**3GPP TSG-RAN WG2 Meeting #101 *R2-180XXXX***

**Athens, Greece, 26th February - 2nd March 2018**

|  |
| --- |
| *CR-Form-v11.2* |
| **CHANGE REQUEST** |
|  |
|  | **38.331** | **CR** | **CRNum** | **rev** | **-** | **Current version:** | **15.0.1** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **X** | Radio Access Network | **X** | Core Network |  |

|  |
| --- |
|  |
| ***Title:***  | TP related to email discussion #23 for 38.331 ASN.1 review part 5 |
|  |  |
| ***Source to WG:*** | Rapporteur (Ericsson) |
| ***Source to TSG:*** | R2 |
|  |  |
| ***Work item code:*** | NR\_newRAT-Core |  | ***Date:*** | 2018-02-26 |
|  |  |  |  |  |
| ***Category:*** | **F** |  | ***Release:*** | Rel-15 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)Rel-12 (Release 12)Rel-13 (Release 13)Rel-14 (Release 14)Rel-15 (Release 15)Rel-16 (Release 16)* |
|  |  |
| ***Reason for change:*** | Including:- Corrections identified during ASN.1 review (RAN2 NR AH 1801), - Corrections identified in email discussion #23 for 38.331 ASN.1 review part 5 - Agreements made in RAN2#101 relevant for this areaThis text proposal is based onR2-1801218 Baseline TS 38331 v1.0.1 for ASN.1 review |
|  |  |
| ***Summary of change:*** |  |
|  |  |
| ***Consequences if not approved:*** |  |
|  |  |
| ***Clauses affected:*** |  |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  |  |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  |  |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  |  |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |

# 1 Scope

The present document specifies the Radio Resource Control protocol for the radio interface between UE and NG-RAN.

The scope of the present document also includes:

- the radio related information transported in a transparent container between source gNB and target gNB upon inter gNB handover;

- the radio related information transported in a transparent container between a source or target gNB and another system upon inter RAT handover.

- the radio related information transported in a transparent container between a source eNB and target gNB during E-UTRA-NR Dual Connectivity.

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 38.300: "NR; Overall description; Stage 2".

[3] 3GPP TS 38.321: "NR; Medium Access Control (MAC); Protocol specification".

[4] 3GPP TS 38.322: "NR; Radio Link Control (RLC) protocol specification".

[5] 3GPP TS 38.323: "NR; Packet Data Convergence Protocol (PDCP) protocol specification".

[6] ITU-T Recommendation X.680 (08/2015) "Information Technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation" (Same as the ISO/IEC International Standard 8824-1).

[7] ITU-T Recommendation X.681 (08/2015) "Information Technology - Abstract Syntax Notation One (ASN.1): Information object specification" (Same as the ISO/IEC International Standard 8824-2).

[8] ITU-T Recommendation X.691 (08/2015) "Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)" (Same as the ISO/IEC International Standard 8825-2).

[9] 3GPP TS 38.215: "NR; Physical layer measurements".

[10] 3GPP TS 36.331: "Evolved Universal Terrestrial Radio Access (E-UTRA) Radio Resource Control (RRC); Protocol Specification".

[11] 3GPP TS 33.501: "Security Architecture and Procedures for 5G System".

[12] 3GPP TS 38.104: "NR; Base Station (BS) radio transmission and reception".

[13] 3GPP TS 38.213: "NR; Physical layer procedures for control".

[14] 3GPP TS 38.133: "NR; Requirements for support of radio resource management".

[15] 3GPP TS 38.101: "NR; User Equipment (UE) radio transmission and reception".

[16] 3GPP TS 38.211: "NR; Physical channels and modulation".

[17] 3GPP TS 38.212: "NR; Multiplexing and channel coding".

[18] ITU-T Recommendation X.683 (08/2015) "Information Technology - Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications" (Same as the ISO/IEC International Standard 8824-4).

[19] 3GPP TS 38.214: "NR; Physical layer procedures for data".

# 3 Definitions, symbols and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

**Field:** The individual contents of an information element are referred as fields.

**Floor:** Mathematical function used to 'round down' i.e. to the nearest integer having a lower or equal value.

**Information element:** A structural element containing a single or multiple fields is referred as information element.

**Primary Cell**: The MCG cell, operating on the primary frequency, in which the UE either performs the initial connection establishment procedure or initiates the connection re-establishment procedure.

**Primary SCG Cell**: For dual connectivity operation, the SCG cell in which the UE performs random access when performing the Reconfiguration with Synch procedure.

**RLC bearer configuration:** The lower layer part of the radio bearer configuration comprising the RLC and logical channel configurations.

**Secondary Cell**: For a UE configured with CA, a cell providing additional radio resources on top of Special Cell.

**Secondary Cell Group**: For a UE configured with dual connectivity, the subset of serving cells comprising of the PSCell and zero or more secondary cells.

**Serving Cell**: For a UE in RRC\_CONNECTED not configured with CA/DC there is only one serving cell comprising of the primary cell. For a UE in RRC\_CONNECTED configured with CA/ DC the term 'serving cells' is used to denote the set of cells comprising of the Special Cell(s) and all secondary cells.

**Special Cell:** For Dual Connectivity operation the term Special Cell refers to the PCell of the MCG or the PSCell of the SCG, otherwise the term Special Cell refers to the PCell.

**SRB1S:** The SCG part of MCG split SRB1 for EN-DC.

**SRB2S:** The SCG part of MCG split SRB2 for EN-DC.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

5GC 5G Core Network

ACK Acknowledgement

AM Acknowledged Mode

ARQ Automatic Repeat Request

AS Access Stratum

ASN.1 Abstract Syntax Notation One

BLER Block Error Rate

BWP Bandwidth Part

CA Carrier Aggregation

CCCH Common Control Channel

CG Cell Group

CMAS Commercial Mobile Alert Service

CP Control Plane

C-RNTI Cell RNTI

CSI Channel State Information

DC Dual Connectivity

DCCH Dedicated Control Channel

DCI Downlink Control Information

DL Downlink

DL-SCH Downlink Shared Channel

DRB (user) Data Radio Bearer

DRX Discontinuous Reception

DTCH Dedicated Traffic Channel

EPC Evolved Packet Core

EPS Evolved Packet System

ETWS Earthquake and Tsunami Warning System

E-UTRA Evolved Universal Terrestrial Radio Access

E-UTRAN Evolved Universal Terrestrial Radio Access Network

FDD Frequency Division Duplex

FFS For Further Study

GERAN GSM/EDGE Radio Access Network

GNSS Global Navigation Satellite System

GSM Global System for Mobile Communications

HARQ Hybrid Automatic Repeat Request

IE Information element

IMSI International Mobile Subscriber Identity

kB Kilobyte (1000 bytes)

L1 Layer 1

L2 Layer 2

L3 Layer 3

MAC Medium Access Control

MCG Master Cell Group

MIB Master Information Block

N/A Not Applicable

PCell Primary Cell

PDCP Packet Data Convergence Protocol

PDU Protocol Data Unit

PLMN Public Land Mobile Network

PSCell Primary Secondary Cell

QoS Quality of Service

RAN Radio Access Network

RAT Radio Access Technology

RLC Radio Link Control

RNTI Radio Network Temporary Identifier

ROHC RObust Header Compression

RRC Radio Resource Control

RS Reference Signal

SCell Secondary Cell

SCG Secondary Cell Group

SFN System Frame Number

SFTD SFN and Frame Timing Difference

SI System Information

SIB System Information Block

SpCell Special Cell

SRB Signalling Radio Bearer

SSB Synchronization Signal Block

TAG Timing Advance Group

TDD Time Division Duplex

TM Transparent Mode

UE User Equipment

UL Uplink

UM Unacknowledged Mode

UP User Plane

In the ASN.1, lower case may be used for some (parts) of the above abbreviations e.g. c-RNTI.

# 4 General

## 4.1 Introduction

This specification is organised as follows:

- sub-clause 4.2 describes the RRC protocol model;

- sub-clause 4.3 specifies the services provided to upper layers as well as the services expected from lower layers;

- sub-clause 4.4 lists the RRC functions;

- clause 5 specifies RRC procedures, including UE state transitions;

- clause 6 specifies the RRC messages in ASN.1 and description;

- clause 7 specifies the variables (including protocol timers and constants) and counters to be used by the UE;

- clause 8 specifies the encoding of the RRC messages;

- clause 9 specifies the specified and default radio configurations;

- clause 10 specifies generic error handling;

- clause 11 specifies the RRC messages transferred across network nodes;

- clause 12 specifies the UE capability related constraints and performance requirements.

## 4.2 Architecture

Editor's note The state model is still a subject for discussion.FFS

### 4.2.1 UE states and state transitions including inter RAT

Editor’s Note: For EN\_DC, only RRC\_CONNECTED is applicable.

A UE is either in RRC\_CONNECTED state or in RRC\_INACTIVE state when an RRC connection has been established. If this is not the case, i.e. no RRC connection is established, the UE is in RRC\_IDLE state. The RRC states can further be characterised as follows:

**- RRC\_IDLE**:

- A UE specific DRX may be configured by upper layers;

- UE controlled mobility based on network configuration;

- The UE:

- Monitors a Paging channel;

- Performs neighbouring cell measurements and cell (re-)selection;

- Acquires system information.

**- RRC\_INACTIVE**:

- A UE specific DRX may be configured by upper layers or by RRC layer;

- UE controlled mobility based on network configuration;

- The UE stores the AS context;

- The UE:

- Monitors a Paging channel;

- Performs neighbouring cell measurements and cell (re-)selection;

- Performs RAN-based notification area updates when moving outside the RAN-based notification area;

Editor’s Note: FFS Whether a RAN-based notification area is always configured or not.

Editor’s Note: FFS UE behavior if it is decided that a RAN-based notification area is not always configured.

- Acquires system information.

**- RRC\_CONNECTED:**

- The UE stores the AS context.

- Transfer of unicast data to/from UE.

- At lower layers, the UE may be configured with a UE specific DRX.

- For UEs supporting CA, use of one or more SCells, aggregated with the SpCell, for increased bandwidth;

- For UEs supporting DC, use of one SCG, aggregated with the MCG, for increased bandwidth;

- Network controlled mobility within NR and to/from E-UTRAN.

- The UE:

- Monitors a Paging channel;

- Monitors control channels associated with the shared data channel to determine if data is scheduled for it;

- Provides channel quality and feedback information;

- Performs neighbouring cell measurements and measurement reporting;

- Acquires system information.

Figure 4.2.1-1 illustrates an overview of UE RRC state machine and state transitions in NR. A UE has only one RRC state in NR at one time.

****

Figure 4.2.1-1: UE state machine and state transitions in NR

Figure 4.2.1-2 illustrates an overview of UE state machine and state transitions in NR as well as the mobility procedures supported between NR/NGC and E-UTRAN/EPC.

****

Figure 4.2.1-2: UE state machine and state transitions between NR/NGC and E-UTRAN/EPC

The UE state machine, state transition and mobility procedures between NR/NGC and E-UTRA/NGC is FFS.

### 4.2.2 Signalling radio bearers

## 4.3 Services

### 4.3.1 Services provided to upper layers

The RRC protocol offers the following services to upper layers:

- Broadcast of common control information;

- Notification of UEs in RRC\_IDLE, e.g. about a terminating call [FFS, for ETWS, for CMAS];

- Transfer of dedicated control information, i.e. information for one specific UE.

### 4.3.2 Services expected from lower layers

In brief, the following are the main services that RRC expects from lower layers:

- PDCP: integrity protection, ciphering and in-sequence delivery of information without duplication [FFS if duplication need to be listed];

- RLC: reliable transfer of information, without introducing duplicates and with support for segmentation.

## 4.4 Functions

The RRC protocol includes the following main functions:

- Broadcast of system information:

- Including NAS common information;

- Information applicable for UEs in RRC\_IDLE and RRC\_INACTIVE, e.g. cell (re-)selection parameters, neighbouring cell information and information (also) applicable for UEs in RRC\_CONNECTED, e.g. common channel configuration information.

- [FFS Including ETWS notification, CMAS notification]

- RRC connection control:

- Paging;

- Establishment/modification/suspension/resumption/release of RRC connection, including e.g. assignment/modification of UE identity (C-RNTI), establishment/modification/release of SRBs, access class barring;

Editor’s note: The terminology for establishment/modification/suspension/resumption is FFS.

- Initial security activation, i.e. initial configuration of AS integrity protection (SRBs) and AS ciphering (SRBs, DRBs);

- RRC connection mobility including e.g. intra-frequency and inter-frequency handover, associated security handling, i.e. key/algorithm change, specification of RRC context information transferred between network nodes;

- Establishment/modification/release of RBs carrying user data (DRBs);

- Radio configuration control including e.g. assignment/modification of ARQ configuration, HARQ configuration, DRX configuration;

- In case of DC, cell management including e.g. change of PSCell, addition/modification/release of SCG cell(s);

- In case of CA, cell management including e.g. addition/modification/release of SCell(s).

- Recovery from radio link failure;

- Inter-RAT mobility including e.g. security activation, transfer of RRC context information;

- Measurement configuration and reporting:

- Establishment/modification/release of measurements (e.g. intra-frequency, inter-frequency and inter- RAT measurements);

- Setup and release of measurement gaps;

- Measurement reporting;

- Other functions including e.g. transfer of dedicated NAS information, transfer of UE radio access capability information [FFS support for RAN sharing (multiple PLMN identities)];

# 5 Procedures

## 5.1 General

### 5.1.1 Introduction

This section covers the general requirements.

### 5.1.2 General requirements

The UE shall:

1> process the received messages in order of reception by RRC, i.e. the processing of a message shall be completed before starting the processing of a subsequent message;

NOTE 1: Network may initiate a subsequent procedure prior to receiving the UE's response of a previously initiated procedure.

1> within a sub-clause execute the steps according to the order specified in the procedural description;

1> consider the term 'radio bearer' (RB) to cover SRBs and DRBs unless explicitly stated otherwise;

1> set the *rrc-TransactionIdentifier* in the response message, if included, to the same value as included in the message received from NR that triggered the response message;

1> upon receiving a choice value set to *setup*:

2> apply the corresponding received configuration and start using the associated resources, unless explicitly specified otherwise;

1> upon receiving a choice value set to *release*:

2> clear the corresponding configuration and stop using the associated resources;

1> in case the size of a list is extended, upon receiving an extension field comprising the entries in addition to the ones carried by the original field (regardless of whether NR signals more entries in total); apply the following generic behaviour unless explicitly stated otherwise:

2> create a combined list by concatenating the additional entries included in the extension field to the original field while maintaining the order among both the original and the additional entries;

2> for the combined list, created according to the previous, apply the same behaviour as defined for the original field;

## 5.2 System information

Editor’s Note: Targeted for completion in June 2018. For EN\_DC, only parts related to MIB acquisition, in sub-clauses 5.2.2.3.1 and 5.2.2.4.1, are applicable.

### 5.2.1 Introduction

System Information (SI) is divided into the *MasterInformationBlock* (MIB) and a number of *SystemInformationBlocks* (SIBs) where:

- the *MasterInformationBlock* (MIB) is always transmitted on the BCH (refer Figure 5.2.2.X.X FFS\_Ref) with a periodicity of 80 ms and repetitions made within 80 ms [X] and it includes parameters that are needed to acquire *SystemInformationBlockType1* (SIB1) from the cell [FFS TBD-RAN1];

- the *SystemInformationBlockType1* (SIB1) is transmitted on the DL-SCH with a periodicity of [X] and repetitions made within [X]. SIB1 includes information regarding the availability and scheduling (e.g. periodicity, SI-window size) of other SIBs. It also indicates whether they (i.e. other SIBs) are provided via periodic broadcast basis or only on-demand basis (refer Figure 5.2.2.X.X FFS\_Ref). If other SIBs are provided on-demand then SIB1 includes information for the UE to perform SI request;

- SIBs other than *SystemInformationBlockType1* are carried in *SystemInformation* (SI) messages, which are transmitted on the DL-SCH. Each SI message is transmitted within periodically occurring time domain windows (referred to as SI-windows);

- For PSCell and SCells, RAN provides the required SI by dedicated signalling. Nevertheless, the UE shall acquire MIB of the PSCell to get SFN timing of the SCG (which may be different from MCG). Upon change of relevant SI for SCell, RAN releases and adds the concerned SCell. For PSCell, SI can only be changed with Reconfiguration with Synch.

Editor’s Note: Reference to RAN1 specification may be used for the MIB/SIB1 periodicities [X].FFS

### 5.2.2 System information acquisition

#### 5.2.2.1 General UE requirements



Figure 5.2.2.X-X: System information acquisition

The UE applies the SI acquisition procedure to acquire the AS- and NAS information. The procedure applies to UEs in RRC\_IDLE, in RRC\_INACTIVE and in RRC\_CONNECTED.

The UE in RRC\_IDLE and RRC\_INACTIVE shall ensure having a valid version of (at least) the *MasterInformationBlock*, *SystemInformationBlockType1* as well as *SystemInformationBlockTypeX* through *SystemInformationBlockTypeY* (depending on support of the concerned RATs for UE controlled mobility).

The UE in RRC\_CONNECTED shall ensure having a valid version of (at least) the *MasterInformationBlock*, *SystemInformationBlockType1* as well as *SystemInformationBlockTypeX* (depending on support of mobility towards the concerned RATs).

The UE shall store relevant SI acquired from the currently camped/serving cell. A version of the SI that the UE acquires and stores remains valid only for a certain time. The UE may use such a stored version of the SI e.g. after cell re-selection, upon return from out of coverage or after SI change indication.

Editor’s Note: [FFS\_Standalone if the UE is required to store SI other than for the currently camped/serving cell].

Editor’s Note: [FFS\_Standalone if different versions of SIBs are provided].

Editor’s Note: [FFS\_Standalone UE may or shall store several versions of SI].

Editor’s Note: FFS\_Standalone To be updated when above is resolved. Another sub-clause under 5.2.2.2 can be considered depending on the resolution of above.

#### 5.2.2.2 SI validity and need to (re)-acquire SI

The UE shall apply the SI acquisition procedure as defined in clause 5.2.2.3 upon cell selection (e.g. upon power on), cell-reselection, return from out of coverage, after reconfiguration with sync completion, after entering RAN from another RAT; whenever the UE does not have a valid version in the stored SI.

Editor’s Note: [FFS\_Standalone if upon receiving HO command the SI acquisition depend on stored SI]

When the UE acquires a *MasterInformationBlock* or a *SystemInformationBlockType1* or a SI message in a currently camped/serving cell as described in clause 5.2.2.3, the UE shall store the acquired SI.

##### 5.2.2.2.1 SI validity

The UE shall:

1> delete any stored version of SI after [FFS] hours from the moment it was successfully confirmed as valid;

1> if the UE does not have in the stored SI a valid version for the required SI corresponding to the *systemInfoAreaIdentifier* and *systemInfoValueTag*/*systemInfoConfigurationIndex* of that SI in the currently camped/serving cell:

2> (re)acquire the SI as specified in clause 5.2.2.3.

NOTE: At the SI acquisition procedure, the UE may assume the acquired SI in the currently camped/serving cell to be valid in other cells than the currently camped/serving cell based on *systemInfoAreaIdentifier* and *systemInfoValueTag*/*systemInfoConfigurationIndex*.

Editor’s Note: [FFS\_Standalone terminology to be used is systemInfoValueTag or systemInfoConfigurationIndex]

Editor’s Note: [FFS\_Standalone terminology to be used for area ID is systemInfoAreaIdentifier]

Editor’s Note: [FFS\_Standalone whether the area ID and valuetag is separately signalled or as a single identifier]

Editor’s Note: [FFS\_Standalone whether the area ID is associated to each SIB/SI message or associated to a group of SIBs/SI messages or all SIBs/SI messages]

##### 5.2.2.2.2 SI change indication and PWS notification

A modification period is used, i.e. updated SI is provided in the modification period following the one where SI change indication is transmitted. RAN transmits SI change indication and PWS notification through paging. Repetitions of SI change indication may occur within preceding modification period.

Editor’s Note : The above descriptive text can remain in this sub-clause or moved under 5.2.1. FFS\_Standalone

If the UE is in RRC\_CONNECTED or is configured to use a DRX cycle smaller than the modification period in RRC\_IDLE or in RRC\_INACTIVE and receives a Paging message:

1> if the received Paging message includes the *etws*/*cmasNotification*;

2> the UE shall immediately re-acquire the SIB1 and apply the SI acquisition procedure as defined in sub-clause [X.X.X.X FFS\_Ref].

