCCH resource" field in this "HARQ information" field should be zero. "n_PUCCH_1_X" field in this "HARQ information" field should be zeros.

Table 44: UCI_CQI_HARQ PDU

3.3.1.2.2.10 SRS

The format of the SRS PDU is given in Table 45.

Field	***Size (bytes)***	***Description***
RNTI	2	The RNTI used for identifying the UE when receiving the PDU See [6] section 8. Value: 0 → 65535.
SRS Bandwidth	1	SRS Bandwidth. This value is fixed for a UE and allocated in RRC connection setup. See [8] section 5.5.3.2 Value: 0 → 3
Frequency Domain Position	1	Frequency-domain position, N_{RRC} This value is fixed for a UE and allocated in RRC connection setup. See [8] section 5.5.3.2 Value: 0 → 23
SRS Hopping Bandwidth	1	Configures the frequency hopping on the SRS. This value is fixed for a UE and allocated in RRC connection setup. See [8] section 5.5.3.2. Value 0 → 3
Transmission Comb	1	Configures the frequency location of the SRS. This value is fixed for a UE and allocated in RRC connection setup. Value: 0 → 1
I_{SRS} / SRS-ConfigIndex	2	Defines the periodicity and subframe location of the SRS. SRS Configuration Index. This value is fixed for a UE and allocated in RRC connection setup. See [6] section 8.2. Value: 0 → 1023.
Sounding Reference Cyclic Shift	1	Configures the SRS sequence generation. This value is fixed for a UE and allocated in RRC connection setup. See [8] section 5.5.3.1. Value: 0 → 7
User index for Doppler estimation	1	The user index defines the memory location of the previous SRS channel estimates for computing the Doppler estimate. Value: 0-31 for TDD and 0-79 for FDD
New User Indicator.	1	This value should be toggled when a new user is assigned to this SRS user index. Value: 0 → 1
Reserved	1	Align to 32-bit boundary

Table 45: SRS PDU

3.3.1.3 HI_DCI.request

The format of the `HI_DCI.request` message is given in Table 46. This message contains two types of control information relating to the uplink. Firstly, it is used for the L2/L3 to send the ACK/NACK response for MAC PDUs received on the ULSCH; LTE has strict timing requirements for returning this information to the UE. Secondly, it is used for DCI control format information relating to the uplink which is broadcast on the PDCCH.

Section 2.2.3 contains a detailed description on when this message should be sent, and the correct value of SFN/SF.

Field		Size (bytes)	Description
SFN/SF		2	The SFN/SF in this message should be the same as the corresponding `SUBFRAME.request` message. A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Number of DCI		1	Number of DCI PDUs included in this message
Number of HI		1	Number of HI PDUs included in this message
For Number of DCI + HI PDUs			
	PDU Type	1	0: HI PDU, see Section 3.3.1.3.1. 1: DCI UL PDU, see Section 3.3.1.3.2.
	PDU Size	1	Size of the PDU control information (in bytes). This length value includes the 2 bytes required for the PDU type and PDU size parameters and the two reserved bytes that follow. The size will always be a multiple of 4 bytes.
	Reserved	2	Align to 32-bit boundary
	HI/DCI PDU Configuration	Variable	See Sections 3.3.1.3.1 to 3.3.1.3.2.

Table 46: `HI_DCI.request` message body

3.3.1.3.1 HI PDU

The format of a HI PDU is shown in Table 47. The HI PDU contains the ACK/NACK response for a ULSCH transmission.

Field	Size (bytes)	Description
Resource Block Start	1	This value is the starting resource block assigned to the ULSCH grant associated with this HI response. It should match the value sent in the DCI format 0 which allocated the ULSCH grant See [6] section 9.1.2 Value: 0 → 99
Cyclic Shift 2 for DMRS	1	This value is the 2nd cyclic shift for DMRS assigned to the ULSCH grant associated with this HI response. It should match the value sent in the DCI format 0 which allocated the ULSCH grant See [6] section 9.1.2 Value: 0 → 7
HI Value	1	The PHICH value which is sent on the resource. 0: HI_NACK 1: HI_ACK

Field	Size (bytes)	Description
I_PHICH	1	Is used in the calculation of the PHICH location. For TDD only. See [6] section 9.1.2 1 = TDD subframe configuration 0 is used and the ULSCH grant associated with this HI was received in subframe 4 or 9 0 = in all other cases
Transmission power	2	Offset to the reference signal power. Value: 0 → 10000, representing -6 dB to 4 dB in 0.001 dB steps.
Reserved	2	Align to 32-bit boundary

Table 47: HI PDU

3.3.1.3.2 DCI UL PDU

The format of a DCI UL PDU is shown in Table 48. The DCI UL PDU contains the information which the L2/L3 software must provide the PHY so it can create the DCI format 0 or format 3/3A described in [9] section 5.3.3.1.