1> else, if the received Paging message includes the *systemInfoModification*;

2> the UE shall apply the SI acquisition procedure as defined in sub-clause [X.X.X.X FFS\_Ref] from the start of the next modification period.

NOTE: For PWS notification the SIB1 is re-acquired to know the scheduling information for the PWS messages.

Editor’s Note: [FFS\_Standalone if upon receiving a SI change indication the SI acquisition depend on stored SI]

Editor’s Note: [FFS\_Standalone if value tags and area identifier included in paging message to reacquire SIB1]

Editor’s Note: [FFS\_Standalone the update mechanism for access control notifications and other non-access control configuration updates]

#### 5.2.2.3 Acquisition of System Information

##### 5.2.2.3.1 Acquisition of MIB and SIB1

The UE shall:

1> if the cell is a PSCell:

2> acquire the *MIB*, which is scheduled as specified in TS 38.213 [13];

2> perform the actions specified in section 5.2.2.4.1;

1> else:

2> acquire the *MIB,* which is scheduled as specified in TS 38.213 [13];

2> if the UE is unable to acquire the *MIB*;

3> follow the actions as specified in clause 5.2.2.5;

2> else:

3> perform the actions specified in section 5.2.2.4.1;

2> acquire the SystemInformationBlockType1 as specified in [X];

2> if the UE is unable to acquire the SystemInformationBlockType1:

3> follow the actions as specified in clause 5.2.2.5;

2> else:

3>perform the actions specified in section 5.2.2.4.2.

Editor’s Note: Reference to RAN1 [X] specification may be used for the scheduling of SIB1.FFS\_Standalone

##### 5.2.2.3.2 Acquisition of an SI message

When acquiring an SI message, the UE shall:

1> determine the start of the SI-window for the concerned SI message as follows:

Editor’s Note: [FFS\_Standalone the details of the mapping to subframes/slots where the SI messages are scheduled]

Editor’s Note: [FFS\_Standalone if there are any exceptions on e.g. subframes where SI messages cannot be transmitted]

Editor’s Note: [FFS\_Standalone if the SI-windows of different SI messages do not overlap].

Editor’s Note: [FFS\_Standalone if multiple SI messages can be mapped to same SI window]

Editor’s Note: [FFS\_Standalone if the length of SI-window is common for all SI messages or if it is configured per SI message]

Editor’s Note: [FFS\_Standalone if the UE may accumulate the SI-Message transmissions across several SI-Windows within the Modification Period]

1> if SI message acquisition not triggered due to UE request:

2> receive DL-SCH using the SI-RNTI from the start of the SI-window and continue until the end of the SI-window whose absolute length in time is given by *si-WindowLength*, or until the SI message was received;

2> if the SI message was not received by the end of the SI-window, repeat reception at the next SI-window occasion for the concerned SI message;

1> if SI message acquisition triggered due to UE request:

2> [FFS\_Standalone receive DL-SCH using the SI-RNTI from the start of the SI-window and continue until the end of the SI-window whose absolute length in time is given by si-WindowLength, or until the SI message was received];

2> [FFS\_Standalone if the SI message was not received by the end of the SI-window, repeat reception at the next SI-window occasion for the concerned SI message];

Editor’s Note: [FFS\_Standalone on the details of from which SI-window the UE shall receive the DL-SCH upon triggering the SI request.

Editor’s Note: [FFS\_Standalone on the details of how many SI-windows the UE should monitor for SI message reception if transmission triggered by UE request]

Editor’s Note: [FFS\_Standalone if UE need to monitor all the TTIs in SI window for receiving SI message]

1> store the acquired SI message as specified in clause 5.2.2.2.

Editor’s Note: FFS\_Standalone The procedural text for SI message acquisition triggered by UE request will be updated upon finalizing the details.

##### 5.2.2.3.3 Request for on demand system information

When acquiring an SI message, which according to the SystemInformationBlockType1 is indicated to be provided upon UE request, the UE shall:

1> if in RRC\_IDLE or in RRC\_INACTIVE:

2> if the [FFS\_Standalone] field is received in *SIB1*:

3> the UE shall trigger the lower layer to initiate the preamble transmission procedure in accordance with TS 38.321 [3] using the [indicated PRACH preamble] and [indicated PRACH resource];

3> if acknowledgement for SI request is received from lower layer;

4> acquire the requested SI message(s) as defined in sub-clause 5.2.2.3.2;

Editor’s Note: To be updated with details of the Msg1 request procedure.FFS\_Standalone

2> else

3> the UE shall trigger the lower layer to initiate the random access procedure in accordance with TS 38.321 [3];

3> if acknowledgement for SI request is received;

4> acquire the requested SI message(s) as defined in sub-clause 5.2.2.3.2;

Editor’s Note: To be updated with details of the Msg3 request procedure. FFS\_Standalone

1> else (in RRC\_CONNECTED):

2> [details FFS\_Standalone]

Editor’s Note: To be updated with details of the on-demand request procedure in RRC\_CONNECTED. FFS\_Standalone

Editor’s Note: [FFS\_Standalone if there is a need for a separate sub-clause to describe case where on demand SI is not successfully received by the UE and where it should initiate a new request]

#### 5.2.2.4 Actions upon receipt of SI message

##### 5.2.2.4.1 Actions upon reception of the *MIB*

Upon receiving the *MIB* the UE shall:

1> store the acquired *MIB*;

1> if the UE is in RRC\_IDLE or if the UE is in RRC\_INACTIVE or if the UE is in RRC\_CONNECTED while *T311* is running: [FFS]

2> if the *cellBarred* in the acquired MIB is set to *barred*;

3> consider the cell as barred in accordance with TS 38.304 [FFS];

2> else,

3> apply the received parameter(s) [FFS] to acquire *SIB1*.

##### 5.2.2.4.2 Actions upon reception of the SystemInformationBlockType1

Upon receiving the SystemInformationBlockType1 the UE shall:

1> store the acquired *SIB1*;

1> if the UE has a stored valid version of the required SIB(s) associated with the *systemInfoAreaIdentifier* and *systemInfoValueTag*/*systemInfoConfigurationIndex* in the acquired *SIB1*:

2> use that stored version of the SIB;

1> else if the *SIB1* message indicates that the SI message(s) is only provided on request:

2> trigger a request to acquire the SI message(s) (if needed) as defined in sub-clause 5.2.2.3;

1> else:

2> acquire the SI message(s) (if needed) as defined in sub-clause 5.2.2.3.2, which are provided according to the schedulingInfoList in the SystemInformationBlockType1;

Editor’s Note: [FFS\_Standalone Whether there is an additional indication that an on-demand SI is actually being broadcast at this instant in time]

Editor’s Note: To be updated when content of the SystemInformationBlockType1 has been agreed. FFS\_Standalone.

Editor’s Note: To be updated how to capture the UE behaviour when some required SIBs are from broadcast and other required SIBs through SI request.

##### 5.2.2.4.3 Actions upon reception of SystemInformationBlockTypeX

Editor’s Note: To be extended with further sub-clauses as more SIBs are defined. FFS\_Standalone

#### 5.2.2.5 Essential system information missing

The UE shall:

1> if in RRC\_IDLE or in RRC\_INACTIVE:

2> if the UE is unable to acquire the *MIB*; or

2> if the UE is unable to acquire the *SIB1* and UE does not have a stored valid version of SIB1; or

2> [FFS\_Standalone if the UE is unable to acquire the [FFS essential SystemInformationBlockTypeX] and UE does not have a stored valid version of SystemInformationBlockTypeX];

3> consider the cell as barred in accordance with TS 38.304 [X]; and

3> perform barring as if *intraFreqReselection* is set to *allowed*;

Editor’s Note: [FFS\_Standalone on details of RRC connection re-establishment procedure and corresponding reading of SI in RRC\_CONNECTED].

Editor’s Note: [FFS\_Standalone whether all the information needed to access the cell is included in SIB1 or if both SIB1 and SIB2 are essential in NR].

## 5.3 Connection control

Editor's note: FFS The structure and content of this subclause is a subject for discussion, e.g. potential merging of connection establishment and re-establishment messages, mobility aspects etc.

### 5.3.1 Introduction

### 5.3.2 Paging

Editor’s Note: Targeted for completion in June 2018.

### 5.3.3 RRC connection establishment

Editor’s Note: Targeted for completion in June 2018.

### 5.3.4 Initial security activation

Editor’s Note: Targeted for completion in June 2018.

### 5.3.5 RRC reconfiguration

#### 5.3.5.1 General



Figure 5.3.5.1-1: RRC reconfiguration, successful



Figure 5.3.5.1-2: RRC reconfiguration, failure

The purpose of this procedure is to modify an RRC connection, e.g. to establish/modify/release RBs, to perform reconfiguration with sync, to setup/modify/release measurements, to add/modify/release SCells and cell groups. As part of the procedure, NAS dedicated information may be transferred from the Network to the UE.

In EN-DC, SRB3 can be used to configure measurements, MAC, RLC, PDCP, physical layer and RLF timers and constants.

#### 5.3.5.2 Initiation

The Network may initiate the RRC reconfiguration procedure to a UE in RRC\_CONNECTED. The Network applies the procedure as follows:

- the establishment of RBs (other than SRB1, that is established during RRC connection establishment) is performed only when AS security has been activated;

- the addition of Secondary Cell Group and SCells is performed only when AS security has been activated;

- the *reconfigurationWithSync* is included in *secondaryCellGroup* only when at least one DRB is setup in SCG;

#### 5.3.5.3 Reception of an *RRCReconfiguration* by the UE

The UE shall perform the following actions upon reception of the *RRCReconfiguration*:

1> if the *RRCReconfiguration* includes the *secondaryCellGroup*:

2> perform the cell group configuration for the SCG according to 5.3.5.5;

1> if the *RRCReconfiguration* message contains the *radioBearerConfig*:

2> perform the radio bearer configuration according to 5.3.5.6;

1> if the *RRCReconfiguration* message includes the *measConfig*:

2> perform the measurement configuration procedure as specified in 5.5.2;

1> if the UE is configured with E-UTRA *nr-SecondaryCellGroupConfig* (MCG is E-UTRA):

2> if *RRCReconfiguration* was received via SRB1:

3> construct *RRCReconfigurationComplete* message and submit it via the EUTRA MCG embedded in E-UTRA RRC message *RRCConnectionReconfigurationComplete* as specified in TS 36.331 [10].

3> if *reconfigurationWithSync* was included in *spCellConfig* of an SCG:

4> initiate the random access procedure on the SpCell, as specified in TS 38.321 [3];

2> else (*RRCReconfiguration* was received via SRB3):

3> submit the *RRCReconfigurationComplete* message via SRB3 to lower layers for transmission using the new configuration;

NOTE: In the case of SRB1, the random access is triggered by RRC layer itself as there is not necessarily other UL transmission. In the case of SRB3, the random access is triggered by the MAC layer due to arrival of *RRCReconfigurationComplete*.

1> if MAC of an NR cell group successfully completes a random access procedure triggered above;

2> stop timer T304 for that cell group;

2> apply the parts of the CQI reporting configuration, the scheduling request configuration and the sounding RS configuration that do not require the UE to know the SFN of the respective target SpCell, if any;

2> apply the parts of the measurement and the radio resource configuration that require the UE to know the SFN of the respective target SpCell (e.g. measurement gaps, periodic CQI reporting, scheduling request configuration, sounding RS configuration), if any, upon acquiring the SFN of that target SpCell;

2> the procedure ends;

#### 5.3.5.4 Secondary cell group release

The UE shall:

1> as a result of SCG release triggered by E-UTRA:

2> reset SCG MAC, if configured;

2> for each RLC bearer that is part of the SCG configuration:

3> perform RLC bearer release procedure as specified in 5.3.5.5.3;

2> release the SCG configuration;

2> stop timer T310 for the corresponding SpCell, if running;

2> stop timer T304 for the corresponding SpCell, if running;

NOTE: Release of cell group means only release of the lower layer configuration of the cell group but the *RadioBearerConfig* may not be released.

#### 5.3.5.5 Cell Group configuration

##### 5.3.5.5.1 General

The network configures the UE with one Secondary Cell Group (SCG). For EN-DC, the MCG is configured as specified in TS 36.331 [10]. The network provides the configuration parameters for a cell group in the *CellGroupConfig* IE.

The UE performs the following actions based on a received *CellGroupConfig* IE:

1> if the *CellGroupConfig* contains the *spCellConfig* with *reconfigurationWithSync*:

2> perform Reconfiguration with sync according to 5.3.5.5.2;

2> resume all suspended radio bearers and resume SCG transmission for all radio bearers, if suspended;

1> if the *CellGroupConfig* contains the *rlc-BearerToReleaseList*:

2> perform RLC bearer release as specified in 5.3.5.5.3;

1> if the *CellGroupConfig* contains the *rlc-BearerToAddModList*:

2> perform the RLC bearer addition/modification as specified in 5.3.5.5.4;

1> if the *CellGroupConfig* contains the *mac-CellGroupConfig*:

2> configure the MAC entity of this cell group as specified in 5.3.5.5.5;

1> if the *CellGroupConfig* contains the *sCellToReleaseList*:

2> perform SCell release as specified in 5.3.5.5.8;

1> if the *CellGroupConfig* contains the spCellConfig:

2> configure the SpCell as specified in 5.3.5.5.7;

1> if the *CellGroupConfig* contains the *sCellToAddModList*:

2> perform SCell addition/modification as specified in 5.3.5.5.9;

##### 5.3.5.5.2 Reconfiguration with sync

The UE shall perform the following actions to execute a reconfiguration with sync.

1> stop timer T310 for the corresponding SpCell, if running;

1> start timer T304 for the corresponding SpCell with the timer value set to *t304*, as included in the *reconfigurationWithSync*;

1> if the *frequencyInfoDL* is included:

2> consider the target SpCell to be one on the frequency indicated by the *frequencyInfoDL* with a physical cell identity indicated by the *physCellId*;

1> else:

2> consider the target SpCell to be one on the frequency of the source SpCell with a physical cell identity indicated by the *physCellId*;

1> start synchronising to the DL of the target SpCell and acquire the *MIB* of the target SpCell as specified in 5.2.2.3.1;

NOTE X: The UE should perform the reconfiguration with sync as soon as possible following the reception of the RRC message triggering the reconfiguration with sync, which could be before confirming successful reception (HARQ and ARQ) of this message.

1> reset the MAC entity of this cell group;

1> consider the SCell(s) of this cell group, if configured, to be in deactivated state;

1> apply the value of the *newUE-Identity* as the C-RNTI for this cell group;

Editor’s Note: Verify that this does not configure some common parameters which are later discarded due to e.g. SCell release or due to LCH release.

1> configure lower layers in accordance with the received s*pCellConfigCommon*;

1> consider the initial bandwidth part to be the active bandwidth part where random access is performed;

1> configure lower layers in accordance with any additional fields, not covered in the previous, if included in the received *reconfigurationWithSync*;

##### 5.3.5.5.3 RLC bearer release

The UE shall:

1> for each *logicalChannelIdentity* value included in the *rlc-BearerToReleaseList* that is part of the current UE configuration (LCH release); or

1> for each *logicalChannelIdentity* value that is to be released as the result of an SCG release according to 5.3.5.4:

2> release the RLC entity or entities (includes discarding all pending RLC PDUs and RLC SDUs);

2> release the corresponding logical channel.

##### 5.3.5.5.4 RLC bearer addition/modification

For each *RLC-Bearer-Config* received in the *rlc-BearerToAddModList* IE the UE shall:

1> if the UE’s current configuration contains a RLC bearer with the received *logicalChannelIdentity*:

2> if *reestablishRLC* is received:

3> re-establish the RLC entity as specified in TS 38.322 [4];

2> reconfigure the RLC entity or entities in accordance with the received *rlc-Config*;

2> reconfigure the logical channel in accordance with the received *mac-LogicalChannelConfig*;

NOTE: The network does not re-associate an already configured logical channel with another radio bearer. Hence *servedRadioBearer* is not present in this case.

1> else (a logical channel with the given *logicalChannelIdentity* was not configured before):

2> if the *logicalChannelIdentity* corresponds to an SRB and *rlc-Config* is not included:

3> establish an RLC entity in accordance with the default configuration defined in 9.2 for the corresponding SRB;

2> else:

3> establish an RLC entity in accordance with the received *rlc-Config*;

2> if the *logicalChannelIdentity* corresponds to an SRB and if *mac-LogicalChannelConfig* is not included:

3> configure this MAC entity with a logical channel in accordance to the default configuration defined in 9.2 for the corresponding SRB;

2> else:

3> configure this MAC entity with a logical channel in accordance to the received *mac-LogicalChannelConfig*;

2> associate this logical channel with the PDCP entity identified by *servedRadioBearer*;

##### 5.3.5.5.5 MAC entity configuration

The UE shall:

1> if SCG MAC is not part of the current UE configuration (i.e. SCG establishment):

2> create an SCG MAC entity;

1> reconfigure the MAC main configuration of the cell group in accordance with the received *mac-CellGroupConfig* other than *tag-ToReleaseList* and *tag-ToAddModList*;

1> if the received *mac-CellGroupConfig* includes the *tag-ToReleaseList*:

2> for each *TAG-Id* value included in the *tag-ToReleaseList* that is part of the current UE configuration:

3> release the TAG indicated by *TAG-Id*;

1> if the received *mac-CellGroupConfig* includes the *tag-ToAddModList*:

2> for each *tag-Id* value included in *tag-ToAddModList* that is not part of the current UE configuration (TAG addition):

3> add the TAG, corresponding to the *tag-Id*, in accordance with the received *timeAlignmentTimer*;

2> for each *tag-Id* value included in *tag-ToAddModList* that is part of the current UE configuration (TAG modification):

3> reconfigure the TAG, corresponding to the *tag-Id*, in accordance with the received *timeAlignmentTimer*;

##### 5.3.5.5.6 RLF Timers & Constants configuration

The UE shall:

1> if the received *rlf-TimersAndConstants* is set to release:

NOTE: In EN-DC, *rlf-TimersAndConstants* cannot be released.

Editor’s Note: Standalone part to be complete by June 2018.

2> stop timer T310 for this cell group, if running, and

2> release the value of timer *t310* as well as constants *n310* and *n310* for this cell group;

1> else:

2> reconfigure the value of timers and constants in accordance with received *rlf-TimersAndConstants*;

##### 5.3.5.5.7 SPCell Configuration

The UE shall:

1> if the *SpCellConfig* contains the *rlf-TimersAndConstants*

2> configure the RLF timers and constants for this cell group as specified in 5.3.5.5.6;

1> if the *SpCellConfig* contains *spCellConfigDedicated*:

2> configure the SpCell in accordance with the *spCellConfigDedicated*;

##### 5.3.5.5.8 SCell Release

The UE shall:

1> if the release is triggered by reception of the *sCellToReleaseList*:

2> for each *sCellIndex* value included in the *sCellToReleaseList*:

3> if the current UE configuration includes an SCell with value *sCellIndex*:

4> release the SCell;

##### 5.3.5.5.9 SCell Addition/Modification

The UE shall:

1> for each *sCellIndex* value included in the *sCellToAddModList* that is not part of the current UE configuration (SCell addition):

2> add the SCell, corresponding to the *sCellIndex*, in accordance with the *sCellConfigCommon* and *sCellConfigDedicated*;

2> configure lower layers to consider the SCell to be in deactivated state;

Editor’s Note: FFS Check automatic measurement handling for SCells.

2> for each *measId* included in the *measIdList* within *VarMeasConfig*:

3> if SCells are not applicable for the associated measurement; and

3> if the concerned SCell is included in *cellsTriggeredList* defined within the *VarMeasReportList* for this *measId*:

4> remove the concerned SCell from *cellsTriggeredList* defined within the *VarMeasReportList* for this *measId*;

1> for each *sCellIndex* value included in the *sCellToAddModList* that is part of the current UE configuration (SCell modification):

2> modify the SCell configuration in accordance with the *sCellConfigDedicated*;

#### 5.3.5.6 Radio Bearer configuration

##### 5.3.5.6.1 General

The UE shall perform the following actions based on a received *RadioBearerConfig* IE:

1> if the *RadioBearerConfig* includes the *srb3-ToRelease* and set to true:

2> perform the SRB release as specified in 5.3.5.6.2;

1> if the *RadioBearerConfig* includes the *srb-ToAddModList*:

2> perform the SRB addition or reconfiguration as specified in 5.3.5.6.3;

1> if the *RadioBearerConfig* includes the *drb-ToReleaseList*:

2> perform DRB release as specified in 5.3.5.6.4;

1> if the *RadioBearerConfig* includes the *drb-ToAddModList*:

2> perform DRB addition or reconfiguration as specified in 5.3.5.6.5;

##### 5.3.5.6.2 SRB release

Editor’s note: FFS / TODO: check handling during full configuration

The UE shall:

1> release the PDCP entity of the SRB3.

##### 5.3.5.6.3 SRB addition/modification

The UE shall:

1> for each *srb-Identity* value included in the *srb-ToAddModList* that is not part of the current UE configuration (SRB establishment or reconfiguration from E-UTRA PDCP to NR PDCP):

2> establish a PDCP entity and configure it with the security algorithms according to *securityConfig* and apply the keys (KRRCenc and KRRCint) associated with the KeNB/S-KgNB as indicated in *keyToUse*, if applicable;

2> if the current UE configuration as configured by E-UTRA in TS 36.331 includes an SRB identified with the same *srb-Identity* value:

3> associate the E-UTRA RLC entity and DCCH of this SRB with the NR PDCP entity;

3> release the E-UTRA PDCP entity of this SRB;

2> if the *pdcp-Config* is included:

3> configure the PDCP entity in accordance with the received *pdcp-Config*;

2> else:

3> configure the PDCP entity in accordance with the default configuration defined in 9.2.1 for the corresponding SRB;

1> for each *srb-Identity* value included in the *srb-ToAddModList* that is part of the current UE configuration:

2> if *reestablishPDCP* is set:

3> configure the PDCP entity to apply the integrity protection algorithm and KRRCint key associated with the KeNB/S-KgNB as indicated in *keyToUse* , i.e. the integrity protection configuration shall be applied to all subsequent messages received and sent by the UE, including the message used to indicate the successful completion of the procedure;

3> configure the PDCP entity to apply the ciphering algorithm and KRRCenc key associated with the KeNB/S-KgNB as indicated in *keyToUse*, i.e. the ciphering configuration shall be applied to all subsequent messages received and sent by the UE, including the message used to indicate the successful completion of the procedure;

3> re-establish the PDCP entity of this SRB as specified in 38.323 [5];

2> else, if *discardOnPDCP* is set:

3> trigger the PDCP entity to perform SDU discard as specified in TS 38.323 [5];

2> if the *pdcp-Config* is included:

3> reconfigure the PDCP entity in accordance with the received *pdcp-Config*.

##### 5.3.5.6.4 DRB release

Editor’s Note: FFS / TODO: Add handling for the new QoS concept (mapping of flows; configuration of QFI-to-DRB mapping; reflective QoS…) but keep also EPS-Bearer handling for the EN-DC case

The UE shall:

1> for each *drb-Identity* value included in the *drb-ToReleaseList* that is part of the current UE configuration (DRB release), or

1> for each *drb-identity* value that is to be released as the result of full configuration option according to 5.3.5.7:

2> release the PDCP entity;

1> if a new bearer is not added either with NR or E-UTRA with same *eps-BearerIdentity*:

2> if the procedure was triggered due to reconfiguration with sync:

3> indicate the release of the DRB and the *eps-BearerIdentity* of the released DRB to upper layers after successful reconfiguration with sync;

2> else:

3> indicate the release of the DRB and the *eps-BearerIdentity* of the released DRB to upper layers immediately;

NOTE: The UE does not consider the message as erroneous if the *drb-ToReleaseList* includes any *drb-Identity* value that is not part of the current UE configuration.

NOTE: Whether or not the RLC and MAC entities associated with this PDCP entity are reset or released is determined by the *CellGroupConfig*.

##### 5.3.5.6.5 DRB addition/modification

The UE shall:

1> for each *drb-Identity* value included in the *drb-ToAddModList* that is not part of the current UE configuration (DRB establishment including the case when full configuration option is used):

2> establish a PDCP entity and configure it in accordance with the received *pdcp-Config*;

2> configure the PDCP entity with the security algorithms according to *securityConfig* and apply the keys (KUPenc) associated with the KeNB/S-KgNB as indicated in *keyToUse*;

2> if DRB was configured with the same *eps-BearerIdentity* either by NR or E-UTRA prior to receiving this reconfiguration:

3> associate the established DRB with the corresponding *eps-BearerIdentity*

2> else:

3> indicate the establishment of the DRB(s) and the *eps-BearerIdentity* of the established DRB(s) to upper layers;

1> for each *drb-Identity* value included in the *drb-ToAddModList* that is part of the current UE configuration:

2> if *reestablishPDCP* is set:

3> configure the PDCP entity of this *RadioBearerConfig* to apply the ciphering algorithm and KUPenc key associated with the KeNB/S-KgNB as indicated in *keyToUse*, i.e. the ciphering configuration shall be applied to all subsequent PDCP PDUs received and sent by the UE;

3> re-establish the PDCP entity of this DRB as specified in 38.323 [5], section 5.1.2;

2> else, if *recoverPDCP* is set:

3> trigger the PDCP entity of this DRB to perform data recovery as specified in 38.323;

2> if the *pdcp-Config* is included:

3> reconfigure the PDCP entity in accordance with the received *pdcp-Config*;

NOTE: Removal and addition of the same *drb-Identity* in a single *radioResourceConfig* is not supported. In case *drb-Identity* is removed and added due to reconfiguration with sync or re-establishment with the full configuration option, the network can use the same value of *drb-Identity*.

NOTE: When determining whether a drb-Identity value is part of the current UE configuration, the UE does not distinguish which *RadioBearerConfig* and *DRB-ToAddModList* that DRB was originally configured in. To re-associate a DRB with a different key (KeNB to S-KeNB or vice versa), the network provides the *drb-Identity* value in the (target) *drb-ToAddModList* and sets the *reestablishPDCP* flag. The network does not list the *drb-Identity* in the (source) *drb-ToReleaseList*.

NOTE: When setting the *reestablishPDCP* flag for a radio bearer, the network ensures that the RLC receiver entities do not deliver old PDCP PDUs to the re-established PDCP entity. It does that e.g. by triggering a reconfiguration with sync of the cell group hosting the old RLC entity or by releasing the old RLC entity.

NOTE: In this specification, UE configuration refers to the parameters configured by NR RRC unless otherwise stated.

#### 5.3.5.8 Security key update

Upon reception of *sk-Counter* as specified in TS 36.331 [10] the UE shall:

1> update the S-KgNB key based on the KeNB key and using the received *sk-Counter* value, as specified in TS 33.501 [11];

1> derive KRRCenc and KUPenc key as specified in TS 33.501 [11];

1> derive the KRRCint and KUPint key as specified in TS 33.501 [11];

#### 5.3.5.9 Reconfiguration failure

##### 5.3.5.9.1 Integrity check failure

Editor’s Note: Removed “SIB3” from heading so that this sub-section can easily be expanded to stand-alone case (if considered necessary). FFS\_Standalone

The UE shall:

1> upon integrity check failure indication from NR lower layers for SRB3:

2> initiate the SCG failure information procedure as specified in subclause 5.7.3 to report SRB3 integrity check failure;

##### 5.3.5.9.2 Inability to comply with RRCReconfiguration

The UE shall:

1> if the UE is operating in EN-DC:

2> if the UE is unable to comply with (part of) the configuration included in the *RRCReconfiguration* message received over SRB3;

3> continue using the configuration used prior to the reception of *RRCReconfiguration* message;

3> initiate the SCG failure information procedure as specified in subclause 5.7.3 to report SCG reconfiguration error, upon which the connection reconfiguration procedure ends;

2> else, if the UE is unable to comply with (part of) the configuration included in the *RRCReconfiguration* message received over MCG SRB1;

3> continue using the configuration used prior to the reception of *RRCReconfiguration* message;

3> initiate the connection re-establishment procedure as specified in TS 36.331 [10, 5.3.7], upon which the connection reconfiguration procedure ends;

NOTE 1: The UE may apply above failure handling also in case the *RRCReconfiguration* message causes a protocol error for which the generic error handling as defined in 10 specifies that the UE shall ignore the message.

NOTE 2: If the UE is unable to comply with part of the configuration, it does not apply any part of the configuration, i.e. there is no partial success/failure.

##### 5.3.5.9.3 T304 expiry (Reconfiguration with sync Failure)

The UE shall:

1> if T304 of a secondary cell group expires:

2> release *rach-ConfigDedicated*;

2> initiate the SCG failure information procedure as specified in subclause 5.7.3 to report SCG reconfiguration with sync failure;

#### 5.3.5.9 Other configuration

Editor’s Note: Targeted for completion in June 2018.

#### 5.3.5.10 EN-DC release

The UE shall:

1> as a result of EN-DC release triggered by E-UTRA:

2> release SRB3 (configured according to *radioBearerConfig*), if present;

2> release *measConfig*;

2> release the SCG configuration as specified in section 5.3.5.4.

### 5.3.6 Counter check

FFS

### 5.3.7 RRC connection re-establishment

Editor’s Note: Targeted for completion in June 2018.

### 5.3.8 RRC connection release

Editor’s Note: Targeted for completion in June 2018.

### 5.3.9 RRC connection release requested by upper layers

Editor’s Note: Targeted for completion in June 2018.

### 5.3.10 Radio link failure related actions

#### 5.3.10.1 Detection of physical layer problems in RRC\_CONNECTED

The UE shall:

1> upon receiving N310 consecutive "out-of-sync" indications for the SpCell from lower layers while T311 is not running:

2> start timer T310 for the corresponding SpCell;

Editor’s Note: FFS: Under which condition physical layer problems detection is performed, e.g. neither T300, T301, T304 nor T311 is running. It’s subject to the harmonization of the RRC procedures for RRC Connection establishment/resume/re-establishment and RRC connection reconfiguration.

#### 5.3.10.2 Recovery of physical layer problems

Upon receiving N311 consecutive "in-sync" indications for the SpCell from lower layers while T310 is running, the UE shall:

1> stop timer T310 for the corresponding SpCell;

NOTE 1: In this case, the UE maintains the RRC connection without explicit signalling, i.e. the UE maintains the entire radio resource configuration.

NOTE 2: Periods in time where neither "in-sync" nor "out-of-sync" is reported by layer 1 do not affect the evaluation of the number of consecutive "in-sync" or "out-of-sync" indications.

#### 5.3.10.3 Detection of radio link failure

The UE shall:

1> upon T310 expiry in PCell; or

1> upon random access problem indication from MCG MAC while T311 is not running; or

Editor’s Note: FFS: Under which condition physical layer problems detection is performed, e.g. neither T300, T301, T304 nor T311 is running. It’s subject to the harmonization of the RRC procedures for RRC Connection establishment/resume/re-establishment and RRC connection reconfiguration.

1> upon indication from MCG RLC that the maximum number of retransmissions has been reached:

Editor’s Note: FFS whether maximum ARQ retransmission is only criteria for RLC failure.

2> consider radio link failure to be detected for the MCG i.e. RLF;

Editor’s Note: FFS Whether indications related to beam failure recovery may affect the declaration of RLF.

Editor’s Note: FFS: How to handle RLC failure in CA duplication for MCG DRB and SRB.

Editor’s Note: FFS: RLF related measurement reports e.g. *VarRLF-Report* is supported in NR.

2> if AS security has not been activated:

3> perform the actions upon leaving RRC\_CONNECTED as specified in x.x.x FFS\_Ref, with release cause 'other';

2> else:

3> initiate the connection re-establishment procedure as specified in x.x.x FFS\_Ref.

The UE shall:

1> upon T310 expiry in PSCell; or

1> upon random access problem indication from SCG MAC; or

1> upon indication from SCG RLC that the maximum number of retransmissions has been reached:

2> consider radio link failure to be detected for the SCG i.e. SCG-RLF;

Editor’s Note: FFS: How to handle RLC failure in CA duplication for SCG DRB and SRB.

2> initiate the SCG failure information procedure as specified in 5.7.3 to report SCG radio link failure;

### 5.3.11 UE actions upon leaving RRC\_CONNECTED

Editor’s Note: Targeted for completion in June 2018.

### 5.3.12 UE actions upon PUCCH/SRS release request

Upon receiving a PUCCH release request from lower layers, for all bandwidth parts of an indicated serving cell the UE shall:

1. release *PUCCH-CSI-Resources* configured in *CSI-ReportConfig*;

1> release *SchedulingRequestResourceConfig* instances configured in *PUCCH-Config*;

Upon receiving an SRS release request from lower layers, for all bandwidth parts of an indicated serving cell the UE shall:

1. release *SRS-Resource* instances configured in *SRS-Config*;

## 5.4 Inter-RAT mobility

Editor’s Note: Targeted for completion in June 2018.

### 5.7.3 SCG failure information

#### 5.7.3.1 General



Figure 5.6.13.1-1: SCG failure information

The purpose of this procedure is to inform EUTRAN or NR MN about an SCG failure the UE has experienced i.e. SCG radio link failure, e failure of SCG reconfiguration with synch, SCG configuration failure for RRC message on SRB3, SCG integrity check failure and exceeding the maximum uplink transmission timing difference.

Editor’s Note: SCG failure considers the case of exceeding the maximum uplink transmission timing difference if RAN1 decides that EN-DC supports the synchronised operation case. FFS how to capture

Editor’s Note: FFS whether to include the handling of SCell Failure in CA duplication case in SCGfailureinformation procedure and whether to rename SCGfailureinformation.

#### 5.7.3.2 Initiation

A UE initiates the procedure to report SCG failures when SCG transmission is not suspended and when one of the following conditions is met:

1> upon detecting radio link failure for the SCG, in accordance with subclause 5.3.10.3;

1> upon reconfiguration with sync failure of the SCG, in accordance with subclause 5.3.5.9.3;

1> upon SCG configuration failure, in accordance with subclause 5.3.5.9.2;

1> upon integrity check failure indication from SCG lower layers, in accordance with subclause 5.3.5.9.1;

Upon initiating the procedure, the UE shall:

1> suspend SCG transmission for all SRBs and DRBs;

1> reset SCG-MAC;

1> stop T304, if running;

1> if the UE is operating in EN-DC:

2> initiate transmission of the *SCGFailureInformationNR* message as specified in TS 36.331 [10, 5.6.13a];

Editor’s Note:
The section for transmission of SCGFailureInformation in NR RRC entity for SA is FFS\_Standalone.

#### 5.7.3.3 Failure type determination

Editor’s Note: FFS / TODO: Either use this section also for NR-DC or change section title (add “for EN-DC”)

The UE shall set the SCG failure type as follows:

1> if the UE initiates transmission of the *SCGFailureInformation* message to provide SCG radio link failure information:

2> set the *failureType* as *scg-RadioLinkFailure*;

1> else if the UE initiates transmission of the *SCGFailureInformationNR* message to provide reconfiguration with sync failure information for an SCG:

2> set the *failureType* as *scg-ChangeFailure*;

Editor’s Note: FFS whether to change scg-ChangeFailure to synchronousReconfigurationFailure-SCG

1> else, if the UE initiates transmission of the *SCGFailureInformationNR* message due to SRB3 IP check failure:

2> set the *failureType* as *srb3-IntegrityFailure*;

1> else, if the UE initiates transmission of the *SCGFailureInformationNR* message due to Reconfiguration failure of NR RRC reconfiguration message:

2> set the *failureType* as *scg-reconfigFailure*;

Editor’s Note: FFS: whether to include *rrc-TransactionIdentifier* information.

#### 5.7.3.4 Setting the contents of *MeasResultSCG-Failure*

The UE shall set the contents of the *MeasResultSCG-Failure* as follows:

1> set the *measResultServFreqList* to include for each SCG cell that is configured by the SN to be measured, if any, within *measResultServingCell* the quantities of the concerned SCell, if available, according to performance requirements in [FFS\_Ref];

1> for each SCG serving frequency included in *measResultServFreqList* include within *measResultBestNeighCell* the *physCellId* and the quantities (including both available cell level and beam level measurement results) of the best non-serving cell, based on RSRP, on the concerned serving frequency, as follows:

2> if RSRP is available:

3> consider RSRP as the sorting quantity;

2> else if RSRQ is available:

3> consider RSRQ as the sorting quantity;

2> else

3> consider SINR as the sorting quantity;

1> set the *measResultNeighCells* to include the best measured cells on non-serving NR frequencies, ordered such that the best cell is listed first, and based on measurements collected up to the moment the UE detected the failure, and set its fields as follows;

2> if RSRP is available:

 3> consider RSRP as the sorting quantity;

2> else if RSRQ is available:

 3> consider RSRQ as the sorting quantity;

 2> else

 3> consider SINR as the sorting quantity;

2> if the UE was configured to perform measurements by the SN for one or more non-serving NR frequencies and measurement results are available, include the *measResultListNR*;

2> for each neighbour cell included:

3> include the optional fields that are available;

NOTE: The measured quantities are filtered by the L3 filter as configured in the mobility measurement configuration. The measurements are based on the time domain measurement resource restriction, if configured. Blacklisted cells are not required to be reported.

# 6 Protocol data units, formats and parameters (ASN.1)

## 6.1 General

### 6.1.1 Introduction

The contents of each RRC message is specified in sub-clause 6.2 using ASN.1 to specify the message syntax and using tables when needed to provide further detailed information about the fields specified in the message syntax. The syntax of the information elements that are defined as stand-alone abstract types is further specified in a similar manner in sub-clause 6.3.

### 6.1.2 Need codes and conditions for optional downlink fields

The need for fields to be present in a message or an abstract type, i.e., the ASN.1 fields that are specified as OPTIONAL in the abstract notation (ASN.1), is specified by means of comment text tags attached to the OPTIONAL statement in the abstract syntax. All comment text tags are available for use in the downlink direction only. The meaning of each tag is specified in table 6.1-1.

If conditions are used, a conditional presence table is provided for the message or information element specifying the need of the field for each condition case. The table also specifies whether UE maintains or releases the value in case the field is not present. The conditions clarify what the UE may expect regarding the setting of the message by the network. Violation of conditions is regarded as invalid network behaviour, which the UE is not required to cope with. Hence the general error handling defined in 10.4 does not apply in case a field is absent although it is mandatory according to the CondC or CondM condition.

For guidelines on the use of need codes and conditions, see Annex A.6 and A.7.

Table 6.1-1: Meaning of abbreviations used to specify the need for fields to be present

| Abbreviation | Meaning |
| --- | --- |
|  |  |
| CondC conditionTag | Configuration conditionPresence of the field is conditional to other configuration settings. |
| CondM conditionTag | Message conditionPresence of the field is conditional to other fields included in the message. |
| Need S | *Specified*Used for (configuration) fields, whose field description or procedure **specifies** the UE behavior performed upon receiving a message with the field absent (and not if field description or procedure specifies the UE behavior when field is not configured). |
| Need M | *Maintain*Used for (configuration) fields that are stored by the UE i.e. not one-shot. Upon receiving a message with the field absent, the UE maintains the current value. |
| Need N | *No action* (one-shot configuration that is not maintained)Used for (configuration) fields that are not stored and whose presence causes a one-time action by the UE. Upon receiving message with the field absent, the UE takes no action. |
| Need R | *Release*Used for (configuration) fields that are stored by the UE i.e. not one-shot. Upon receiving a message with the field absent, the UE releases the current value. |

## 6.2 RRC messages

### 6.2.1 General message structure

#### *– NR-RRC-Definitions*

This ASN.1 segment is the start of the NR RRC PDU definitions.

-- ASN1START

-- TAG-NR-RRC-DEFINITIONS-START

NR-RRC-Definitions DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- TAG-NR-RRC-DEFINITIONS-STOP

-- ASN1STOP

#### *– BCCH-BCH-Message*

The *BCCH-BCH-Message* class is the set of RRC messages that may be sent from the network to the UE via BCH on the BCCH logical channel.

-- ASN1START

-- TAG-BCCH-BCH-MESSAGE-START

BCCH-BCH-Message ::= SEQUENCE {

 message BCCH-BCH-MessageType

}

BCCH-BCH-MessageType ::= CHOICE {

 mib MIB,

 messageClassExtension SEQUENCE {}

}

-- TAG-BCCH-BCH-MESSAGE-STOP

-- ASN1STOP

#### *– DL-DCCH-Message*

The *DL-DCCH-Message* class is the set of RRC messages that may be sent from the network to the UE on the downlink DCCH logical channel.