Field	Size (bytes)	Description
DCI Format	1	Format of the DCI 0 = 0 1 = 3 2 = 3A
CCE Index	1	CCE index used to send the DCI. Value: 0 → 88
Aggregation Level	1	The aggregation level used Value: 1,2,4,8
Reserved	1	Align to 32-bit boundary
RNTI	2	The RNTI used for identifying the UE when receiving the PDU Valid for all DCI formats Value: 0 → 65535.
Resource Block Start	1	The starting resource block for this ULSCH allocation. Valid for DCI format 0 Value: 0 → 99 (A value of 255 is valid for SPS release, see section 3.3.1.3.2.1.)
Number of Resource Blocks	1	The number of resource blocks allocated to this ULSCH grant. Valid for DCI format 0 (see also section 3.3.1.3.2.1). If resource block start is 255, this field is reserved. Value: 1 → 100
MCS	1	The modulation and redundancy version. See [6] section 8.6. Valid for DCI format 0 Value: 0 → 31

Field	*Size (bytes)*	*Description*
Cyclic Shift 2 for DMRS	1	The 2nd cyclic shift for DMRS assigned to the UE in the ULSCH grant. Valid for DCI format 0 Value: 0 → 7
Frequency enabled flag	1	Indicates if hopping is being used. See [6] Section 8.4. Valid for DCI format 0 0 = no hopping, 1= hopping enabled
Frequency hopping bits	1	The frequency hopping bits See [6] Section 8.4 Valid for DCI format 0 Value: 0 → 3
New Data Indication	1	Specify whether this received PUSCH is a new transmission from UE. Valid for DCI format 0 Value: 0..1, If the value is the same as previous value, then it is defined as Not toggled, otherwise it is defined as Toggled. Toggled: new data Not toggled: repeated data.
UE TX antenna selection	1	Indicates how the CRC is calculated on the PDCCH. See [9] section 5.3.2.2 Valid for DCI format 0 0 = Not configured; 1 = Configured and using UE port 0; 2 = Configured and using UE port 1.
TPC	1	Tx power control command for PUSCH. Valid for DCI format 0 Value: 0..3.
CQI request	1	Aperiodic CQI request flag Valid for DCI format 0 0 = Aperiodic CQI not requested 1 = Aperiodic CQI requested
UL index	1	UL index. This field is valid only for DCI format 0 in TDD operation with uplink-downlink configuration 0. Value: 0,1,2,3
DL assignment index	1	DL assignment index. Valid only for DCI format 0 in TDD uplink-downlink configurations 1-6. Value: 1,2,3,4
Reserved	2	Align to 32-bit boundary

Field	Size (bytes)	Description

TPC bitmap	4	TPC commands for PUCCH and PUSCH
		Valid for DCI formats: 3,3A
		The encoding follows [9] section 5.3.3.1.6
		
		For format 3, the encoding is as follows:
		
		`Bit      7 6   5 4   3 2   1 0`
		`Byte  0 TPC0  TPC1  TPC2  TPC3`
		`Byte  1 TPC4  TPC5  TPC6  TPC7`
		`Byte  2 TPC8   00    00    00`
		`Byte  3  00    00    00    00`
		
		For format 3A, the encoding is as follows:
		
		`Bit      7     6     5     4     3     2     1     0`
		`Byte  0 TPC0  TPC1  TPC2  TPC3  TPC4  TPC5  TPC6  TPC7`
		`Byte  1 TPC8  0     0     0     0     0     0     0`
		`Byte  2  0    0     0     0     0     0     0     0`
		`Byte  3  0    0     0     0     0     0     0     0`

Table 48: DCI UL PDU

3.3.1.3.2.1 Special fields for semi-persistent scheduling release

Section 9.2 of [6] defines specific values of DCI fields relating to the configuration of semi-persistent scheduling. All of these values fall within the bounds of the fields described in Table 48, except the resource block assignment field which is required to be set to "all ones" for SPS release. This can be achieved by setting the "*Resource Block Start*" field in the DCI UL PDU to 0xFF; in this case the "*Number of Resource Blocks*" field is invalid and ignored.

3.3.1.4 SUBFRAME Errors

Section removed. Please see section 3.2.6.1.