-- ASN1START

-- TAG-DL-DCCH-MESSAGE-START

DL-DCCH-Message ::= SEQUENCE {

 message DL-DCCH-MessageType

}

DL-DCCH-MessageType ::= CHOICE {

 c1 CHOICE {

 rrcReconfiguration RRCReconfiguration,

 spare15 NULL, spare14 NULL, spare13 NULL,

 spare12 NULL, spare11 NULL, spare10 NULL,

 spare9 NULL, spare8 NULL, spare7 NULL,

 spare6 NULL, spare5 NULL, spare4 NULL,

 spare3 NULL, spare2 NULL, spare1 NULL

 },

 messageClassExtension SEQUENCE {}

}

-- TAG-DL-DCCH-MESSAGE-STOP

-- ASN1STOP

#### *– UL-DCCH-Message*

The *UL-DCCH-Message* class is the set of RRC messages that may be sent from the UE to the network on the uplink DCCH logical channel.

-- ASN1START

-- TAG-UL-DCCH-MESSAGE-START

UL-DCCH-Message ::= SEQUENCE {

 message UL-DCCH-MessageType

}

UL-DCCH-MessageType ::= CHOICE {

 c1 CHOICE {

 measurementReport MeasurementReport,

 rrcReconfigurationComplete RRCReconfigurationComplete,

 spare14 NULL, spare13 NULL, spare12 NULL,

 spare11 NULL, spare10 NULL, spare9 NULL,

 spare8 NULL, spare7 NULL, spare6 NULL,

 spare5 NULL, spare4 NULL, spare3 NULL,

 spare2 NULL, spare1 NULL

 },

 messageClassExtension SEQUENCE {}

}

-- TAG-UL-DCCH-MESSAGE-STOP

-- ASN1STOP

### 6.2.2 Message definitions

#### – *MIB*

The *MIB* includes the system information transmitted on BCH.

Signalling radio bearer: N/A

RLC-SAP: TM

Logical channel: BCCH

Direction: Network to UE

*MIB*

-- ASN1START

-- TAG-MIB-START

MIB ::= SEQUENCE {

 -- The 6 most significant bit (MSB) of the 10 bit System Frame Number. The 4 LSB of the SFN are conveyed in the PBCH transport block

 -- as well but outside the MIB.

 systemFrameNumber BIT STRING (SIZE (6)),

 -- Subcarrier spacing for SIB1, Msg.2/4 for initial access and broadcast SI-messages.

 -- If the UE acquires this MIB on a carrier frequency <6GHz, the values 15 and 30 kHz are applicable.

 -- If the UE acquires this MIB on a carrier frequency >6GHz, the values 60 and 120 kHz are applicable.

 subCarrierSpacingCommon ENUMERATED {scs15or60, scs30or120},

 -- The frequency domain offset between SSB and the overall resource block grid in number of subcarriers. (See 38.211, section 7.4.3.1)

 -- Note: For frequencies <6 GHz a fith, this field may comprise only the 4 least significant bits of the ssb-SubcarrierOffset.

 ssb-SubcarrierOffset INTEGER (0..15),

 -- Position of (first) DL DM-RS. Corresponds to L1 parameter 'DL-DMRS-typeA-pos' (see 38.211, section 7.4.1.1.1)

 dmrs-TypeA-Position ENUMERATED {pos2, pos3},

 -- Determines a bandwidth for PDCCH/SIB, a common ControlResourceSet (CORESET) a common search space and necessary PDCCH parameters.

 -- The codepoint "FFS\_RAN1" indicates that this cell does not provide SIB1 and that there is hence no common CORESET.

 -- Corresponds to L1 parameter 'RMSI-PDCCH-Config' (see FFS\_Specification, section FFS\_Section)

 pdcch-ConfigSIB1 INTEGER (0..255),

 -- Indicates that UE shall not camp on this cell

 cellBarred ENUMERATED {barred, notBarred},

 -- Controls cell reselection to intra-frequency cells when the highest ranked cell is barred, or treated as barred by the UE,

 -- as specified in TS 38.304.

 intraFreqReselection ENUMERATED {allowed, notAllowed},

 spare BIT STRING (SIZE (1))

}

-- TAG-MIB-STOP

-- ASN1STOP

| *MIB* field descriptions |
| --- |
|  |

#### – *MeasurementReport*

The *MeasurementReport* message is used for the indication of measurement results.

Signalling radio bearer: SRB1, SRB3

RLC-SAP: AM

Logical channel: DCCH

Direction: UE to Network

*MeasurementReport message*

-- ASN1START

-- TAG-MEASUREMENTREPORT-START

MeasurementReport ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 measurementReport MeasurementReport-IEs,

 criticalExtensionsFuture SEQUENCE {}

 }

}

MeasurementReport-IEs ::= SEQUENCE {

 measResults MeasResults,

 lateNonCriticalExtension OCTET STRING OPTIONAL,

 nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-MEASUREMENTREPORT-STOP

-- ASN1STOP

#### – *RRCReconfiguration*

The *RRCReconfiguration* message is the command to modify an RRC connection. It may convey information for measurement configuration, mobility control, radio resource configuration (including RBs, MAC main configuration and physical channel configuration) including and security configuration.

Signalling radio bearer: SRB1 or SRB3

RLC-SAP: AM

Logical channel: DCCH

Direction: Network to UE

*RRCReconfiguration message*

-- ASN1START

-- TAG-RRCRECONFIGURATION-START

RRCReconfiguration ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 rrcReconfiguration RRCReconfiguration-IEs,

 criticalExtensionsFuture SEQUENCE {}

 }

}

RRCReconfiguration-IEs ::= SEQUENCE {

 -- Configuration of Radio Bearers (DRBs, SRBs) including SDAP/PDCP.

 -- In EN-DC this field may only be present if the RRCReconfiguration

 -- is transmitted over SRB3.

 radioBearerConfig RadioBearerConfig OPTIONAL, -- Need M

 -- Configuration of secondary cell group (EN-DC):

 secondaryCellGroup OCTET STRING OPTIONAL, -- Need M

 measConfig MeasConfig OPTIONAL, -- Need M

 lateNonCriticalExtension OCTET STRING OPTIONAL,

 nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-RRCRECONFIGURATION-STOP

-- ASN1STOP

| *RRCReconfiguration* field descriptions |
| --- |
| ***secondaryCellGroup***Includes *CellGroupConfig* IE for the secondary cell group inside the OCTET STRING. |

| Conditional presence | Explanation |
| --- | --- |
| *FFS* | FFS |

#### *– RRCReconfigurationComplete*

The *RRCReconfigurationComplete* message is used to confirm the successful completion of an RRC connection reconfiguration.

Signalling radio bearer: SRB1 or SRB3

RLC-SAP: AM

Logical channel: DCCH

Direction: UE to Network

*RRCReconfigurationComplete message*

-- ASN1START

-- TAG-RRCRECONFIGURATIONCOMPLETE-START

RRCReconfigurationComplete ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 rrcReconfigurationComplete RRCReconfigurationComplete-IEs,

 criticalExtensionsFuture SEQUENCE {}

 }

}

RRCReconfigurationComplete-IEs ::= SEQUENCE {

 lateNonCriticalExtension OCTET STRING OPTIONAL,

 nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-RRCRECONFIGURATIONCOMPLETE-STOP

-- ASN1STOP

| *RRCReconfigurationComplete* field descriptions |
| --- |
| ***FFS***FFS |

#### – *SIB1*

Editor’s Note: Discuss whether to keep SIB1 for the December version. FFS

*SIB1* contains information relevant when evaluating if a UE is allowed to access a cell and defines the scheduling of other system information.It also contains radio resource configuration information that is common for all UEs.

Signalling radio bearer: N/A

RLC-SAP: TM

Logical channels: BCCH and BR-BCCH

Direction: Network to UE

*SIB1 message*

-- ASN1START

-- TAG-SIB1-START

SIB1 ::= SEQUENCE {

 -- FFS / TODO: Add other parameters.

 -- Frequency offset for the SSB of -5kHz (M=-1) or +5kHz (M=1). When the field is absent, the UE applies no offset (M=0).

 -- The offset is only applicable for the frequency range 0-2.65GHz. Corresponds to parameter 'M' (see 38.101, section FFS\_Section)

 frequencyOffsetSSB FrequencyOffsetSSB OPTIONAL, -- Need R

 -- Time domain positions of the transmitted SS-blocks in an SS-Burst-Set (see 38.213, section 4.1)

 ssb-PositionsInBurst SEQUENCE {

 -- Indicates the presence of the up to 8 SSBs in one group

 inOneGroup BIT STRING (SIZE (8)),

 -- For above 6 GHz: indicates which groups of SSBs is present

 groupPresence BIT STRING (SIZE (8)) OPTIONAL -- Cond above6GHzOnly

 },

 -- The SSB periodicity in msec for the rate matching purpose (see 38.211, section [7.4.3.1])

 ssb-PeriodicityServingCell ENUMERATED {ms5, ms10, ms20, ms40, ms80, ms160, spare1, spare2},

 -- TX power that the NW used for SSB transmission. The UE uses it to estimate the RA preamble TX power.

 -- (see 38.213, section 7.4)

 ss-PBCH-BlockPower INTEGER (-60..50),

 uplinkConfigCommon UplinkConfigCommon OPTIONAL,

 -- FFS: How to indicate the FrequencyInfoUL for the SUL

 supplementaryUplink SEQUENCE {

 uplinkConfigCommon UplinkConfigCommon OPTIONAL

 -- FFS: Add additional (selection) criteria determining when/whether the UE shall use the SUL frequency

 } OPTIONAL, -- Cond SUL

 tdd-UL-DL-Configuration TDD-UL-DL-ConfigCommon OPTIONAL, -- Cond TDD

 tdd-UL-DL-configurationCommon2 TDD-UL-DL-ConfigCommon OPTIONAL, -- Cond TDD

 pdcch-ConfigCommon PDCCH-ConfigCommon OPTIONAL,

 pucch-ConfigCommon PUCCH-ConfigCommon OPTIONAL,

 lateNonCriticalExtension OCTET STRING OPTIONAL,

 nonCriticalExtension SEQUENCE{} OPTIONAL

}

-- TAG-SIB1-STOP

-- ASN1STOP

## 6.3 RRC information elements

### 6.3.0 Parameterized types

### – SetupRelease Information Element

*SetupRelease* allows the *ElementTypeParam* to be used as the referenced data type for the setup and release entries. See A.3.8 for guidelines.

-- ASN1START

-- TAG-SETUP-RELEASE-START

SetupRelease { ElementTypeParam } ::= CHOICE {

 release NULL,

 setup ElementTypeParam

}

-- TAG-SETUP-RELEASE-STOP

-- ASN1STOP

## 6.3 RRC information elements

### 6.3.0 Parameterized types

### – SetupRelease Information Element

*SetupRelease* allows the *ElementTypeParam* to be used as the referenced data type for the setup and release entries. See A.3.8 for guidelines.

-- ASN1START

-- TAG-SETUP-RELEASE-START

SetupRelease { ElementTypeParam } ::= CHOICE {

 release NULL,

 setup ElementTypeParam

}

-- TAG-SETUP-RELEASE-STOP

-- ASN1STOP

### 6.3.1 System information blocks

### 6.3.2 Radio resource control information elements

#### – *CellGroupConfig*

The *CellGroupConfig* IE is used to configure a master cell group (MCG) or secondary cell group (SCG). A cell group comprises of one MAC entity, a set of logical channels with associated RLC entities and of a primary cell (SpCell) and one or more secondary cells (SCells).

*CellGroupConfig* information element

-- ASN1START

-- TAG-CELL-GROUP-CONFIG-START

-- Configuration of one Cell-Group:

CellGroupConfig ::= SEQUENCE {

 cellGroupId CellGroupId,

 -- Logical Channel configuration and association with radio bearers:

 rlc-BearerToAddModList SEQUENCE (SIZE(1..maxLC-ID)) OF RLC-Bearer-Config OPTIONAL, -- Need N

 rlc-BearerToReleaseList SEQUENCE (SIZE(1..maxLC-ID)) OF LogicalChannelIdentity OPTIONAL, -- Need N

 -- Parameters applicable for the entire cell group:

 mac-CellGroupConfig MAC-CellGroupConfig OPTIONAL, -- Need M

 physicalCellGroupConfig PhysicalCellGroupConfig OPTIONAL, -- Need M

 -- Serving Cell specific parameters (SpCell and SCells)

 spCellConfig SpCellConfig OPTIONAL, -- Need M

 sCellToAddModList SEQUENCE (SIZE (1..maxNrofSCells)) OF SCellConfig OPTIONAL, -- Need N

 -- List of seconary serving cells to be released (not applicable for SpCells)

 sCellToReleaseList SEQUENCE (SIZE (1..maxNrofSCells)) OF SCellIndex OPTIONAL, -- Need N

 ...

}

-- The ID of a cell group. 0 identifies the master cell group. Other values identify secondary cell groups.

-- In this version of the specification only values 0 and 1 are supported.

-- FFS: This should be moved to be own IE section

CellGroupId ::= INTEGER (0.. maxSecondaryCellGroups)

RLC-Bearer-Config ::= SEQUENCE {

 -- ID used commonly for the MAC logical channel and for the RLC bearer.

 logicalChannelIdentity LogicalChannelIdentity,

 -- Associates the RLC Bearer with an SRB or a DRB. The UE shall deliver DL RLC SDUs received via the RLC entity of this

 -- RLC bearer to the PDCP entity of the servedRadioBearer. Furthermore, the UE shall advertise and deliver uplink PDCP PDUs of the

 -- uplink PDCP entity of the servedRadioBearer to the uplink RLC entity of this RLC bearer unless the uplink scheduling

 -- restrictions ('moreThanOneRLC' in PDCP-Config and the restrictions in LogicalChannelConfig) forbid it to do so.

 servedRadioBearer CHOICE {

 srb-Identity SRB-Identity,

 drb-Identity DRB-Identity

 } OPTIONAL, -- Cond LCH-SetupOnly

 reestablishRLC ENUMERATED {true} OPTIONAL, -- Need N

 rlc-Config RLC-Config OPTIONAL, -- Cond LCH-Setup

 mac-LogicalChannelConfig LogicalChannelConfig OPTIONAL, -- Cond LCH-Setup

 ...

}

LogicalChannelIdentity ::= INTEGER (1..maxLC-ID)

-- Cell-Group specific L1 parameters

PhysicalCellGroupConfig ::= SEQUENCE {

 -- Enables spatial bundling of HARQ ACKs. It is configured per cell group (i.e. for all the cells within the cell group) for PUCCH

 -- reporting of HARQ-ACK. It is only applicable when more than 4 layers are possible to schedule.

 -- Corresponds to L1 parameter 'HARQ-ACK-spatial-bundling' (see 38.213, section FFS\_Section)

 -- Absence indicates that spatial bundling is disabled.

 harq-ACK-SpatialBundlingPUCCH ENUMERATED {true} OPTIONAL, -- Need R

 -- Enables spatial bundling of HARQ ACKs. It is configured per cell group (i.e. for all the cells within the cell group) for PUSCH

 -- reporting of HARQ-ACK. It is only applicable when more than 4 layers are possible to schedule.

 -- Corresponds to L1 parameter 'HARQ-ACK-spatial-bundling' (see 38.213, section FFS\_Section)

 -- Absence indicates that spatial bundling is disabled.

 harq-ACK-SpatialBundlingPUSCH ENUMERATED {true} OPTIONAL, -- Need R

 -- The maximum transmit power to be used by the UE in this NR cell group.

 p-NR P-Max OPTIONAL, -- Need R

 -- The PDSCH HARQ-ACK codebook is either semi-static of dynamic. This is applicable to both CA and none CA operation.

 -- Corresponds to L1 parameter 'HARQ-ACK-codebook' (see 38.213, section FFS\_Section)

 pdsch-HARQ-ACK-Codebook ENUMERATED {semiStatic, dynamic},

 ...

}

-- Serving cell specific MAC and PHY parameters for a SpCell:

SpCellConfig ::= SEQUENCE {

 -- Serving cell ID of a PSCell (the PCell of the Master Cell Group uses ID = 0)

 servCellIndex ServCellIndex OPTIONAL, -- Cond SCG

 -- Parameters for the synchronous reconfiguration to the target SpCell:

 reconfigurationWithSync ReconfigurationWithSynch OPTIONAL, -- Cond ReconfWithSync

 rlf-TimersAndConstants RLF-TimersAndConstants OPTIONAL, -- Need M

 rlmInSyncOutOfSyncThrehold INTEGER (0..1) OPTIONAL, -- Need M

 spCellConfigDedicated ServingCellConfig OPTIONAL, -- Need M

 ...

}

ReconfigurationWithSynch ::= SEQUENCE {

 spCellConfigCommon ServingCellConfigCommon, OPTIONAL, -- Need M

 newUE-Identity RNTI-Value,

 t304 ENUMERATED {ms50, ms100, ms150, ms200, ms500, ms1000, ms2000, ms10000},

 rach-ConfigDedicated CHOICE {

 uplink RACH-ConfigDedicated,

 supplementaryUplink RACH-ConfigDedicated

 } OPTIONAL, -- Need N

 ...

}

SCellConfig ::= SEQUENCE {

 sCellIndex SCellIndex,

 sCellConfigCommon ServingCellConfigCommon OPTIONAL, -- Cond SCellAdd

 sCellConfigDedicated ServingCellConfig OPTIONAL, -- Cond SCellAddMod

 ...

}

-- TAG-CELL-GROUP-CONFIG-STOP

-- ASN1STOP

|  |
| --- |
| *CellGroupConfig* field descriptions |
| ***logicalChannelIdentity***The logical channel identity for both UL and DL. |

|  |  |
| --- | --- |
| Conditional Presence | Explanation |
| *LCH-SetupOnly* | The field is mandatory present if the corresponding LCH is being set up; otherwise it is not present. |
| *LCH-Setup* | The field is mandatory present if the corresponding LCH is being set up for DRB; otherwise it is optionally present, need M. |
| *ReconfWithSync* | The field is mandatory present in case of SpCell change and security key change; otherwise it is optionally present, need M. |
| *SCellAdd* | The field is optionally present, need M, upon SCell addition; otherwise it is not present |
| *SCellAddMod* | The field is mandatory present upon SCell addition; otherwise it is optionally present, need M. |

#### – *DRB-Identity*

The IE *DRB-Identity* is used to identify a DRB used by a UE.

*DRB-Identity* information elements

-- ASN1START

-- TAG-DRB-IDENTITY-START

DRB-Identity ::= INTEGER (1..32)

-- TAG-DRB-IDENTITY-STOP

-- ASN1STOP

#### *–* *MeasResultSCG-Failure*

The IE *MeasResultSCG-Failure* is used to provide information regarding failures detected by the UE in case of EN-DC.

*MeasResultSCG-Failure* information element

-- ASN1START

-- TAG-MEAS-RESULT -SCG-FAILURE-START

-- FFS if failureType is needed

MeasResultSCG-Failure ::= SEQUENCE {

 measResultServFreqList MeasResultServFreqList2NR,

 measResultNeighCells MeasResultList2NR,

 ...

}

MeasResultServFreqList2NR ::= SEQUENCE (SIZE (1..maxNrofServingCells)) OF MeasResultServFreq2NR

MeasResultServFreq2NR ::= SEQUENCE {

 carrierFreq ARFCN-ValueNR,

 measResultServingCell MeasResultNR,

 measResultBestNeighCell MeasResultNR OPTIONAL

}

MeasResultList2NR ::= SEQUENCE (SIZE (1..maxFreq)) OF MeasResult2NR

MeasResult2NR ::= SEQUENCE {

 carrierFreq ARFCN-ValueNR,

 measResultListNR MeasResultListNR

}

-- TAG-MEAS-RESULT -SCG-FAILURE-STOP

-- ASN1STOP

#### – *RadioBearerConfig*

The IE *RadioBearerConfig* is used to add, modify and release signalling and/or data radio bearers. Specifically, this IE carries the parameters for PDCP and, if applicable, SDAP entities for the radio bearers.

*RadioBearerConfig* information element

-- ASN1START

-- TAG-RADIO-BEARER-CONFIG-START

RadioBearerConfig ::= SEQUENCE {

 srb-ToAddModList SRB-ToAddModList OPTIONAL, -- Need N

 srb3-ToRelease ENUMERATED{true} OPTIONAL, -- Need N

 drb-ToAddModList DRB-ToAddModList OPTIONAL, -- Need N

 drb-ToReleaseList DRB-ToReleaseList OPTIONAL, -- Need N

 securityConfig SecurityConfig OPTIONAL, -- Cond M

 ...