3.3.2 Downlink Data

3.3.2.1 TX.request

The format of the `TX.request` message is described in Table 49. 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 `SUBFRAME.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	Size (bytes)	Description
SFN/SF	2	The SFN/SF in this message should be the same as the corresponding `SUBFRAME.request` message. A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Number of PDUs	2	Number of PDUs included in this message.
For each PDU		

Field	*Size (bytes)*	*Description*
PDU Length	2	The total length (in bytes) of this PDU description and the PDU data. This should always be a multiple of 32-bits.
PDU index	2	This is a count value which starts from 0. It is incremented following each BCH, MCH, PCH or DLSCH PDU. This value was included in `SUBFRAME.request` and associates the data to the control information. It is reset to 0 at the beginning of each subframe. Range 0 → 65535
PDU data	variable	Always a multiple of 32-bits.

Table 49: `TX.request` message body

3.3.2.2 Downlink Data Errors

Section removed. Please see section 3.2.6.1.

3.3.3 Uplink Data

3.3.3.1 RX.indication

The format of the `RX.indication` message is shown in Table 50. This message contains the following information:

- For each ULSCH PDU configured in `SUBFRAME.request` an ULSCH PDU is included
- For each ULSCH_CQI_RI PDU configured in `SUBFRAME.request` an ULSCH and CQI PDU is included
- For each ULSCH_HARQ PDU configured in `SUBFRAME.request` an ULSCH PDU is included. The HARQ information is included in the `HARQ.indication` message
- For each ULSCH_CQI_HARQ_RI PDU configured in `SUBFRAME.request` an ULSCH and CQI PDU is included. The HARQ information is included in the `HARQ.indication` message
- For each UCI_CQI PDU configured in `SUBFRAME.request` a CQI PDU is included.
- For each UCI_SR PDU configured in `SUBFRAME.request`, the SR RNTI will be included in an `SR.indication` if received from the UE.
- For each UCI_SR_HARQ PDU configured in `SUBFRAME.request` the HARQ information is included in the `HARQ.indication` message. If a SR is received from the UE, the RNTI will be included in an SR.indication.
- For each UCI_CQI_HARQ PDU configured in `SUBFRAME.request` a CQI PDU is included. The HARQ information is included in the `HARQ.indication` message.

Field	*Size (bytes)*	*Description*
SFN/SF	2	The SFN/SF of the SUBFRAME this information was received in. A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Number of PDUs	2	Number of PDUs included in this message.
For (Number of PDUs) {		

Field	*Size (bytes)*	*Description*
RNTI	2	The RNTI passed to the PHY in a ULSCH PDU or UCI PDU. See [6] section 8. Value: 0 → 65535.
Length	2	Length of PDU. For ULSCH and CQI the length is in bytes. For SR this field is invalid, and should be ignored.
Data Offset	2	Gives the PDU#i data address offset from the beginning of the 'Number of PDUs' field. For ULSCH an offset of 0 indicates a CRC or decoding error. For CQI an offset of 0 indicates a CRC or decoding error.
Received PDU Type	1	The type of received PDU. Following types are defined. 0: ULSCH 1:CQI
RI	1	The rank indication reported by the UE, See Secion 7.2 in TS36.213. Value: 1..4, For ULSCH this field is always valid, For CQI this field is not valid if a ULSCH with the same RNTI was received.
Timing Advance Vector Real Part	2	Timing advance vector real part computed by PHY. See Appendix: Timing Advance Calculation
Timing Advance vector Imaginary part	2	Timing advance vector imaginary part computed by PHY. See Appendix: Timing Advance Calculation
}		
PDU#1	Variable	Contents of PDU#1. The definition of the PDUs are given in Section 3.3.3.1.1 *et seq*.
Reserved	Variable	Align to 32-bit boundary
PDU#2	Variable	Contents of PDU#2
Reserved	Variable	Align to 32-bit boundaryy
..	...	...
PDU#n	Variable	Contents of PDU#n
Reserved	Variable	Align to 32-bit boundary

Table 50: `RX.indication` message body

3.3.3.1.1 ULSCH PDU

The contents of the ULSCH PDU sent from the PHY to L2/L3 software are shown in Figure 39. The ULSCH PDU is a MAC PDU; its format is described in [2].

MAC PDU

Figure 39: Contents of ULSCH PDU

3.3.3.1.2 CQI PDU

The format of DL CQI feedback and reports varies depending upon the channel used for feedback (PUSCH or PUCCH) and the DL transmission mode. This is detailed in [9]. The formats differ in the fields reported and the resultant number of bits required for the report.

The CQI PDU contains DL CQI feedback in raw format and is described in Table 51.