}

SRB-ToAddModList ::= SEQUENCE (SIZE (1..2)) OF SRB-ToAddMod

SRB-ToAddMod ::= SEQUENCE {

 srb-Identity SRB-Identity,

 -- may only be set if the cell groups of all linked logical channels are reset or released

 reestablishPDCP ENUMERATED{true} OPTIONAL, -- Need N

 discardOnPDCP ENUMERATED{true} OPTIONAL, -- Need N

 pdcp-Config PDCP-Config OPTIONAL, -- Cond PDCP

 ...

}

DRB-ToAddModList ::= SEQUENCE (SIZE (1..maxDRB)) OF DRB-ToAddMod

DRB-ToAddMod ::= SEQUENCE {

 cnAssociation CHOICE {

 -- The EPS bearer ID determines the EPS bearer when NR connects to EPC using EN-DC

 eps-BearerIdentity INTEGER (0..15), -- EPS-DRB-Setup

 -- The SDAP configuration determines how to map QoS flows to DRBs when NR connects to the 5GC

 sdap-Config SDAP-Config -- 5GC

 } OPTIONAL, -- Cond DRBSetup

 drb-Identity DRB-Identity,

 -- may only be set if the cell groups of all linked logical channels are reset or released

 reestablishPDCP ENUMERATED{true} OPTIONAL, -- Need N

 recoverPDCP ENUMERATED{true} OPTIONAL, -- Need N

 pdcp-Config PDCP-Config OPTIONAL, -- Cond PDCP

 ...

}

DRB-ToReleaseList ::= SEQUENCE (SIZE (1..maxDRB)) OF DRB-Identity

SecurityConfig ::= SEQUENCE {

 securityAlgorithmConfig SecurityAlgorithmConfig OPTIONAL, -- Cond RBTermChange

 keyToUse ENUMERATED{keNB, s-KgNB} OPTIONAL, -- Cond RBTermChange

 ...

}

-- TAG-RADIO-BEARER-CONFIG-STOP

-- ASN1STOP

|  |
| --- |
| *RadioBearerConfig* field descriptions |
| ***drb-Identity***In case of DC, the DRB identity is unique within the scope of the UE, i.e. an MCG DRB cannot use the same value as a split DRB. For a split DRB the same identity is used for the MCG and SCG parts of the configuration. |
| ***cnAssociation***Indicates if the bearer is associated with the eps-bearerIdentity (when connected to EPC) or sdap-Config (when connected to 5GC). |
| ***keyToUse***Indicates if the bearers configured with the list in this *radioBearerConfig* is using KeNB or S-KgNB for deriving ciphering and/or integrity protection keys. Network should not configure SRB1 and SRB2 with S-KgNB and SRB3 with KeNB. When the field is not included, the UE shall continue to use the currently configured *keyToUse* for the radio bearers reconfigured with the lists in this *radioBearerConfig*. |
| reestablishPDCPIndicates that PDCP should be re-established. Network sets this to TRUE whenever the security key used for this radio bearer changes. |
| ***srb-Identity***Value 1 is applicable for SRB1 only.Value 2 is applicable for SRB2 only.Value 3 is applicable for SRB3 only. |
| ***securityAlgorithmConfig***Indicates the security algorithm for the signalling and data radio bearers configured with the list in this radioBearerConfig. When the field is not included, the UE shall continue to use the currently configured security algorithm for the radio bearers reconfigured with the lists in this radioBearerConfig. |
| ***securityConfig***Indicates the security algorithm and key to use for the signalling and data radio bearers configured with the list in this radioBearerConfig When the field is not included, the UE shall continue to use the currently configured keyToUse and security algorithm for the radio bearers reconfigured with the lists in this radioBearerConfig. |
| ***srb3-toRelease***Release SRB3. SRB3 release can only be done at SCG release and reconfiguration with sync |

|  |  |
| --- | --- |
| Conditional Presence | Explanation |
| *RBTermChange* | The field is mandatory present in case of set up of signalling and data radio bearer and change of termination point for the radio bearer between MN and SN. It is optionally present otherwise, Need S. |
| *PDCP* | The field is mandatory present if the corresponding DRB is being setup or corresponding RB is reconfigured with NR PDCP; otherwise the field is optionally present, need M. |
| *DRBSetup* | The field is mandatory present if the corresponding DRB is being setup; otherwise the field is optionally present, need M. |

#### – *RLF-TimersAndConstants*

Editor’s Note: FFS / TODO: Insert the RLF timers and related functionality. Check what is needed for EN-DC.

The *RLF-TimersAndConstants* IE is used to configure UE specific timers and constants.

*RLF-TimersAndConstants* information element

-- ASN1START

-- TAG-RLF-TIMERS-AND-CONSTANTS-START

RLF-TimersAndConstants ::= SetupRelease {

 SEQUENCE {

 t310 ENUMERATED {ms0, ms50, ms100, ms200, ms500, ms1000, ms2000, ms4000, ms6000},

 n310 ENUMERATED {n1, n2, n3, n4, n6, n8, n10, n20},

 n311 ENUMERATED {n1, n2, n3, n4, n5, n6, n8, n10},

 ...

 }

}

-- TAG-RLF-TIMERS-AND-CONSTANTS-STOP

-- ASN1STOP

| *RLF-TimersAndConstants* field descriptions |
| --- |
| ***n3xy***Constants are described in section 7.4. n1 corresponds with 1, n2 corresponds to 2 and so on. |
| ***t3xy***Timers are described in section 7.3. Value ms0 corresponds with 0 ms, ms50 corresponds to 50 ms and so on. |

#### – *SecurityAlgorithmConfig*

The IE *SecurityAlgorithmConfig* is used to configure AS integrity protection algorithm (SRBs) and AS ciphering algorithm (SRBs and DRBs).

*SecurityAlgorithmConfig* information element

-- ASN1START

-- TAG-SECURITY-ALGORITHM-CONFIG-START

SecurityAlgorithmConfig ::= SEQUENCE {

 cipheringAlgorithm CipheringAlgorithm,

 integrityProtAlgorithm IntegrityProtAlgorithm OPTIONAL -- Need R

 ...

}

IntegrityProtAlgorithm ::= ENUMERATED {

 nia0, nia1, nia2, nia3, spare4, spare3,

 spare2, spare1, ...}

CipheringAlgorithm ::= ENUMERATED {

 nea0, nea1, nea2, nea3, spare4, spare3,

 spare2, spare1, ...}

-- TAG-SECURITY-ALGORITHM-CONFIG-STOP

-- ASN1STOP

| *SecurityAlgorithmConfig* field descriptions |
| --- |
| ***cipheringAlgorithm***Indicates the ciphering algorithm to be used for SRBs and DRBs, as specified in TS 33.501 [11]. The algorithms nea0-nea3 are identical to the LTE algorithms eea0-3. For EN-DC, the algorithms configured for bearers using KeNB shall be the same as for all bearers using KeNB and the algorithms configured for bearers using KgNB shall be the same as for all bearers using KgNB. |
| ***integrityProtAlgorithm***For EN-DC, this IE indicates the integrity protection algorithm to be used for SRBs, as specified in TS 33.501 [11]. The algorithms nia0-nia3 is identical to the LTE algorithms eia0-3. For EN-DC, the algorithms configured for SRBs using KeNB shall be the same as for all SRBs using KeNB and the algorithms configured for bearers using KgNB shall be the same as for all bearers using KgNB. |

#### – *ServCellIndex*

The IE *ServCellIndex* concerns a short identity, used to identify a serving cell (i.e. the PCell or an SCell). Value 0 applies for the PCell, while the *SCellIndex* that has previously been assigned applies for SCells.

*ServCellIndex* information element

-- ASN1START

-- TAG-SERV-CELL-INDEX-START

ServCellIndex ::= INTEGER (0..maxNrofServingCells-1)

-- TAG-SERV-CELL-INDEX-STOP

-- ASN1STOP

#### – *SRB-Identity*

The IE SRB-Identity is used to identify a Signalling Radio Bearer (SRB) used by a UE.

-- ASN1START

-- TAG-SRB-IDENTITY-START

SRB-Identity ::= INTEGER (1..3)

-- TAG-SRB-IDENTITY-STOP

-- ASN1STOP

### 6.3.4 Other information elements

### – Multiplicity and type constraint definitions

-- ASN1START

-- TAG-MULTIPLICITY-AND-TYPE-CONSTRAINT-DEFINITIONS-START

maxBandComb INTEGER ::= 65536 -- Maximum number of DL band combinations

maxBasebandProcComb INTEGER ::= 65536 -- Maximum number of base band processing combinations

maxNrofServingCells INTEGER ::= 32 -- Max number of serving cells (SpCell + SCells) per cell group

maxNrofServingCells-1 INTEGER ::= 31 -- Max number of serving cells (SpCell + SCells) per cell group minus 1

maxNrofSCells INTEGER ::= 31 -- Max number of secondary serving cells per cell group

maxNrofCellMeas INTEGER ::= 32 -- Maximum number of entries in each of the cell lists in a measurement object

maxNrofSS-BlocksToAverage INTEGER ::= 16 -- Max number for the (max) number of SS blocks to average to determine cell

 -- measurement

maxNrofCSI-RS-ResourcesToAverage INTEGER ::= 16 -- Max number for the (max) number of CSI-RS to average to determine cell

 -- measurement

maxNrofDL-Allocations INTEGER ::= 16 -- Maximum number of PDSCH time domain resource allocations

maxNrofSR-ConfigPerCellGroup INTEGER ::= 8 -- Maximum number of SR configurations per cell group

maxLCG-ID INTEGER ::= 7 -- Maximum value of LCG ID

maxLC-ID INTEGER ::= 32 -- Maximum value of Logical Channel ID

maxNrofTAGs INTEGER ::= 4 -- Maximum number of Timing Advance Groups

maxNrofTAGs-1 INTEGER ::= 3 -- Maximum number of Timing Advance Groups minus 1

maxNrofBWPs INTEGER ::= 4 -- Maximum number of BWPs per serving cell

maxNrofBWPs-1 INTEGER ::= 3 -- Maximum number of BWPs per serving cell minus 1

maxNrofSymbols-1 INTEGER ::= 13 -- Maximum index identifying a symbol within a slot (14 symbols, indexed from 0..13)

maxNrofSlots INTEGER ::= 320 -- Maximum number of slots in a 10 ms period

maxNrofSlots-1 INTEGER ::= 319 -- Maximum number of slots in a 10 ms period minus 1

maxNrofPhysicalResourceBlocks INTEGER ::= 275 -- Maximum number of PRBs

maxNrofPhysicalResourceBlocks-1 INTEGER ::= 274 -- Maximum number of PRBs

maxNrofControlResourceSets INTEGER ::= 12 -- Max number of CoReSets configurable on a serving cell

maxNrofControlResourceSets-1 INTEGER ::= 11 -- Max number of CoReSets configurable on a serving cell minus 1

maxCoReSetDuration INTEGER ::= 3 -- Max number of OFDM symbols in a control resource set

maxNrofSearchSpaces INTEGER ::= 40 -- Max number of Search Spaces

maxNrofSearchSpaces-1 INTEGER ::= 39 -- Max number of Search Spaces minus 1

maxSFI-DCI-PayloadSize INTEGER ::= 128 -- Max number payload of a DCI scrambled with SFI-RNTI

maxSFI-DCI-PayloadSize-1 INTEGER ::= 127 -- Max number payload of a DCI scrambled with SFI-RNTI minus 1

maxINT-DCI-PayloadSize INTEGER ::= ffsValue -- Max number payload of a DCI scrambled with INT-RNTI

maxINT-DCI-PayloadSize-1 INTEGER ::= ffsValue -- Max number payload of a DCI scrambled with INT-RNTI minus 1

maxNrofRateMatchPatterns INTEGER ::= 4 -- Max number of rate matching patterns that may be configured

maxNrofRateMatchPatterns-1 INTEGER ::= 3 -- Max number of rate matching patterns that may be configured minus 1

maxNrofCSI-Reports INTEGER ::= 48 -- Maximum number of report configurations

maxNrofCSI-RS-CellsRRM INTEGER ::= 96 -- Maximum number of FFS

maxNrofReportConfigIdsPerTrigger INTEGER ::= 16 -- Maximum number of report configurations per reportTrigger

maxNrofCSI-ResourceConfigurations INTEGER ::= 112 -- Maximum number of resource configurations

maxNrofCSI-ResourceConfigurations-1 INTEGER ::= 111 -- Maximum number of resource configurations minus 1

maxNrofCSI-ResourceSets INTEGER ::= 16 -- Maximum number of resource sets per resource configuration

maxNrofCSI-ResourceSets-1 INTEGER ::= 15 -- Maximum number of resource sets per resource configuration minus 1

maxNrofFailureDetectionResources INTEGER ::= 8 -- Maximum number of failure detection resources

maxNrofNZP-CSI-RS-Resources-1 INTEGER ::= ffsValue -- Maximum number of Non-Zero-Power (NZP) CSI-RS resources minus 1

maxNrofZP-CSI-RS-Resources INTEGER ::= 3 -- Maximum number of Zero-Power (NZP) CSI-RS resources

maxNrofZP-CSI-RS-Resources-1 INTEGER ::= 2 -- Maximum number of Zero-Power (NZP) CSI-RS resources minus 1

maxNrofCSI-IM-Resources INTEGER ::= 12 -- Maximum number of CSI-IM resources. See CSI-IM-ResourceMax in 38.214.

maxNrofCSI-IM-Resources-1 INTEGER ::= 11 -- Maximum number of CSI-IM resources minus 1. See CSI-IM-ResourceMax in 38.214.

maxNrofCSI-IM-ResourcesPerSet INTEGER ::= 8 -- Maximum number of CSI-IM resources per set. See CSI-IM-ResourcePerSetMax in 38.214

maxNrofSSB-Resources-1 INTEGER ::= 63 -- Maximum number of SSB resources in a resource set minus 1

maxNrofCSI-RS-ResourcesPerSet INTEGER ::= 8 -- Maximum number of CSI-RS resources per resource set

maxNrofCSI-MeasId INTEGER ::= ffsValue -- Maximum number of link configurations

maxNrofCSI-MeasId-1 INTEGER ::= ffsValue -- Maximum number of link configurations minus 1

maxNrofCSI-RS-ResourcesRRM INTEGER ::= 96 -- Maximum number of CSI-RS resources for an RRM measurement object

maxNrofCSI-RS-ResourcesRRM-1 INTEGER ::= 95 -- Maximum number of CSI-RS resources for an RRM measurement object minus 1

maxNrofObjectId INTEGER ::= 64 -- Maximum number of measurement objects

maxNrofPCI-Ranges INTEGER ::= 8 -- Maximum number of PCI ranges

maxReportConfigId INTEGER ::= 64 -- Maximum number of reporting configurations

maxNrofMeasId INTEGER ::= 64 -- Maximum number of configured measurements

maxNrofQuantityConfig INTEGER ::= 2 -- Maximum number of quantity configurations

maxNrofSRS-ResourceSets INTEGER ::= ffsValue -- Maximum number of SRS resource sets in a BWP.

maxNrofSRS-ResourceSets-1 INTEGER ::= ffsValue -- Maximum number of SRS resource sets in a BWP minus 1.

maxNrofSRS-Resources INTEGER ::= ffsValue -- Maximum number of SRS resources in an SRS resource set.

maxNrofSRS-Resources-1 INTEGER ::= ffsValue -- Maximum number of SRS resources in an SRS resource set minus 1.

maxNrofSRS-TriggerStates-1 INTEGER ::= 3 -- Maximum number of SRS trigger states minus 1, i.e., the largest code point.

maxRAT-CapabilityContainers INTEGER ::= 8 -- Maximum number of interworking RAT containers (incl NR and MRDC)

maxSimultaneousBands INTEGER ::= 32 -- Maximum number of simultaneously aggregated bands

maxNrofSlotFormatCombinationsPerCell INTEGER ::= 16 -- Maximum number of

maxNrofSlotFormatCombinationsPerSet INTEGER ::= 4096 -- Maximum number of Slot Format Combinations in a SF-Set.

maxNrofSlotFormatCombinationsPerSet-1 INTEGER ::= 4095 -- Maximum number of Slot Format Combinations in a SF-Set minus 1.

maxNrofPUCCH-ResourceSets INTEGER ::= 4 -- Maximum number of PUCCH Resource Sets

maxNrofPUCCH-ResourceSets-1 INTEGER ::= 3 -- Maximum number of PUCCH Resource Sets minus 1.

maxNrofPUCCH-ResourcesPerSet INTEGER ::= 8 -- Maximum number of PUCCH Resources per PUCCH-ResourceSet

maxNrofPUCCH-ResourcesPerSet-1 INTEGER ::= 7 -- Maximum number of PUCCH Resources per PUCCH-ResourceSet minus 1.

maxNrofPUCCH-P0-PerSet INTEGER ::= 8 -- Maximum number of P0-pucch present in a p0-pucch set

maxNrofPUCCH-PathlossReferenceRSs INTEGER ::= 4 -- Maximum number of RSs used as pathloss reference for PUCCH power control.

maxNrofPUCCH-PathlossReferenceRSs-1 INTEGER ::= 3 -- Maximum number of RSs used as pathloss reference for PUCCH power control minus 1.

maxNrofP0-PUSCH-AlphaSets INTEGER ::= 30 -- Maximum number of P0-pusch-alpha-sets (see 38,213, section 7.1)

maxNrofP0-PUSCH-AlphaSets-1 INTEGER ::= 29 -- Maximum number of P0-pusch-alpha-sets minus 1 (see 38,213, section 7.1)

maxNrofPUSCH-PathlossReferenceRSs INTEGER ::= 4 -- Maximum number of RSs used as pathloss reference for PUSCH power control.

maxNrofPUSCH-PathlossReferenceRSs-1 INTEGER ::= 3 -- Maximum number of RSs used as pathloss reference for PUSCH power control minus 1.

maxEARFCN INTEGER ::= 262143 -- Highest value of extended E-UTRA EARFCN range

maxBands INTEGER ::= 1024 -- Maximum number of supported bands in UE capability.

maxCellPrep INTEGER ::= 32

maxCellReport INTEGER ::= 8

maxDRB INTEGER ::= 29 -- Maximum number of DRBs (that can be added in DRB-ToAddModLIst).

maxFreq INTEGER ::= 8 -- Max number of non-serving frequencies in *MeasResultSCG-Failure.*

maxNrofCSI-RS INTEGER ::= ffsValue

maxNrofCandidateBeams INTEGER ::= 16 -- Max number of PRACH-ResourceDedicatedBFR that in BFR config.

maxNrofCSI-ReportConfig-1 INTEGER ::= ffsValue

maxNrofPCIsPerSMTC INTEGER ::= 64 -- Maximun number of PCIs per SMTC.

maxNrofQFIs INTEGER ::= ffsValue

maxNrofSR-Resources INTEGER ::= 8 -- Maximum number of SR resources per BWP in a cell.

maxNrofSlotFormatsPerCombination INTEGER ::= 256

maxNrofSpatialRelationInfos INTEGER ::= ffsValue

maxNrofSRS-ResourcesPerSet INTEGER ::= ffsValue

maxNrofIndexesToReport INTEGER ::= 32

maxNrofSSBs INTEGER ::= 64 -- Maximum number of SSB resources in a resource set.

maxNrofSSBs-1 INTEGER ::= 63 -- Maximum number of SSB resources in a resource set minus 1.

maxNrofTCI-StatesPDCCH INTEGER ::= ffsValue

maxNrofTCI-States INTEGER ::= 64 -- Maximum number of TCI states.

maxNrofTCI-States-1 INTEGER ::= 63 -- Maximum number of TCI states minus 1.

maxNrofUL-Allocations INTEGER ::= 16 -- Maximum number of PUSCH time domain resource allocations.

maxQFI INTEGER ::= ffsValue

maxRA-CSIRS-Resources INTEGER ::= ffsValue

maxRA-SSB-Resources INTEGER ::= ffsValue

maxSCSs INTEGER ::= 5

maxSecondaryCellGroups INTEGER ::= 3

ffsValue INTEGER ::= 64

-- IE definitions introduced to not get warning at ASN.1 syntax check

CandidateRS-IndexInfoList ::= ENUMERATED {ffsTypeAndValue}

CellIdentity ::= ENUMERATED {ffsTypeAndValue}

CSI-RS-Index ::= ENUMERATED {ffsTypeAndValue}

FilterCoefficient ::= ENUMERATED {ffsTypeAndValue}

Hysteresis ::= ENUMERATED {ffsTypeAndValue}

MeasResultSSTD ::= ENUMERATED {ffsTypeAndValue}

PDU-SessionID ::= ENUMERATED {ffsTypeAndValue}

PhyCellNR ::= ENUMERATED {ffsTypeAndValue}

PhysCellIdRange ::= ENUMERATED {ffsTypeAndValue}

P-Max ::= ENUMERATED {ffsTypeAndValue}

RA-Resources ::= ENUMERATED {ffsTypeAndValue}

RRC-TransactionIdentifier ::= ENUMERATED {ffsTypeAndValue}

ShortMAC-I ::= ENUMERATED {ffsTypeAndValue}

SSB-Id ::= ENUMERATED {ffsTypeAndValue}

TimeToTrigger ::= ENUMERATED {ffsTypeAndValue}

UECapabilityInformation ::= ENUMERATED {ffsTypeAndValue}

BW-PerCC ::= ENUMERATED {ffsTypeAndValue}

FFS\_Value ::= ENUMERATED {ffsTypeAndValue}

FreqBandIndicatorNR ::= ENUMERATED {ffsTypeAndValue}

MBSFN-SubframeConfigList ::= ENUMERATED {ffsTypeAndValue}

NZP-CSI-RS-ResourceConfigId ::= ENUMERATED {ffsTypeAndValue}

SlotFormatIndicator ::= ENUMERATED {ffsTypeAndValue}

-- TAG-MULTIPLICITY-AND-TYPE-CONSTRAINT-DEFINITIONS-STOP

-- ASN1STOP

### – End of NR-RRC-Definitions

-- ASN1START

END

-- ASN1STOP

# 7 Variables and constants

## 7.1 Timers

### 7.1.1 Timers (Informative)

| Timer | Start | Stop | At expiry |
| --- | --- | --- | --- |
| T304 | Reception of *RRCReconfiguration* message including *reconfigurationWithSync* | Successful completion of random access on the corresponding SpCellFor T304 of SCG, upon SCG release | For T304 of SCG, inform network about the reconfiguration with synch failure by initiating the SCG failure information procedure as specified in 5.7.3. |
| T310 | Upon detecting physical layer problems for the SpCell i.e. upon receiving N310 consecutive out-of-sync indications from lower layers. | Upon receiving N311 consecutive in-sync indications from lower layers for the SpCell, upon receiving RRCReconfiguration with *reconfigurationWithSync* for that cell group, and upon initiating the connection re-establishment procedure.Upon SCG release, if the T310 is kept in SCG. | If the T310 is kept in MCG: If security is not activated: go to RRC\_IDLE else: initiate the connection re-establishment procedure. If the T310 is kept in SCG, Inform E-UTRAN/NR about the SCG radio link failure by initiating the SCG failure information procedure as specified in 5.7.3. |
| T311 | Upon initiating the RRC connection re-establishment procedure | Selection of a suitable NR cell or a cell using another RAT. | Enter RRC\_IDLE |
|  |  |  |  |

### 7.1.2 Timer handling

When the UE applies zero value for a timer, the timer shall be started and immediately expire unless explicitly stated otherwise.

## 7.2 Counters

| Counter | Reset | Incremented | When reaching max value |
| --- | --- | --- | --- |
|  |  |  |  |

## 7.3 Constants

| Constant | Usage |
| --- | --- |
| N310 | Maximum number of consecutive "out-of-sync" indications for the PCell received from lower layers |
| N311 | Maximum number of consecutive "in-sync" indications for the PCell received from lower layers |
|  |  |
|  |  |

## 7.4 UE variables

NOTE: To facilitate the specification of the UE behavioural requirements, UE variables are represented using ASN.1. Unless explicitly specified otherwise, it is however up to UE implementation how to store the variables. The optionality of the IEs in ASN.1 is used only to indicate that the values may not always be available.

#### – *NR-UE-Variables*

This ASN.1 segment is the start of the NR UE variable definitions.

-- ASN1START

NR-UE-Variables DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS

 MeasId,

 MeasIdToAddModList,

 MeasObjectToAddModList,

 PhysCellIdEUTRA,

 PhyCellNR,

 ReportConfigToAddModList,

 RSRP-Range,

 QuantityConfig,

 maxNrofCellMeas,

 maxNrofMeasId

FROM NR-RRC-Definitions;

-- ASN1STOP

#### – *VarMeasConfig*

The UE variable *VarMeasConfig* includes the accumulated configuration of the measurements to be performed by the UE, covering intra-frequency, inter-frequency and inter-RAT mobility related measurements.

*VarMeasConfig UE variable*

-- ASN1START

-- TAG-VAR-MEAS-CONFIG-START

VarMeasConfig ::= SEQUENCE {

 -- Measurement identities

 measIdList MeasIdToAddModList OPTIONAL,

 -- Measurement objects

 measObjectList MeasObjectToAddModList OPTIONAL,

 -- Reporting configurations

 reportConfigList ReportConfigToAddModList OPTIONAL,

 -- Other parameters

 quantityConfig QuantityConfig OPTIONAL,

 s-MeasureConfig CHOICE {

 ssb-RSRP RSRP-Range,

 csi-RSRP RSRP-Range

 } OPTIONAL

}

-- TAG-VAR-MEAS-CONFIG-STOP

-- ASN1STOP

Editor’s Note: FFS Revisit whether we really need *VarMeasConfig*.

#### – *VarMeasReportList*

The UE variable *VarMeasReportList* includes information about the measurements for which the triggering conditions have been met.

*VarMeasReportList UE variable*

-- ASN1START

-- TAG-VAR-MEAS-REPORT-START

VarMeasReportList ::= SEQUENCE (SIZE (1..maxNrofMeasId)) OF VarMeasReport

VarMeasReport ::= SEQUENCE {

 -- List of measurement that have been triggered

 measId MeasId,

 cellsTriggeredList CellsTriggeredList OPTIONAL,

 numberOfReportsSent INTEGER

}

CellsTriggeredList ::= SEQUENCE (SIZE (1.. maxNrofCellMeas)) OF CHOICE {

 physCellIdEUTRA PhysCellIdEUTRA,

 phyCellNR PhyCellNR

 }

-- TAG-VAR-MEAS-REPORT-STOP

-- ASN1STOP

#### – End of *NR-UE-Variables*

-- ASN1START

END

-- ASN1STOP

# 8 Protocol data unit abstract syntax

## 8.1 General

The RRC PDU contents in clause 6 and clause 10 are described using abstract syntax notation one (ASN.1) as specified in ITU-T Rec. X.680 [6] and X.681 [7]. Transfer syntax for RRC PDUs is derived from their ASN.1 definitions by use of Packed Encoding Rules, unaligned as specified in ITU-T Rec. X.691 [8].

The following encoding rules apply in addition to what has been specified in X.691:

- When a bit string value is placed in a bit-field as specified in 15.6 to 15.11 in X.691, the leading bit of the bit string value shall be placed in the leading bit of the bit-field, and the trailing bit of the bit string value shall be placed in the trailing bit of the bit-field.

NOTE: The terms 'leading bit' and 'trailing bit' are defined in ITU-T Rec. X.680. When using the 'bstring' notation, the leading bit of the bit string value is on the left, and the trailing bit of the bit string value is on the right.

- When decoding types constrained with the ASN.1 Contents Constraint ("CONTAINING"), automatic decoding of the contained type should not be performed because errors in the decoding of the contained type should not cause the decoding of the entire RRC message PDU to fail. It is recommended that the decoder first decodes the outer PDU type that contains the OCTET STRING or BIT STRING with the Contents Constraint, and then decodes the contained type that is nested within the OCTET STRING or BIT STRING as a separate step.

- When decoding a) RRC message PDUs, b) BIT STRING constrained with a Contents Constraint, or c) OCTET STRING constrained with a Contents Constraint, PER decoders are required to never report an error if there are extraneous zero or non-zero bits at the end of the encoded RRC message PDU, BIT STRING or OCTET STRING.

## 8.2 Structure of encoded RRC messages

An RRC PDU, which is the bit string that is exchanged between peer entities/across the radio interface contains the basic production as defined in X.691.

RRC PDUs shall be mapped to and from PDCP SDUs (in case of DCCH) or RLC SDUs (in case of PCCH, BCCH or CCCH) upon transmission and reception as follows:

- when delivering an RRC PDU as an PDCP SDU to the PDCP layer for transmission, the first bit of the RRC PDU shall be represented as the first bit in the PDCP SDU and onwards; and

- when delivering an RRC PDU as an RLC SDU to the RLC layer for transmission, the first bit of the RRC PDU shall be represented as the first bit in the RLC SDU and onwards; and

- upon reception of an PDCP SDU from the PDCP layer, the first bit of the PDCP SDU shall represent the first bit of the RRC PDU and onwards; and

- upon reception of an RLC SDU from the RLC layer, the first bit of the RLC SDU shall represent the first bit of the RRC PDU and onwards.

## 8.3 Basic production

The 'basic production' is obtained by applying UNALIGNED PER to the abstract syntax value (the ASN.1 description) as specified in X.691. It always contains a multiple of 8 bits.

## 8.4 Extension

The following rules apply with respect to the use of protocol extensions:

- A transmitter compliant with this version of the specification shall, unless explicitly indicated otherwise on a PDU type basis, set the extension part empty. Transmitters compliant with a later version may send non-empty extensions;

- A transmitter compliant with this version of the specification shall set spare bits to zero;

## 8.5 Padding

If the encoded RRC message does not fill a transport block, the RRC layer shall add padding bits. This applies to PCCH and BCCH.

Padding bits shall be set to 0 and the number of padding bits is a multiple of 8.



Figure 8.5-1: RRC level padding

# 10 Generic error handling

## 10.1 General

The generic error handling defined in the subsequent sub-clauses applies unless explicitly specified otherwise e.g. within the procedure specific error handling.

The UE shall consider a value as not comprehended when it is set:

- to an extended value that is not defined in the version of the transfer syntax supported by the UE.

- to a spare or reserved value unless the specification defines specific behaviour that the UE shall apply upon receiving the concerned spare/reserved value.

The UE shall consider a field as not comprehended when it is defined:

- as spare or reserved unless the specification defines specific behaviour that the UE shall apply upon receiving the concerned spare/reserved field.

## 10.2 ASN.1 violation or encoding error

The UE shall:

1> when receiving an RRC message on the [BCCH] for which the abstract syntax is invalid [6]:

2> ignore the message;

NOTE: This section applies in case one or more fields is set to a value, other than a spare, reserved or extended value, not defined in this version of the transfer syntax. E.g. in the case the UE receives value 12 for a field defined as INTEGER (1..11). In cases like this, it may not be possible to reliably detect which field is in the error hence the error handling is at the message level.

## 10.3 Field set to a not comprehended value

The UE shall, when receiving an RRC message on any logical channel:

1> if the message includes a field that has a value that the UE does not comprehend:

2> if a default value is defined for this field:

3> treat the message while using the default value defined for this field;

2> else if the concerned field is optional:

3> treat the message as if the field were absent and in accordance with the need code for absence of the concerned field;

2> else:

3> treat the message as if the field were absent and in accordance with sub-clause 10.4;

## 10.4 Mandatory field missing

The UE shall:

1> if the message includes a field that is mandatory to include in the message (e.g. because conditions for mandatory presence are fulfilled) and that field is absent or treated as absent:

2> if the RRC message was received on DCCH or CCCH:

3> ignore the message;

2> else:

3> if the field concerns a (sub-field of) an entry of a list (i.e. a SEQUENCE OF):

4> treat the list as if the entry including the missing or not comprehended field was not present;

3> else if the field concerns a sub-field of another field, referred to as the 'parent' field i.e. the field that is one nesting level up compared to the erroneous field:

4> consider the 'parent' field to be set to a not comprehended value;

4> apply the generic error handling to the subsequent 'parent' field(s), until reaching the top nesting level i.e. the message level;

3> else (field at message level):

4> ignore the message;

NOTE 1: The error handling defined in these sub-clauses implies that the UE ignores a message with the message type or version set to a not comprehended value.

NOTE 2: The nested error handling for messages received on logical channels other than DCCH and CCCH applies for errors in extensions also, even for errors that can be regarded as invalid network operation e.g. the network not observing conditional presence.

The following ASN.1 further clarifies the levels applicable in case of nested error handling for errors in extension fields.

-- /example/ ASN1START

-- Example with extension addition group

ItemInfoList ::= SEQUENCE (SIZE (1..max)) OF ItemInfo

ItemInfo ::= SEQUENCE {

 itemIdentity INTEGER (1..max),

 field1 Field1,

 field2 Field2 OPTIONAL, -- Need N

 ...

 [[ field3-r9 Field3-r9 OPTIONAL, -- Cond Cond1

 field4-r9 Field4-r9 OPTIONAL -- Need N

 ]]

}

-- Example with traditional non-critical extension (empty sequence)

BroadcastInfoBlock1 ::= SEQUENCE {

 itemIdentity INTEGER (1..max),

 field1 Field1,

 field2 Field2 OPTIONAL, -- Need N

 nonCriticalExtension BroadcastInfoBlock1-v940-IEs OPTIONAL

}

BroadcastInfoBlock1-v940-IEs::= SEQUENCE {

 field3-r9 Field3-r9 OPTIONAL, -- Cond Cond1

 field4-r9 Field4-r9 OPTIONAL, -- Need N

 nonCriticalExtension SEQUENCE {} OPTIONAL -- Need S

}

-- ASN1STOP

The UE shall, apply the following principles regarding the levels applicable in case of nested error handling:

- an extension additon group is not regarded as a level on its own. E.g. in the ASN.1 extract in the previous, a error regarding the conditionality of *field3* would result in the entire itemInfo entry to be ignored (rather than just the extension addition group containing *field3* and *field4*)

- a traditional *nonCriticalExtension* is not regarded as a level on its own. E.g. in the ASN.1 extract in the previous, a error regarding the conditionality of *field3* would result in the entire *BroadcastInfoBlock1* to be ignored (rather than just the non critical extension containing *field3* and *field4*).

## 10.5 Not comprehended field

The UE shall, when receiving an RRC message on any logical channel:

1> if the message includes a field that the UE does not comprehend:

2> treat the rest of the message as if the field was absent;

NOTE: This section does not apply to the case of an extension to the value range of a field. Such cases are addressed instead by the requirements in section 10.3.

# 12 Processing delay requirements for RRC procedures

The UE performance requirements for RRC procedures are specified in the following tables. The performance requirement is expressed as the time in [ms] from the end of reception of the network -> UE message on the UE physical layer up to when the UE shall be ready for the reception of uplink grant for the UE -> network response message with no access delay other than the TTI-alignment (e.g. excluding delays caused by scheduling, the random access procedure or physical layer synchronisation).



Figure 11.2-1: Illustration of RRC procedure delay

Table 11.2-1: UE performance requirements for RRC procedures for UEs

| **Procedure title:** | **Network -> UE** | **UE -> Network** | **Value [ms]** | **Notes** |
| --- | --- | --- | --- | --- |
| **RRC Connection Control Procedures** |
| RRC reconfiguration | *RRCReconfiguration* | *RRCReconfigurationComplete* | X |  |

Annex A (informative): Guidelines, mainly on use of ASN.1

A.1 Introduction

The following clauses contain guidelines for the specification of RRC protocol data units (PDUs) with ASN.1.

A.2 Procedural specification

A.2.1 General principles

The procedural specification provides an overall high level description regarding the UE behaviour in a particular scenario.

It should be noted that most of the UE behaviour associated with the reception of a particular field is covered by the applicable parts of the PDU specification. The procedural specification may also include specific details of the UE behaviour upon reception of a field, but typically this should be done only for cases that are not easy to capture in the PDU section e.g. general actions, more complicated actions depending on the value of multiple fields.

Likewise, the procedural specification need not specify the UE requirements regarding the setting of fields within the messages that are sent to the network i.e. this may also be covered by the PDU specification.

A.2.2 More detailed aspects

The following more detailed conventions should be used:

- Bullets:

- Capitals should be used in the same manner as in other parts of the procedural text i.e. in most cases no capital applies since the bullets are part of the sentence starting with 'The UE shall:'

- All bullets, including the last one in a sub-clause, should end with a semi-colon i.e. an ';'

- Conditions

- Whenever multiple conditions apply, a semi-colon should be used at the end of each conditions with the exception of the last one, i.e. as in 'if cond1, or cond2:

A.3 PDU specification

A.3.1 General principles

A.3.1.1 ASN.1 sections

The RRC PDU contents are formally and completely described using abstract syntax notation (ASN.1), see X.680 [13], X.681 (02/2002) [14].

The complete ASN.1 code is divided into a number of ASN.1 sections in the specifications. In order to facilitate the extraction of the complete ASN.1 code from the specification, each ASN.1 section begins with the following:

- a first text paragraph consisting entirely of an *ASN.1 start tag*, which consists of a double hyphen followed by a single space and the text string "ASN1START" (in all upper case letters).

- a second text paragraph consisting entirely of a *block start tag* is included, which consists of a double hyphen followed by a single space and the text string "TAG-NAME-START" (in all upper case letters), where the "NAME" refers to the main name of the paragraph (in all upper-case letters).

Similarly, each ASN.1 section ends with the following:

- a first text paragraph consisting entirely of a *block* *stop tag*, which consists of a double hyphen followed by a single space and the text string "TAG-NAME-STOP" (in all upper-case letters), where the "NAME" refers to the main name of the paragraph (in all upper-case letters).

- a second text paragraph consisting entirely of an *ASN.1 stop tag*, which consists of a double hyphen followed by a singlespace and the text "ASN1STOP" (in all upper case letters):

This results in the following tags:

-- ASN1START

-- TAG-NAME-START

-- TAG-NAME-STOP

-- ASN1STOP

The text paragraphs containing either of thestart and stop tags should not contain any ASN.1 code significant for the complete description of the RRC PDU contents. The complete ASN.1 code may be extracted by copying all the text paragraphs between an ASN.1 start tag and the following ASN.1 stop tag in the order they appear, throughout the specification.

NOTE: A typical procedure for extraction of the complete ASN.1 code consists of a first step where the entire RRC PDU contents description (ultimately the entire specification) is saved into a plain text (ASCII) file format, followed by a second step where the actual extraction takes place, based on the occurrence of the ASN.1 start and stop tags.

A.3.1.2 ASN.1 identifier naming conventions

The naming of identifiers (i.e., the ASN.1 field and type identifiers) should be based on the following guidelines:

- Message (PDU) identifiers should be ordinary mixed case without hyphenation. These identifiers, *e.g.*, the *RRCConnectionModificationCommand*, should be used for reference in the procedure text. Abbreviations should be avoided in these identifiers and abbreviated forms of these identifiers should not be used.

- Type identifiers other than PDU identifiers should be ordinary mixed case, with hyphenation used to set off acronyms only where an adjacent letter is a capital, *e.g.*, *EstablishmentCause, SelectedPLMN* (not *Selected-PLMN*, since the "d" in "Selected" is lowercase)*, InitialUE-Identity* and *MeasSFN-SFN-TimeDifference*.

- Field identifiers shall start with a lowercase letter and use mixed case thereafter, *e.g.*, *establishmentCause*. If a field identifier begins with an acronym (which would normally be in upper case), the entire acronym is lowercase (*plmn-Identity*, not *pLMN-Identity*). The acronym is set off with a hyphen (*ue-Identity*, not *ueIdentity*), in order to facilitate a consistent search pattern with corresponding type identifiers.

- Identifiers should convey the meaning of the identifier and should avoid adding unnecessary postfixes (e.g. abstractions like 'Info') for the name.

- Identifiers that are likely to be keywords of some language, especially widely used languages, such as C++ or Java, should be avoided to the extent possible.

- Identifiers, other than PDU identifiers, longer than 25 characters should be avoided where possible. It is recommended to use abbreviations, which should be done in a consistent manner i.e. use 'Meas' instead of 'Measurement' for all occurrences. Examples of typical abbreviations are given in table A.3.1.2.1-1 below.