Field	***Size (bytes)***	***Description***
CQI raw report	$\left\lceil \frac{\#bits}{8} \right\rceil$	Raw format CQI report as defined in [9]. The first bit of the CQI report is bit [0] of byte 0.
Reserved	Variable	Align to 32-bit boundary
PUCCH RSSI	1	An estimate of the received PUCCH energy for this CQI report. If this CQI was conveyed by PUSCH, this field is reserved. Value: 0-255, representing -127.5 dBfs up to 0 dBfs, with 0.5 dB step size.
Reserved	3	Align to 32-bit boundary

Table 51: CQI PDU format

3.3.3.2 HARQ.indication

The format of the uplink HARQ control from the UE is dependent on whether a TDD or FDD PHY is used. To accommodate this difference two `HARQ.indication` messages are defined, one for TDD and one for FDD.

The `HARQ.indication` messages provide the following results for each ACK/NACK report.

- ACK – The PHY confidently detected an ACK
- NACK – The PHY confidently detected an NACK
- DTX – The PHY confidently detected that the UE did not transmit an ACK/NACK response
- ACK or NACK – The PHY is unsure whether it detected an ACK or NACK.
- ACK or DTX - The PHY is unsure whether it detected an ACK or DTX
- NACK or DTX - The PHY is unsure whether it detected an NACK or DTX
- ACK or NACK or DTX - The PHY is unsure whether it detected an ACK or NACK or DTX

3.3.3.2.1 TDD Format

The format of the `HARQ.indication` message for a TDD PHY is given in Table 52.

The bundling and multiplexing options resulting in this indication were passed to the PHY in an uplink subframe configuration PDU. If the ACK/NACK is combined with either CQI or SR information then a special ACK/NACK encoding is used which reports the number of ACKs, rather than providing specific ACK/NACK values. This is identified separately and called SPECIAL_BUNDLING in this API. (See [6] section 7.3 and section 3.3.1.2.2.9 of this document for more information.)

Field	*Size (bytes)*	*Description*
SFN/SF	2	The SFN/SF of the SUBFRAME this information was received in. A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Number of HARQs	2	Number of HARQs included in this message.
For (Number of HARQs)		
RNTI	2	The RNTI passed to the PHY in an uplink subframe configuration PDU. See [6] section 8. Value: 0 → 65535.
Mode	1	The format of the ACK/NACK response expected. 0 = BUNDLING 1 = MULTIPLEXING 2= SPECIAL_BUNDLING
Number of ACK/NACK	1	The number of ACK/NACK results reported for this UE. See [6] section 10. Value: 1 → 4 This field is invalid for special bundling.
HARQ Data	4	The format of the data is dependent on the HARQ mode; BUNDLING, MULTIPLEXING, or SPECIAL BUNDLING. See Table 53 to Table 55.
Bundle Scrambling Sequence	1	The detected index of the scrambling sequence for ACK/NACK bundling, which is used by MAC to determine N_bundled (see 5.2.2.6 in [9]). This field is only valid for ACK/NACK bundling in PUSCH.
PUCCH RSSI	1	An estimate of the received PUCCH energy for this HARQ report. If this HARQ was conveyed by PUSCH, this field is reserved. Value: 0-255, representing -127.5 dBfs up to 0 dBfs, with 0.5 dB step size.
Reserved	2	Align to 32-bit boundary

Table 52: `HARQ.indication` message body for TDD

Field	*Size (bytes)*	*Description*
Value 0	1	Indicates HARQ results. Range 1 → 7 1 = ACK 2 = NACK 3 = ACK or NACK 4 = DTX 5 = ACK or DTX 6 = NACK or DTX 7 = ACK or NACK or DTX

Field	Size (bytes)	Description
Value 1	1	Indicates HARQ results. Range 1 → 7, see above.
Reserved	2	Align to 32-bit boundary

Table 53: TDD HARQ data format for mode = BUNDLING

Field	Size (bytes)	Description
Value 0	1	Indicates HARQ results. Range 1 → 7 1 = ACK 2 = NACK 3 = ACK or NACK 4 = DTX 5 = ACK or DTX 6 = NACK or DTX 7 = ACK or NACK or DTX
Value 1	1	Indicates HARQ results. Range 1 → 7, see above.
Value 2	1	Indicates HARQ results. Range 1 → 7, see above.
Value 3	1	Indicates HARQ results. Range 1 → 7, see above.

Table 54: TDD HARQ data format for mode = MULTIPLEXING

Field	Size (bytes)	Description
Value 0	1	Number of ACK among multiple ACK/NACK responses, see [6] table 7.3.-1 0 = 0 or None (UE detect at least one DL assignment is missed) 1 = 2 or 5 or 8 ACKs reported 2 = 3 or 6 or 9 ACKs reported 3 = 1 or 4 or 7 ACKs reported 4 = DTX (UE did not transmit anything)
Reserved	3	Align to 32-bit boundary

Table 55: TDD HARQ data format for mode = SPECIAL BUNDLING

3.3.3.2.2 FDD Format

The format of the `HARQ.indication` message for a FDD PHY is given in Table 56.

Field	*Size (bytes)*	*Description*
SFN/SF	2	The SFN/SF of the SUBFRAME this information was received in. A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Number of HARQs	2	Number of HARQs included in this message.
For (Number of HARQs)		
RNTI	2	The RNTI passed to the PHY in an uplink subframe configuration PDU. Value: 0 → 65535.
HARQ TB1	1	HARQ feedback of 1st TB. Range 1 → 7 1 = ACK 2 = NACK 3 = ACK or NACK 4 = DTX 5 = ACK or DTX 6 = NACK or DTX 7 = ACK or NACK or DTX
HARQ TB2	1	HARQ feedback of 2nd TB. Range 1 → 7 1 = ACK 2 = NACK 3 = ACK or NACK 4 = DTX 5 = ACK or DTX 6 = NACK or DTX 7 = ACK or NACK or DTX
PUCCH RSSI	1	An estimate of the received PUCCH energy for this HARQ report. If this HARQ was conveyed by PUSCH, this field is reserved. Value: 0-255, representing -127.5 dBfs up to 0 dBfs, with 0.5 dB step size.
Reserved	3	Align to 32 bits

Table 56: `HARQ.indication` message body for FDD

3.3.3.3 CRC.indication

The format of the `CRC.indication` message is given in Table 57. Note that a `CRC.indication` may legitimately contain zero CRCs.

Field	*Size (bytes)*	*Description*
SFN/SF	2	The SFN/SF of the SUBFRAME this information was received in. A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Number of CRCs	2	Number of CRCs included in this message, only CRCs for ULSCH payload are included in this message. (Range includes 0 i.e., no CRCs to report for the relevant subframe.)
Number of Resource Blocks	1	Number of resource blocks in the UL bandwidth for interference power and RSSI reporting.
Reserved	3	
For (Number of CRCs)		
RNTI	2	The RNTI passed to the PHY in an uplink subframe configuration PDU. Value: 0 → 65535.
CRC Flag	1	A flag indicating if a CRC error was detected. 0 = CRC_CORRECT 1 = CRC_ERROR
RSSI	1	The power in the allocated RBs for this UL grant. Value: 0-255, representing -127.5 dBfs up to 0 dBfs, with 0.5 dB step size.
Effective SNR	1	A post-processed SNR from the output of the space-freq-equalizer (SFE). This value is a good indicator of expected decode performance. Value: 0-255, representing -64dB to 63.5dB, with 0.5dB step size.
Reserved	1	This field was previously an additional SINR but is now reserved.
Reserved	2	
For (Number of Resource Blocks)		
Received Interference Power	1	The average resource element interference power at the corresponding RB Value: 0-255, representing -127.5 dBfs up to 0 dBfs, with 0.5 dB step size.
RSSI	1	The average resource element power (desired signal and all undesired signals) at the corresponding RB. Value: 0-255, representing -127.5 dBfs up to 0 dBfs, with 0.5 dB step size.
Reserved	2	

Table 57: `CRC.indication` message body

3.3.3.4 SR.indication

The format of the SR.`indication` message is given in Table 58.

Field	*Size (bytes)*	*Description*
SFN/SF	2	The SFN/SF of the SUBFRAME this information was received in. A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Number of SRs	2	Number of SRs included in this message.
For (Number of SRs)		
RNTI	2	The RNTI passed to the PHY in an uplink subframe configuration PDU. Value: 0 → 65535.
PUCCH RSSI	1	An estimate of the received PUCCH energy for this positive SR. Value: 0-255, representing -127.5 dBfs up to 0 dBfs, with 0.5 dB step size.
Reserved	1	Align to 32-bit boundary

Table 58: `SR.indication` message body

3.3.3.5 RACH.indication

The format of the `RACH.indication` message is given in Table 59.

Field	*Size (bytes)*	*Description*
SFN/SF	2	The SFN/SF of the SUBFRAME this information was received in. A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Number of Preambles	1	Number of RACH preambles
Reserved	1	Align to 32-bit boundary
For (Number of Preambles) {		
RNTI	2	The RA-RNTI value See [2] section 5.1.4 Value: 0 → 65535.
Preamble	1	The detected preamble Value: 0 → 63
Reserved	1	
Timing Advance	2	The measured timing advance for the preamble in units of 16Ts Value: 0 → 1282
RSSI	1	The total received power of the detected preamble. Value: 0-255, representing -127.5 dBfs up to 0 dBfs, with 0.5 dB step size.

Field	Size (bytes)	Description
Reserved	1	

Table 59: `RACH.indication` message body

3.3.3.6 SRS.indication

The format of the `SRS.indication` message is given in Table 60.

Field	Size (bytes)	Description
SFN/SF	2	The SFN/SF of the SUBFRAME this information was received in. A 2-byte value where, [15:4] SFN, range 0 → 1023 [3:0] SF, range 0 → 9
Number of UEs	1	Number of UEs contributing to the uplink SRS
Reserved	1	32-bit alignment
For (Number of UEs){		
RNTI	2	The RNTI passed to the PHY in an uplink subframe configuration PDU
Doppler spread estimation	2	Doppler spread estimation for the target UE. This is the temporal channel correlation coefficient between two consecutive periodic SRS resources. Values: 0 → 255, Representing value from 0 to 1 of ρ in step 1/255.
Number of resource blocks	1	Number of resource blocks to be reported for this UE
RB start	1	The starting point of the RBs to be reported.
Reserved	2	
For (Number of RBs) {		
SNR	1	Post equalisation SNR. Value: 0-255, representing -64dB to 63.5dB, with 0.5dB step size.
}		
Reserved	N	Align to 32 bits (size depends on number of preceeding RBs)
}		

Table 60: `SRS.indication` message body

4 Message Encoding

The API messages can be encoded using either little-endian byte order or big-endian byte order. This allows the L2/L3 software to select the endianess native to its host-processor, or OS build. Support for both endian options is provided in the PHY at compile-time. The default build will always be big-endian, also called network-byte ordering. If the PHY has been compiled with a different endianess, from the L2/L3 software, the API mechanism will not work.

4.1 Byte Encoding

The API messages consist of parameters which are assigned a length in bytes. The parameters are encoded using one of the following formats:

- Unsigned
- 2's complement signed

An example of 1,2 and 4-byte values encoded in both little-endian and big-endian format is given in Figure 40.

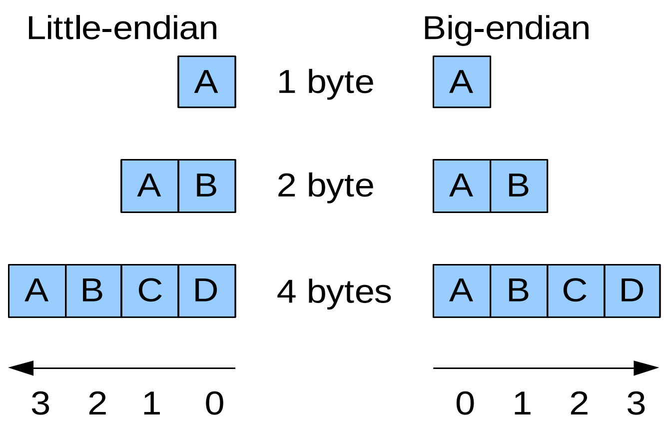

Figure 40: Endianess

5 Transport Layer

The relationship between the transport mechanism and L1 API is given in Figure 41. This shows one L2/L3 entity connected to two PHY entities. The connection is managed by the transport layer. A detailed description of the transport layer is out of scope of this document. Figure 41 is included to illustrate support for sectorisation where multiple PHYs can be handled by a transport layer. In addition, the fragmentation of L1 API messages, during transit from the L2/L3 entity to the PHY entity, must be handled by the transport layer.

The L1 API, defined in this document, assumes that the transport layer provides a reliable, in-order delivery of messages.

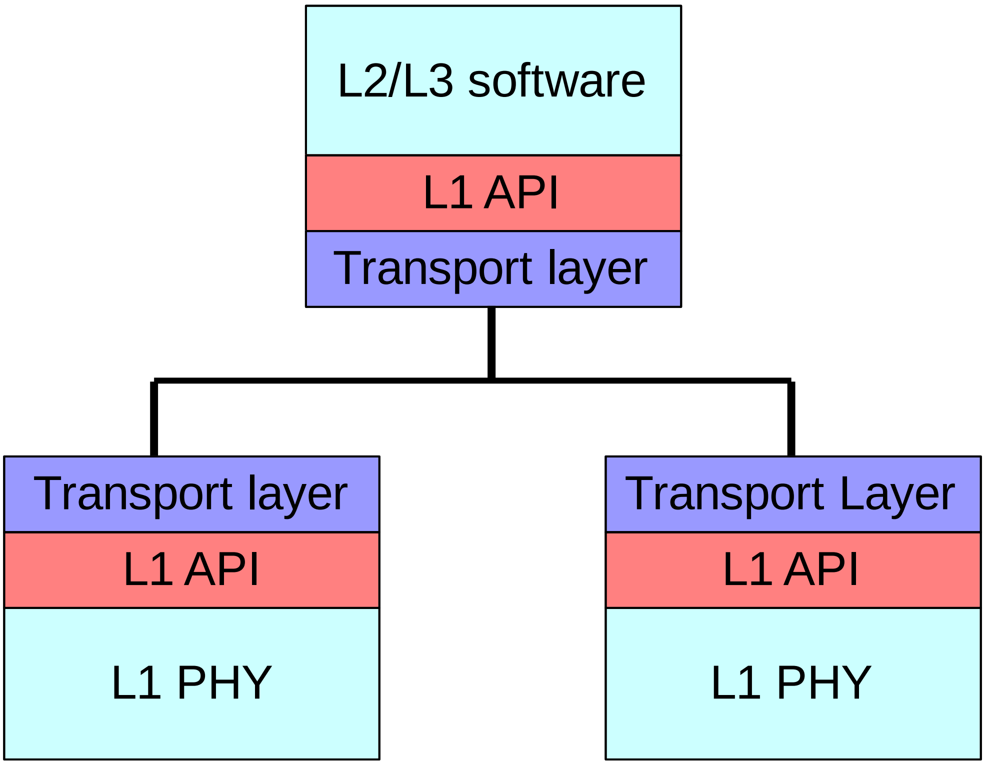

Figure 41: Transport Mechanism

Appendix A Timing Advance Calculation for PUSCH

Due to the nature of LTE uplink signals, it is not possible in all channel conditions to provide accurate timing advance estimation from a single PUSCH allocation. For this reason, L2 must calculate a timing advance estimate based on a number of PUSCH allocations to a given user. An algorithm for this is described below.

Assume that following an uplink PUSCH grant, the PHY reports a complex timing advance vector with real and imaginary parts $V_r(in)$ and $V_i(in)$. The previous estimate of the timing vector, as maintained by L2, is given by real and imaginary parts $V_r(n-1)$ and $V_i(n-1)$. The real part of the new timing advance vector estimate is found from:

$$V_r(n) = \begin{cases} (1-\gamma)V_r(n-1) + \gamma V_r(in), n \geq 1 \\ V_{r(in),} \qquad n\langle 1 \end{cases}$$

where the gain γ is chosen based on performance requirements, typically between 1/8 and 1/128. The imaginary part $V_i(n)$ can be computed similarly.

Once enough iterations have occurred a timing advance estimate can be calculated. The phase for the timing advance vector at current time slot can be computed as:

$$\varphi(n) = \tan^{-1}\frac{V_i(n)}{V_r(n)}$$

Then the timing advance itself can be computed as:

$$T(n) = \frac{N_{FFT}}{2\pi K_s}\varphi(n),$$

where the frequency spacing K_s is 3, and the size of FFT $N_{\mathrm{FFT}} = 1024$ for 10MHz system and $N_{\mathrm{FFT}} = 512$ for 5MHz system.

Appendix B L1 API Message Timing

The picoArray™ PHY implementation imposes a strict message sequence on the API messages (see section 2.2.3). It also requires that message exchanges between L2/L3 and L1 occur within predefined time windows. These windows are designed to ensure that both L1 and L2/L3 have the information necessary to complete their respective tasks according to the LTE standard and that the architecture of both can be kept clear and concise.

Figure 42 shows the message windows at the interface to L1, whilst Table 61 and Table 62 detail the timings with respect to the frame interrupt. The reference plane for these timings is within the picoArray™, at the interface to the L1 firmware itself; any latency and throughput constraints imposed by the physical transport mechanism to L2/L3, or by additional picoArray™ firmware associated with this transport mechanism, are in addition to the timings shown in Figure 42.

The following applies to Figure 42:

- The air subframes depicted refer to the DL subframe at the digital interface between the PHY and the radio hardware.
- TTI(x) refers to the subframe interrupt (see section 2.2.1) for air subframe x.
- Only a subset of messages are shown for clarity.
- For messages from L2/L3 to L1, the message transfer must occur entirely within the window shown i.e., the transfer must begin at some point after the left-hand side of the window and complete before the right-hand side.
- For messages from L1 to L2/L3, the message becomes available from the PHY at the left-hand edge of the window. Transfer must be completed by the right-hand edge.
- Note that the subframe interrupt is advanced with respect to the on-air subframe timing.

Message	***Time delay from TTI(n) (ms)***	***Time delay from subframe (n) on-air (ms)***
On-Air Subframe Start	1.326	0.000
HARQ.ind	3.013	1.688
CRC.ind	4.220	2.895
RX.ind, RACH.ind, SRS.ind	4.800	3.474
SR.ind	3.500	2.174

Table 61: Timings of L1 API messages from L1 to L2 with respect to TTI frame interrupt

Message	***Window Start from TTI(n) (ms)***	***Window End from TTI(n) (ms)***
CONFIG.req	0.000	0.250
SUBFRAME.req	0.000	0.250
TX.req	0.000	0.250
HI_DCI.req	0.220	0.776

Table 62: Message windows for L1 API messages from L2 to L1 with respect to TTI frame interrupt

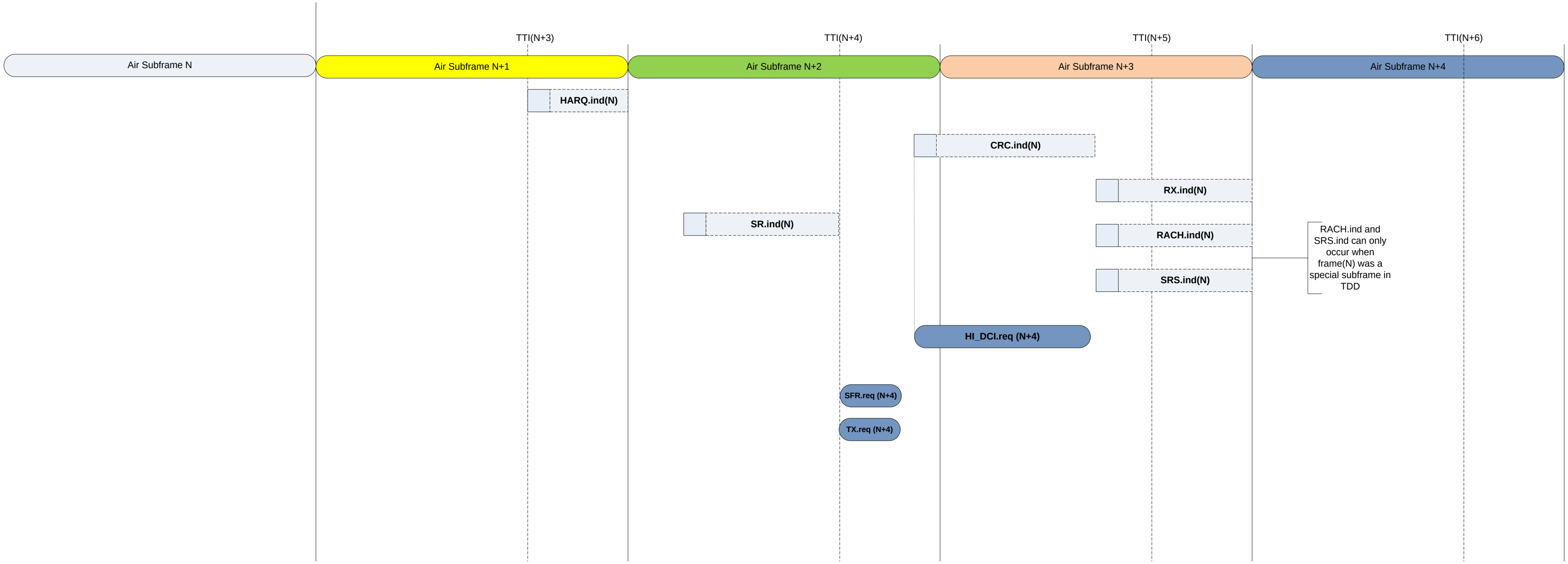

Figure 42: L1 API message timings

Appendix C DTX-ACK Probability Mappings

Table 63 shows the mapping of TLV value to probability for the DTX-ACK configuration TLVs listed in section 0.

TLV Value	*Target DTX to ACK Probability*
0	0.5
1	0.2
2	0.1
3	0.09
4	0.07
5	0.05
6	0.02
7	0.015
8	0.013
9	0.012
10	0.011
11	0.0105
12	0.01
13	0.0095
14	0.009
15	0.0087
16	0.0085
17	0.008
18	0.007
19	0.005
20	0.002
21	0.001
22	0.0009
23	0.0007
24	0.0005
25	0.0002
26	0.0001
27	0.00009
28	0.00007
29	0.00005
30	0.00002
31	0.00001

Table 63: Mapping of TLV value to DTX-ACK probability