- *For future extension:* When an extension is introduced a suffix is added to the identifier of the concerned ASN.1 field and/or type. A suffix of the form "‑rX" is used, with X indicating the release, for ASN.1 fields or types introduced in a later release (i.e. a release later than the original/first release of the protocol) as well as for ASN.1 fields or types for which a revision is introduced in a later release replacing a previous version, *e.g.*, *Foo-r9* for the Rel-9 version of the ASN.1 type *Foo*. A suffix of the form "‑rXb" is used for the first revision of a field that it appears in the same release (X) as the original version of the field, "‑rXc" for a second intra-release revision and so on. A suffix of the form "‑vXYZ" is used for ASN.1 fields or types that only are an extension of a corresponding earlier field or type (see sub-clause A.4), e.g., *AnElement-v10b0* for the extension of the ASN.1 type *AnElement* introduced in version 10.11.0 of the specification. A number *0...9, 10, 11, etc.* is used to represent the first part of the version number, indicating the release of the protocol. Lower case letters *a, b, c, etc.* are used to represent the second (and third) part of the version number if they are greater than 9. In the procedural specification, in field descriptions as well as in headings suffices are not used, unless there is a clear need to distinguish the extension from the original field.

- More generally, in case there is a need to distinguish different variants of an ASN.1 field or IE, a suffix should be added at the end of the identifiers e.g. *MeasObjectUTRA*, *ConfigCommon*. When there is no particular need to distinguish the fields (e.g. because the field is included in different IEs), a common field identifier name may be used. This may be attractive e.g. in case the procedural specification is the same for the different variants.

- It should be avoided to use field identifiers with the same name within the elements of a CHOICE, including using a CHOICE inside a SEQUENCE (to avoid certain compiler errors).

**Table A.3.1.2-1: Examples of typical abbreviations used in ASN.1 identifiers**

| **Abbreviation** | **Abbreviated word** |
| --- | --- |
| Config | Configuration |
| DL | Downlink |
| Ext | Extension |
| Freq | Frequency |
| Id | Identity |
| Ind | Indication |
| Meas | Measurement |
| MIB | MasterInformationBlock |
| Neigh | Neighbour(ing) |
| Param(s) | Parameter(s) |
| Phys | Physical |
| PCI | Physical Cell Id |
| Proc | Process |
| Reconfig | Reconfiguration |
| Reest | Re-establishment |
| Req | Request |
| Rx | Reception |
| Sched | Scheduling |
| SIB | SystemInformationBlock |
| Sync | Synchronisation |
| Thr | Threshold |
| Tx | Transmission |
| UL | Uplink |

NOTE: The table A.3.1.2.1-1 is not exhaustive. Additional abbreviations may be used in ASN.1 identifiers when needed.

A.3.1.3 Text references using ASN.1 identifiers

A text reference into the RRC PDU contents description from other parts of the specification is made using the ASN.1 field identifier of the referenced type. The ASN.1 field and type identifiers used in text references should be in the *italic font style*. The "do not check spelling and grammar" attribute in Word should be set. Quotation marks (i.e., " ") should not be used around the ASN.1 field or type identifier.

A reference to an RRC PDU should be made using the corresponding ASN.1 field identifier followed by the word "message", e.g., a reference to the *RRCRelease* message.

A reference to a specific part of an RRC PDU, or to a specific part of any other ASN.1 type, should be made using the corresponding ASN.1 field identifier followed by the word "field", e.g., a reference to the *prioritisedBitRate* field in the example below.

-- /example/ ASN1START

LogicalChannelConfig ::= SEQUENCE {

 ul-SpecificParameters SEQUENCE {

 priority Priority,

 prioritisedBitRate PrioritisedBitRate,

 bucketSizeDuration BucketSizeDuration,

 logicalChannelGroup INTEGER (0..3)

 } OPTIONAL

}

-- ASN1STOP

NOTE: All the ASN.1 start tags in the ASN.1 sections, used as examples in this annex to the specification, are deliberately distorted, in order not to include them when the ASN.1 description of the RRC PDU contents is extracted from the specification.

A reference to a specific type of information element should be made using the corresponding ASN.1 type identifier preceded by the acronym "IE", e.g., a reference to the IE *LogicalChannelConfig* in the example above.

References to a specific type of information element should only be used when those are generic, i.e., without regard to the particular context wherein the specific type of information element is used. If the reference is related to a particular context, e.g., an RRC PDU type (message) wherein the information element is used, the corresponding field identifier in that context should be used in the text reference.

A reference to a specific value of an ASN.1 field should be made using the corresponding ASN.1 value without using quotation marks around the ASN.1 value, e.g., 'if the *status* field is set to value *true*'.

A.3.2 High-level message structure

Within each logical channel type, the associated RRC PDU (message) types are alternatives within a CHOICE, as shown in the example below.

-- /example/ ASN1START

DL-DCCH-Message ::= SEQUENCE {

 message DL-DCCH-MessageType

}

DL-DCCH-MessageType ::= CHOICE {

 c1 CHOICE {

 dlInformationTransfer DLInformationTransfer,

 handoverFromEUTRAPreparationRequest HandoverFromEUTRAPreparationRequest,

 mobilityFromEUTRACommand MobilityFromEUTRACommand,

 rrcConnectionReconfiguration RRCConnectionReconfiguration,

 rrcConnectionRelease RRCConnectionRelease,

 securityModeCommand SecurityModeCommand,

 ueCapabilityEnquiry UECapabilityEnquiry,

 spare1 NULL

 },

 messageClassExtension SEQUENCE {}

}

-- ASN1STOP

A nested two-level CHOICE structure is used, where the alternative PDU types are alternatives within the inner level *c1* CHOICE.

Spare alternatives (i.e., *spare1* in this case) may be included within the *c1* CHOICE to facilitate future extension. The number of such spare alternatives should not extend the total number of alternatives beyond an integer-power-of-two number of alternatives (i.e., eight in this case).

Further extension of the number of alternative PDU types is facilitated using the *messageClassExtension* alternative in the outer level CHOICE.

A.3.3 Message definition

Each PDU (message) type is specified in an ASN.1 section similar to the one shown in the example below.

-- /example/ ASN1START

RRCConnectionReconfiguration ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 c1 CHOICE{

 rrcConnectionReconfiguration-r8 RRCConnectionReconfiguration-r8-IEs,

 spare3 NULL, spare2 NULL, spare1 NULL

 },

 criticalExtensionsFuture SEQUENCE {}

 }

}

RRCConnectionReconfiguration-r8-IEs ::= SEQUENCE {

 -- Enter the IEs here.

 ...

}

-- ASN1STOP

Hooks for *critical* and *non-critical* extension should normally be included in the PDU type specification. How these hooks are used is further described in sub-clause A.4.

Critical extensions are characterised by a redefinition of the PDU contents and need to be governed by a mechanism for protocol version agreement between the encoder and the decoder of the PDU, such that the encoder is prevented from sending a critically extended version of the PDU type, which is not comprehended by the decoder.

Critical extension of a PDU type is facilitated by a two-level CHOICE structure, where the alternative PDU contents are alternatives within the inner level *c1* CHOICE. Spare alternatives (i.e., *spare3* down to *spare1* in this case) may be included within the *c1* CHOICE. The number of spare alternatives to be included in the original PDU specification should be decided case by case, based on the expected rate of critical extension in the future releases of the protocol.

Further critical extension, when the spare alternatives from the original specifications are used up, is facilitated using the *criticalExtensionsFuture* in the outer level CHOICE.

In PDU types where critical extension is not expected in the future releases of the protocol, the inner level *c1* CHOICE and the spare alternatives may be excluded, as shown in the example below.

-- /example/ ASN1START

RRCConnectionReconfigurationComplete ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 rrcConnectionReconfigurationComplete-r8

 RRCConnectionReconfigurationComplete-r8-IEs,

 criticalExtensionsFuture SEQUENCE {}

 }

}

RRCConnectionReconfigurationComplete-r8-IEs ::= SEQUENCE {

 -- Enter the fields here.

 ...

}

-- ASN1STOP

Non-critical extensions are characterised by the addition of new information to the original specification of the PDU type. If not comprehended, a non-critical extension may be skipped by the decoder, whilst the decoder is still able to complete the decoding of the comprehended parts of the PDU contents.

Non-critical extensions at locations other than the end of the message or other than at the end of a field contained in a BIT or OCTET STRING are facilitated by use of the ASN.1 extension marker "...". The original specification of a PDU type should normally include the extension marker at the end of the sequence of information elements contained.

Non-critical extensions at the end of the message or at the end of a field that is contained in a BIT or OCTET STRING may be facilitated by use of an empty sequence that is marked OPTIONAL e.g. as shown in the following example:

-- /example/ ASN1START

RRCMessage-r8-IEs ::= SEQUENCE {

 field1 InformationElement1,

 field2 InformationElement2,

 nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- ASN1STOP

The ASN.1 section specifying the contents of a PDU type may be followed by a *field description* table where a further description of, e.g., the semantic properties of the fields may be included. The general format of this table is shown in the example below. The field description table is absent in case there are no fields for which further description needs to be provided e.g. because the PDU does not include any fields, or because an IE is defined for each field while there is nothing specific regarding the use of this IE that needs to be specified.

| *%PDU-TypeIdentifier%* field descriptions |
| --- |
| ***%field identifier%***Field description. |
| ***%field identifier%***Field description. |

The field description table has one column. The header row shall contain the ASN.1 type identifier of the PDU type.

The following rows are used to provide field descriptions. Each row shall include a first paragraph with a *field identifier* (in ***bold and italic*** font style) referring to the part of the PDU to which it applies. The following paragraphs at the same row may include (in regular font style), e.g., semantic description, references to other specifications and/or specification of value units, which are relevant for the particular part of the PDU.

The parts of the PDU contents that do not require a field description shall be omitted from the field description table.

A.3.4 Information elements

Each IE (information element) type is specified in an ASN.1 section similar to the one shown in the example below.

-- /example/ ASN1START

PRACH-ConfigSIB ::= SEQUENCE {

 rootSequenceIndex INTEGER (0..1023),

 prach-ConfigInfo PRACH-ConfigInfo

}

PRACH-Config ::= SEQUENCE {

 rootSequenceIndex INTEGER (0..1023),

 prach-ConfigInfo PRACH-ConfigInfo OPTIONAL -- Need N

}

PRACH-ConfigInfo ::= SEQUENCE {

 prach-ConfigIndex ENUMERATED {ffs},

 highSpeedFlag ENUMERATED {ffs},

 zeroCorrelationZoneConfig ENUMERATED {ffs}

}

-- ASN1STOP

IEs should be introduced whenever there are multiple fields for which the same set of values apply. IEs may also be defined for other reasons e.g. to break down a ASN.1 definition in to smaller pieces.

A group of closely related IE type definitions, like the IEs *PRACH-ConfigSIB* and *PRACH-Config* in this example, are preferably placed together in a common ASN.1 section. The IE type identifiers should in this case have a common base, defined as the *generic type identifier*. It may be complemented by a suffix to distinguish the different variants. The "*PRACH-Config*" is the generic type identifier in this example, and the "*SIB*" suffix is added to distinguish the variant. The sub-clause heading and generic references to a group of closely related IEs defined in this way should use the generic type identifier.

The same principle should apply if a new version, or an extension version, of an existing IE is created for *critical* or *non-critical* extension of the protocol (see sub-clause A.4). The new version, or the extension version, of the IE is included in the same ASN.1 section defining the original. A suffix is added to the type identifier, using the naming conventions defined in sub-clause A.3.1.2, indicating the release or version of the where the new version, or extension version, was introduced.

Local IE type definitions, like the IE *PRACH-ConfigInfo* in the example above, may be included in the ASN.1 section and be referenced in the other IE types defined in the same ASN.1 section. The use of locally defined IE types should be encouraged, as a tool to break up large and complex IE type definitions. It can improve the readability of the code. There may also be a benefit for the software implementation of the protocol end-points, as these IE types are typically provided by the ASN.1 compiler as independent data elements, to be used in the software implementation.

An IE type defined in a local context, like the IE *PRACH-ConfigInfo*, should not be referenced directly from other ASN.1 sections in the RRC specification. An IE type which is referenced in more than one ASN.1 section should be defined in a separate sub-clause, with a separate heading and a separate ASN.1 section (possibly as one in a set of closely related IE types, like the IEs *PRACH-ConfigSIB* and *PRACH-Config* in the example above). Such IE types are also referred to as 'global IEs'.

NOTE: Referring to an IE type, that is defined as a local IE type in the context of another ASN.1 section, does not generate an ASN.1 compilation error. Nevertheless, using a locally defined IE type in that way makes the IE type definition difficult to find, as it would not be visible at an outline level of the specification. It should be avoided.

The ASN.1 section specifying the contents of one or more IE types, like in the example above, may be followed by a *field description* table, where a further description of, e.g., the semantic properties of the fields of the information elements may be included. This table may be absent, similar as indicated in sub-clause A.3.3 for the specification of the PDU type. The general format of the *field description* table is the same as shown in sub-clause A.3.3 for the specification of the PDU type.

A.3.5 Fields with optional presence

A field with optional presence may be declared with the keyword DEFAULT. It identifies a default value to be assumed, if the sender does not include a value for that field in the encoding:

-- /example/ ASN1START

PreambleInfo ::= SEQUENCE {

 numberOfRA-Preambles INTEGER (1..64) DEFAULT 1,

 ...

}

-- ASN1STOP

Alternatively, a field with optional presence may be declared with the keyword OPTIONAL. It identifies a field for which a value can be omitted. The omission carries semantics, which is different from any normal value of the field:

-- /example/ ASN1START

PRACH-Config ::= SEQUENCE {

 rootSequenceIndex INTEGER (0..1023),

 prach-ConfigInfo PRACH-ConfigInfo OPTIONAL -- Need N

}

-- ASN1STOP

The semantics of an optionally present field, in the case it is omitted, should be indicated at the end of the paragraph including the keyword OPTIONAL, using a short comment text with a need code. The need code includes the keyword "Need", followed by one of the predefined semantics tags (S, M, N or R) defined in sub-clause 6.1. If the semantics tag S is used, the semantics of the absent field are further specified either in the field description table following the ASN.1 section, or in procedure text.

The addition of OPTIONAL keywords for capability groups is based on the following guideline. If there is more than one field in the lower level IE, then OPTIONAL keyword is added at the group level. If there is only one field in the lower level IE, OPTIONAL keyword is not added at the group level.

A.3.6 Fields with conditional presence

A field with conditional presence is declared with the keyword OPTIONAL. In addition, a short comment text shall be included at the end of the paragraph including the keyword OPTIONAL. The comment text includes the keyword "Cond", followed by a condition tag associated with the field ("UL" in this example):

-- /example/ ASN1START

LogicalChannelConfig ::= SEQUENCE {

 ul-SpecificParameters SEQUENCE {

 priority INTEGER (0),

 ...

 } OPTIONAL -- Cond UL

}

-- ASN1STOP

When conditionally present fields are included in an ASN.1 section, the field description table after the ASN.1 section shall be followed by a *conditional presence* table. The conditional presence table specifies the conditions for including the fields with conditional presence in the particular ASN.1 section.

| **Conditional presence** | **Explanation** |
| --- | --- |
| UL | **Specification of the conditions for including the field associated with the condition tag = "UL". Semantics in case of optional presence under certain conditions may also be specified.** |

The conditional presence table has two columns. The first column (heading: "Conditional presence") contains the condition tag (in *italic* font style), which links the fields with a condition tag in the ASN.1 section to an entry in the table. The second column (heading: "Explanation") contains a text specification of the conditions and requirements for the presence of the field. The second column may also include semantics, in case of an optional presence of the field, under certain conditions i.e. using the same predefined tags as defined for optional fields in A.3.5.

Conditional presence should primarily be used when presence of a field depends on the presence and/or value of other fields within the same message. If the presence of a field depends on whether another feature/function has been configured, while this function can be configured independently e.g. by another message and/or at another point in time, the relation is best reflected by means of a statement in the field description table.

If the ASN.1 section does not include any fields with conditional presence, the conditional presence table shall not be included.

Whenever a field is only applicable in specific cases e.g. TDD, use of conditional presence should be considered.

A.3.7 Guidelines on use of lists with elements of SEQUENCE type

Where an information element has the form of a list (the SEQUENCE OF construct in ASN.1) with the type of the list elements being a SEQUENCE data type, an information element shall be defined for the list elements even if it would not otherwise be needed.

For example, a list of PLMN identities with reservation flags is defined as in the following example:

-- /example/ ASN1START

PLMN-IdentityInfoList ::= SEQUENCE (SIZE (1..6)) OF PLMN-IdentityInfo

PLMN-IdentityInfo ::= SEQUENCE {

 plmn-Identity PLMN-Identity,

 cellReservedForOperatorUse ENUMERATED {reserved, notReserved}

}

-- ASN1STOP

rather than as in the following (bad) example, which may cause generated code to contain types with unpredictable names:

-- /bad example/ ASN1START

PLMN-IdentityList ::= SEQUENCE (SIZE (1..6)) OF SEQUENCE {

 plmn-Identity PLMN-Identity,

 cellReservedForOperatorUse ENUMERATED {reserved, notReserved}

}

-- ASN1STOP

### A.3.8 Guidelines on use of parameterised SetupRelease type

The usage of the parameterised *SetupRelease* type is like a function call in programming languages where the element type parameter is passed as a parameter. The parameterised type only implies a textual change in abstract syntax where all references to the parameterised type are replaced by the compiler with the release/setup choice. Two examples of the usage are shown below;

-- /example/ ASN1START

RRCMessage-r15-IEs ::= SEQUENCE {

 field-r15 SetupRelease { IE-r15 } OPTIONAL, -- Need M

 ...

}

RRCMessage-r15-IEs ::= SEQUENCE {

 field-r15 SetupRelease { Element-r15 }

} OPTIONAL, -- Need M

Element-r15 ::= SEQUENCE {

 field1-r15 IE1-r15,

 field2-r15 IE2-r15 OPTIONAL -- Need N

} OPTIONAL, -- Need M

-- /example/ ASN1STOP

The *SetupRelease* is always be used with only named IEs, i.e. the example below is not allowed:

-- /example/ ASN1START

RRCMessage-r15-IEs ::= SEQUENCE {

 field-r15 SetupRelease { SEQUENCE { -- Unnamed SEQUENCEs are not allowed!

 field1-r15 IE1-r15,

 field2-r15 IE2-r15 OPTIONAL -- Need N

 }

 } OPTIONAL, -- Need M

}

-- /example/ ASN1STOP

If a field defined using the parameterized SetupRelease type requires procedural text, the field is referred to using the values defined for the type itself, namely, "setup" and "release". For example, procedural text for field-r15 above could be as follows:

1> if *field-r15* is set to "setup":

2> do something;

1> else (*field-r15* is set to "release"):

2> release *field-r15* (if appropriate);

### A.3.9 Guidelines on use of ToAddModList and ToReleaseList

In order to benefit from delta signalling when modifying lists with many and/or large elements, so-called add/mod- and release- lists should be used. Instead of a single list containing all elements of the list, the ASN.1 provides two lists. One list is used to convey the actual elements that are to be added to the list or modified in the list. The second list conveys only the identities (IDs) of the list elements that are to be released from the list. In other words, the ASN.1 defines only means to signal modifications to a list maintained in the receiver (typically the UE). An example is provided below:

-- /example/ ASN1START

AnExampleIE ::= SEQUENCE {

 elementsToAddModList SEQUENCE (SIZE (1..maxNrofElements)) OF Element OPTIONAL, -- Need N

 elementsToReleaseList SEQUENCE (SIZE (1..maxNrofElements)) OF ElementId OPTIONAL, -- Need N

 ...

}

Element ::= SEQUENCE {

 elementId ElementId,

 aField INTEGER (0..16777215),

 anotherField OCTET STRING,

 ...

}

ElementId ::= INTEGER (0..maxNrofElements-1)

maxNrofElements INTEGER ::= 50

maxNrofElements-1 INTEGER ::= 49

-- /example/ ASN1STOP

As can be seen, the elements of the list must contain an identity (INTEGER) that identifies the elements unambiguously upon addition, modification and removal. It is recommended to define an IE for that identifier (here ElementId) so that it can be used both for a field inside the element as well as in the *elementsToReleaseList*.

Both lists should be made OPTIONAL and flagged as ”Need N”. The need code reflects that the UE does not maintain the received lists as such but rather updates its configuration using the information therein. In other words, it is not possible to provide via delta signalling an update to a previously signalled *elementsToAddModList* or elementsToReleaseList (which Need M would imply). The update is always in relation to the UE's internal configuration.

If no procedural text is provided for a set of ToAddModList and ToReleaseList, the following generic procedure applies:

The UE shall:

1> for each *ElementId* in the *elementsToReleaseList*,:

2> if the current UE configuration includes an *Element* with the given *ElementId*:

3> release the *Element* from the current UE configuration;

1> for each *Element* in the *elementsToAddModList*:

2> if the current UE configuration includes an *Element* with the given *ElementId*:

3> modify the configured *Element* in accordance with the received *Element*;

2> else:

3> add received *Element* to the UE configuration;

A.4 Extension of the PDU specifications

A.4.1 General principles to ensure compatibility

It is essential that extension of the protocol does not affect interoperability i.e. it is essential that implementations based on different versions of the RRC protocol are able to interoperate. In particular, this requirement applies for the following kind of protocol extensions:

- Introduction of new PDU types (i.e. these should not cause unexpected behaviour or damage).

- Introduction of additional fields in an extensible PDUs (i.e. it should be possible to ignore uncomprehended extensions without affecting the handling of the other parts of the message).

- Introduction of additional values of an extensible field of PDUs. If used, the behaviour upon reception of an uncomprehended value should be defined.

It should be noted that the PDU extension mechanism may depend on the logical channel used to transfer the message e.g. for some PDUs an implementation may be aware of the protocol version of the peer in which case selective ignoring of extensions may not be required.

The non-critical extension mechanism is the primary mechanism for introducing protocol extensions i.e. the critical extension mechanism is used merely when there is a need to introduce a 'clean' message version. Such a need appears when the last message version includes a large number of non-critical extensions, which results in issues like readability, overhead associated with the extension markers. The critical extension mechanism may also be considered when it is complicated to accommodate the extensions by means of non-critical extension mechanisms.

A.4.2 Critical extension of messages and fields

The mechanisms to critically extend a message are defined in A.3.3. There are both "outer branch" and "inner branch" mechanisms available. The "outer branch" consists of a CHOICE having the name *criticalExtensions*, with two values, *c1* and *criticalExtensionsFuture*. The *criticalExtensionsFuture* branch consists of an empty SEQUENCE, while the c1 branch contains the "inner branch" mechanism.

The "inner branch" structure is a CHOICE with values of the form "*MessageName-rX-IEs*" (e.g., "*RRCConnectionReconfiguration-r8-IEs*") or "*spareX*", with the spare values having type NULL. The "-rX-IEs" structures contain the *complete* structure of the message IEs for the appropriate release; i.e., the critical extension branch for the Rel-10 version of a message includes all Rel-8 and Rel-9 fields (that are not obviated in the later version), rather than containing only the additional Rel-10 fields.

The following guidelines may be used when deciding which mechanism to introduce for a particular message, i.e. only an 'outer branch', or an 'outer branch' in combination with an 'inner branch' including a certain number of spares:

- For certain messages, e.g. initial uplink messages, messages transmitted on a broadcast channel, critical extension may not be applicable.

- An outer branch may be sufficient for messages not including any fields.

- The number of spares within inner branch should reflect the likelihood that the message will be critically extended in future releases (since each release with a critical extension for the message consumes one of the spare values). The estimation of the critical extension likelyhood may be based on the number, size and changeability of the fields included in the message.

- In messages where an inner branch extension mechanism is available, all spare values of the inner branch should be used before any critical extensions are added using the outer branch.

The following example illustrates the use of the critical extension mechanism by showing the ASN.1 of the original and of a later release

-- /example/ ASN1START -- Original release

RRCMessage ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 c1 CHOICE{

 rrcMessage-r8 RRCMessage-r8-IEs,

 spare3 NULL, spare2 NULL, spare1 NULL

 },

 criticalExtensionsFuture SEQUENCE {}

 }

}

-- ASN1STOP

-- /example/ ASN1START -- Later release

RRCMessage ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 c1 CHOICE{

 rrcMessage-r8 RRCMessage-r8-IEs,

 rrcMessage-r10 RRCMessage-r10-IEs,

 rrcMessage-r11 RRCMessage-r11-IEs,

 rrcMessage-r14 RRCMessage-r14-IEs

 },

 later CHOICE {

 c2 CHOICE{

 rrcMessage-r16 RRCMessage-r16-IEs,

 spare7 NULL, spare6 NULL, spare5 NULL, spare4 NULL,

 spare3 NULL, spare2 NULL, spare1 NULL

 },

 criticalExtensionsFuture SEQUENCE {}

 }

 }

}

-- ASN1STOP

It is important to note that critical extensions may also be used at the level of individual fields i.e. a field may be replaced by a critically extended version. When sending the extended version, the original version may also be included (e.g. original field is mandatory, EUTRAN is unaware if UE supports the extended version). In such cases, a UE supporting both versions may be required to ignore the original field. The following example illustrates the use of the critical extension mechanism by showing the ASN.1 of the original and of a later release

-- /example/ ASN1START -- Original release

RRCMessage ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 c1 CHOICE{

 rrcMessage-r8 RRCMessage-r8-IEs,

 spare3 NULL, spare2 NULL, spare1 NULL

 },

 criticalExtensionsFuture SEQUENCE {}

 }

}

RRCMessage-rN-IEs ::= SEQUENCE {

 field1-rN ENUMERATED {

 value1, value2, value3, value4} OPTIONAL, -- Need N

 field2-rN InformationElement2-rN OPTIONAL, -- Need N

 nonCriticalExtension RRCConnectionReconfiguration-vMxy-IEs OPTIONAL

}

RRCConnectionReconfiguration-vMxy-IEs ::= SEQUENCE {

 field2-rM InformationElement2-rM OPTIONAL, -- Cond NoField2rN

 nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- ASN1STOP

| **Conditional presence** | **Explanation** |
| --- | --- |
| *NoField2rN* | The field is optionally present, need N, if *field2-rN* is absent. Otherwise the field is not present |

Finally, it is noted that a critical extension may be introduced in the same release as the one in which the original field was introduced e.g. to correct an essential ASN.1 error. In such cases a UE capability may be introduced, to assist the network in deciding whether or not to use the critically extension.

A.4.3 Non-critical extension of messages

A.4.3.1 General principles

The mechanisms to extend a message in a non-critical manner are defined in A.3.3. W.r.t. the use of extension markers, the following additional guidelines apply:

- When further non-critical extensions are added to a message that has been critically extended, the inclusion of these non-critical extensions in earlier critical branches of the message should be avoided when possible.

- The extension marker ("…") is the primary non-critical extension mechanism that is used but empty sequences may be used if length determinant is not required. Examples of cases where a length determinant is not required:

- at the end of a message,

- at the end of a structure contained in a BIT STRING or OCTET STRING

- When an extension marker is available, non-critical extensions are preferably placed at the location (e.g. the IE) where the concerned parameter belongs from a logical/ functional perspective (referred to as the '*default extension location*')

- It is desirable to aggregate extensions of the same release or version of the specification into a group, which should be placed at the lowest possible level.

- In specific cases it may be preferrable to place extensions elsewhere (referred to as the '*actual extension location*') e.g. when it is possible to aggregate several extensions in a group. In such a case, the group should be placed at the lowest suitable level in the message. <TBD: ref to seperate example>

- In case placement at the default extension location affects earlier critical branches of the message, locating the extension at a following higher level in the message should be considered.

- In case an extension is not placed at the defaultextension location, an IE should be defined. The IE's ASN.1 definition should be placed in the same ASN.1 section as the default extension location. In case there are intermediate levels in-between the actual and the defaultextension location, an IE may be defined for each level. Intermediate levels are primarily introduced for readability and overview. Hence intermediate levels need not allways be introduced e.g. they may not be needed when the default and the actual extension location are within the same ASN.1 section. <TBD: ref to seperate example>

A.4.3.2 Further guidelines

Further to the general principles defined in the previous section, the following additional guidelines apply regarding the use of extension markers:

- Extension markers within SEQUENCE

- Extension markers are primarily, but not exclusively, introduced at the higher nesting levels

- Extension markers are introduced for a SEQUENCE comprising several fields as well as for information elements whose extension would result in complex structures without it (e.g. re-introducing another list)

- Extension markers are introduced to make it possible to maintain important information structures e.g. parameters relevant for one particular RAT

- Extension markers are also used for size critical messages (i.e. messages on BCCH, BR-BCCH, PCCH and CCCH), although introduced somewhat more carefully

- The extension fields introduced (or frozen) in a specific version of the specification are grouped together using double brackets.

- Extension markers within ENUMERATED

- Spare values may be used until the number of values reaches the next power of 2, while the extension marker caters for extension beyond that limit, given that the use of spare values in a later Release is possible without any error cases

- A suffix of the form "vXYZ" is used for the identifier of each new value, e.g. "value-vXYZ".

- Extension markers within CHOICE:

- Extension markers are introduced when extension is foreseen and when comprehension is not required by the receiver i.e. behaviour is defined for the case where the receiver cannot comprehend the extended value (e.g. ignoring an optional CHOICE field). It should be noted that defining the behaviour of a receiver upon receiving a not comprehended choice value is not required if the sender is aware whether or not the receiver supports the extended value.

- A suffix of the form "vXYZ" is used for the identifier of each new choice value, e.g. "choice-vXYZ".

Non-critical extensions at the end of a message/ of a field contained in an OCTET or BIT STRING:

- When a nonCriticalExtension is actually used, a "Need" code should not be provided for the field, which always is a group including at least one extension and a field facilitating further possible extensions. For simplicity, it is recommended not to provide a "Need" code when the field is not actually used either.

Further, more general, guidelines:

- In case a need code is not provided for a group, a "Need" code is provided for all individual extension fields within the group i.e. including for fields that are not marked as OPTIONAL. The latter is to clarify the action upon absence of the whole group.

A.4.3.3 Typical example of evolution of IE with local extensions

The following example illustrates the use of the extension marker for a number of elementary cases (sequence, enumerated, choice). The example also illustrates how the IE may be revised in case the critical extension mechanism is used.

NOTE In case there is a need to support further extensions of release n while the ASN.1 of release (n+1) has been frozen, without requiring the release n receiver to support decoding of release (n+1) extensions, more advanced mechanisms are needed e.g. including multiple extension markers.

-- /example/ ASN1START

InformationElement1 ::= SEQUENCE {

 field1 ENUMERATED {

 value1, value2, value3, value4-v880,

 ..., value5-v960 },

 field2 CHOICE {

 field2a BOOLEAN,

 field2b InformationElement2b,

 ...,

 field2c-v960 InformationElement2c-r9

 },

 ...,

 [[ field3-r9 InformationElement3-r9 OPTIONAL -- Need R

 ]],

 [[ field3-v9a0 InformationElement3-v9a0 OPTIONAL, -- Need R

 field4-r9 InformationElement4 OPTIONAL -- Need R

 ]]

}

InformationElement1-r10 ::= SEQUENCE {

 field1 ENUMERATED {

 value1, value2, value3, value4-v880,

 value5-v960, value6-v1170, spare2, spare1, ... },

 field2 CHOICE {

 field2a BOOLEAN,

 field2b InformationElement2b,

 field2c-v960 InformationElement2c-r9,

 ...,

 field2d-v12b0 INTEGER (0..63)

 },

 field3-r9 InformationElement3-r10 OPTIONAL, -- Need R

 field4-r9 InformationElement4 OPTIONAL, -- Need R

 field5-r10 BOOLEAN,

 field6-r10 InformationElement6-r10 OPTIONAL, -- Need R

 ...,

 [[ field3-v1170 InformationElement3-v1170 OPTIONAL -- Need R

 ]]

}

-- ASN1STOP

Some remarks regarding the extensions of *InformationElement1* as shown in the above example:

– The *InformationElement1* is initially extended with a number of non-critical extensions. In release 10 however, a critical extension is introduced for the message using this IE. Consequently, a new version of the IE *InformationElement1* (i.e. *InformationElement1-r10*) is defined in which the earlier non-critical extensions are incorporated by means of a revision of the original field.

– The *value4-v880* is replacing a spare value defined in the original protocol version for *field1*. Likewise *value6-v1170* replaces *spare3* that was originally defined in the r10 version of *field1*

– Within the critically extended release 10 version of *InformationElement1*, the names of the original fields/IEs are not changed, unless there is a real need to distinguish them from other fields/IEs. E.g. the *field1* and *InformationElement4* were defined in the original protocol version (release 8) and hence not tagged. Moreover, the *field3-r9* is introduced in release 9 and not re-tagged; although, the *InformationElement3* is also critically extended and therefore tagged *InformationElement3-r10* in the release 10 version of InformationElement1.

A.4.3.4 Typical examples of non critical extension at the end of a message

The following example illustrates the use of non-critical extensions at the end of the message or at the end of a field that is contained in a BIT or OCTET STRING i.e. when an empty sequence is used.

-- /example/ ASN1START

RRCMessage-r8-IEs ::= SEQUENCE {

 field1 InformationElement1,

 field2 InformationElement2,

 field3 InformationElement3 OPTIONAL, -- Need N

 nonCriticalExtension RRCMessage-v860-IEs OPTIONAL

}

RRCMessage-v860-IEs ::= SEQUENCE {

 field4-v860 InformationElement4 OPTIONAL, -- Need S

 field5-v860 BOOLEAN OPTIONAL, -- Cond C54

 nonCriticalExtension RRCMessage-v940-IEs OPTIONAL

}

RRCMessage-v940-IEs ::= SEQUENCE {

 field6-v940 InformationElement6-r9 OPTIONAL, -- Need R

 nonCriticalExtensions SEQUENCE {} OPTIONAL

}

-- ASN1STOP

Some remarks regarding the extensions shown in the above example:

– The *InformationElement4* is introduced in the original version of the protocol (release 8) and hence no suffix is used.

A.4.3.5 Examples of non-critical extensions not placed at the default extension location

The following example illustrates the use of non-critical extensions in case an extension is not placed at the defaultextension location.

#### *– ParentIE-WithEM*

The IE *ParentIE-WithEM* is an example of a high level IE including the extension marker (EM). The root encoding of this IE includes two lower level IEs *ChildIE1-WithoutEM* and *ChildIE2-WithoutEM* which not include the extension marker. Consequently, non-critical extensions of the Child-IEs have to be included at the level of the Parent-IE.

The example illustrates how the two extension IEs *ChildIE1-WithoutEM-vNx0* and *ChildIE2-WithoutEM-vNx0* (both in release N) are used to connect non-critical extensions with a default extension location in the lower level IEs to the actual extension location in this IE.

*ParentIE-WithEM* information element

-- /example/ ASN1START

ParentIE-WithEM ::= SEQUENCE {

 -- Root encoding, including:

 childIE1-WithoutEM ChildIE1-WithoutEM OPTIONAL, -- Need N

 childIE2-WithoutEM ChildIE2-WithoutEM OPTIONAL, -- Need N

 ...,

 [[ childIE1-WithoutEM-vNx0 ChildIE1-WithoutEM-vNx0 OPTIONAL, -- Need N

 childIE2-WithoutEM-vNx0 ChildIE2-WithoutEM-vNx0 OPTIONAL -- Need N

 ]]

}

-- ASN1STOP

Some remarks regarding the extensions shown in the above example:

– The fields *childIEx-WithoutEM-vNx0* may not really need to be optional (depends on what is defined at the next lower level).

– In general, especially when there are several nesting levels, fields should be marked as optional only when there is a clear reason.

#### *– ChildIE1-WithoutEM*

The IE *ChildIE1-WithoutEM* is an example of a lower level IE, used to control certain radio configurations including a configurable feature which can be setup or released using the local IE *ChIE1-ConfigurableFeature*. The example illustrates how the new field *chIE1-NewField* is added in release N to the configuration of the configurable feature. The example is based on the following assumptions:

– when initially configuring as well as when modifying the new field, the original fields of the configurable feature have to be provided also i.e. as if the extended ones were present within the setup branch of this feature.

– when the configurable feature is released, the new field should be released also.

– when omitting the original fields of the configurable feature the UE continues using the existing values (which is used to optimise the signalling for features that typically continue unchanged upon handover).

– when omitting the new field of the configurable feature the UE releases the existing values and discontinues the associated functionality (which may be used to support release of unsupported functionality upon handover to an eNB supporting an earlier protocol version).

The above assumptions, which affect the use of conditions and need codes, may not always apply. Hence, the example should not be re-used blindly.

*ChildIE1-WithoutEM* information elements

-- /example/ ASN1START

ChildIE1-WithoutEM ::= SEQUENCE {

 -- Root encoding, including:

 chIE1-ConfigurableFeature ChIE1-ConfigurableFeature OPTIONAL -- Need N

}

ChildIE1-WithoutEM-vNx0 ::= SEQUENCE {

 chIE1-ConfigurableFeature-vNx0 ChIE1-ConfigurableFeature-vNx0 OPTIONAL -- Cond ConfigF

}

ChIE1-ConfigurableFeature ::= CHOICE {

 release NULL,

 setup SEQUENCE {

 -- Root encoding

 }

}

ChIE1-ConfigurableFeature-vNx0 ::= SEQUENCE {

 chIE1-NewField-rN INTEGER (0..31)

}

-- ASN1STOP

| **Conditional presence** | **Explanation** |
| --- | --- |
| *ConfigF* | The field is optional present, need R, in case of *chIE1-ConfigurableFeature* is included and set to "setup"; otherwise the field is not present and the UE shall delete any existing value for this field. |

#### *– ChildIE2-WithoutEM*

The IE *ChildIE2-WithoutEM* is an example of a lower level IE, typically used to control certain radio configurations. The example illustrates how the new field *chIE1-NewField* is added in release N to the configuration of the configurable feature.

*ChildIE2-WithoutEM* information element

-- /example/ ASN1START

ChildIE2-WithoutEM ::= CHOICE {

 release NULL,

 setup SEQUENCE {

 -- Root encoding

 }

}

ChildIE2-WithoutEM-vNx0 ::= SEQUENCE {

 chIE2-NewField-rN INTEGER (0..31) OPTIONAL -- Cond ConfigF

}

-- ASN1STOP

| **Conditional presence** | **Explanation** |
| --- | --- |
| *ConfigF* | The field is optional present, need R, in case of *chIE2-ConfigurableFeature* is included and set to "setup"; otherwise the field is not present and the UE shall delete any existing value for this field. |

A.5 Guidelines regarding inclusion of transaction identifiers in RRC messages

The following rules provide guidance on which messages should include a Transaction identifier

1: DL messages on CCCH that move UE to RRC-Idle should not include the RRC transaction identifier.

2: All network initiated DL messages by default should include the RRC transaction identifier.

3: All UL messages that are direct response to a DL message with an RRC Transaction identifier should include the RRC Transaction identifier.

4: All UL messages that require a direct DL response message should include an RRC transaction identifier.

5: All UL messages that are not in response to a DL message nor require a corresponding response from the network should not include the RRC Transaction identifier.

## A.6 Guidelines regarding use of need codes

The following rule provides guidance for determining need codes for optional downlink fields:

- if the field needs to be stored by the UE (i.e. maintained) when absent:

- use Need M (=Maintain)

- else, if the field needs to be released by the UE when absent:

- use Need R (=Release)

- else, if UE shall take no action when the field is absent (i.e. UE does not even need to maintain any existing value of the field):

- use Need N (=None)

- else (UE behaviour upon absence doesn’t fit any of the above conditions):

- use Need S (=Specified)

- specify the UE behaviour upon absence of the field in the procedural text or in the field description table.

## A.7 Guidelines regarding use of conditions

Conditions are primarily used to specify network restrictions, for which the following types can be distinguished:

- CondM: Message Contents related constraints e.g. that a field B is mandatory present if the same message includes field A and when it is set value X

- CondC: Configuration Constraints e.g. that a field D can only be signalled if field C is configured and set to value Y. (i.e. regardless of whether field C is present in the same message or previously configured)

The use of these conditions is illustrated by an example.

-- /example/ ASN1START

RRCMessage-IEs ::= SEQUENCE {

 fieldA FieldA OPTIONAL, -- Need M

 fieldB FieldB OPTIONAL, -- CondM-FieldAsetToX

 fieldC FieldC OPTIONAL, -- Need M

 fieldD FieldD OPTIONAL, -- CondC-FieldCsetToY

 nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- /example/ ASN1STOP

| Conditional presence | Explanation |
| --- | --- |
| *Message (content) constraints* |
| *CondM-FieldAsetToX* | The field is mandatory present if fieldA is included and set to valueX. Otherwise the field is optional present, need R. |
| *Configuration constraints* |
| *CondC- FieldCsetToY* | The field is optional present, need M, if fieldC is configured and set to valueY. Otherwise the field is not present and the UE does not maintain the value |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |