**3GPP TSG-RAN WG2 Meeting #110-e *R2-200xxxx***

**Elbonia, 01 – 11 June 2020 R2-2004904**

|  |
| --- |
| *CR-Form-v12.0* |
| **CHANGE REQUEST** |
|  |
|  | **38.331** | **CR** | **1633** | **rev** | **1** | **Current version:** | **15.9.0** |  |
|  |
| *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:***  | Corrections to CORESET and PDCCH TCI state release |
|  |  |
| ***Source to WG:*** | Nokia, Nokia Shanghai Bell |
| ***Source to TSG:*** | R2 |
|  |  |
| ***Work item code:*** | NR\_NewRAT-Core |  | ***Date:*** | 2020-05-22 |
|  |  |  |  |  |
| ***Category:*** | **F** |  | ***Release:*** | 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:*** | Some ambiguities exist in Rel-15 specification regarding the release of CORESETs and PDCCH states as illustrate below.1. The AddModRelease-lists are used for storing/modifying/releasing entries, but in some cases there are nested entries, i.e. AddModRelease inside another AddModRelease. Since the UE stores the child AddMod entries outside the parent AddMod-entry, release of the parent AddMod-entry that originally configured those entries does not release the child AddMod-entries and those need to be released explicitly. In particular, this occurs for PDCCH TCI state IDs inside CORESETs, wherein the CORESETs may be released but the TCI state IDs could remain. However, if all dedicated CORESETs are released, it is not clear whether UE should also retain all the TCI state IDs configured via them, which could lead to configuration mismatch between UE and network. Similar to the TCI state IDs in *PDCCH-Config*, the TCI States configured by *PDSCH-Config* are released when the parent configuration is released.

During the email discussion [AT110e][006], the following was agreed on this: “[006] RAN2 confirms that release of parent field also releases all of the child fields, regardless of whether they have been added via AddModList or as normal fields.”1. CORESETs can be defined in two places: In PDCCH-Config (dedicated CORESETs) and PDCCH-ConfigCommon (CORESET#0 and the additional common CORESET). Since the network may use different identifiers for dedicated and common CORESETs in different cells, it may occur that the CORESET ID of common CORESET coincides with dedicated CORESET ID of another cell. For the sake of example, let’s assume CORESET id = 1 is used: When handover occurs, this may mean that network asks UE to release the dedicated CORESET id = 1 while also configuring common CORESET with CORESET id = 1 and no other configuration change (i.e. SearchSpace entries retain the same CORESET ID as before). This shuold result in UE first releasing the dedicated CORESET and then configuring the newly defined common CORESET, but since the order of common vs. dedicated actions is not defined, UE could also first add the common CORESET and then release CORESET id = 1, which could lead to configuration error as UE would have two CORESETs with id = 1 and the subsequently processed CORESET release could be misinterpreted to apply for both, leading to UE to release both of the CORESETs. This would mean SearchSpace would refer to a CORESET that no longer exists, which causes handover failure.

During the email discussion [AT110e][006], the following was agreed on this: “[006] RAN2 confirms that UE shall not release the PDCCH-ConfigCommon::commonControlResourceSet even if the same CORESET ID is included in PDCCH-Config:: controlResourceSetToReleaseList.”1. Generally, the same as for 1) and 2) applies for all AddModLists, so a general statement in the RRC guidelines could clarify any other cases not spotted here.

During the email discussion [AT110e][006], the following was agreed on this: “[006] RAN2 confirms that it is up to network implementation to release “hanging” fields (i.e. fields referring to other IEs that no longer remain in UE configuration).” |
|  |  |
| ***Summary of change:*** | 1. Clarify that the *PDCCH-Config::controlResourceSetToReleaseList* does not release the *commonControlResourceSet* from PDCCH-ConfigCommon.
2. Clarify in RRC guidelines for ToAddModList that UE shall release the entries added by the AddModList when the parent field is released.

**Impact analysis**Impacted architecture options: EN-DC, NGEN-DC, NR SA, NE-DC, NR-DCImpacted functionality: CORESETs and PDCCH TCI states.Inter-operability: 1. If the network is implemented according to the CR and the UE is not, UE might release fields when network does not expect it, which could lead to connection failure due to RRC configuration mismatch.
2. If the UE is implemented according to the CR and the network is not, the network might expect UE to retain the fields even though UE has released them, which could lead to connection failure due to RRC configuration mismatch.
 |
|  |  |
| ***Consequences if not approved:*** | RRC configuration mismatch may occur as UE and network have different understanding of configured fields, which will lead to connection or handover failures. |
|  |  |
| ***Clauses affected:*** | 6.3.2, A.3.9 |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **x** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **x** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **x** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

*First Modified Subclause*

### 6.3.2 Radio resource control information elements

<UNNECESSARY PARTS OMITTED>

– *PDCCH-Config*

The IE *PDCCH-Config* is used to configure UE specific PDCCH parameters such as control resource sets (CORESET), search spaces and additional parameters for acquiring the PDCCH. If this IE is used for the scheduled cell in case of cross carrier scheduling, the fields other than *searchSpacesToAddModList* and *searchSpacesToReleaseList* are absent.

***PDCCH-Config* information element**

-- ASN1START

-- TAG-PDCCH-CONFIG-START

PDCCH-Config ::= SEQUENCE {

 controlResourceSetToAddModList SEQUENCE(SIZE (1..3)) OF ControlResourceSet OPTIONAL, -- Need N

 controlResourceSetToReleaseList SEQUENCE(SIZE (1..3)) OF ControlResourceSetId OPTIONAL, -- Need N

 searchSpacesToAddModList SEQUENCE(SIZE (1..10)) OF SearchSpace OPTIONAL, -- Need N

 searchSpacesToReleaseList SEQUENCE(SIZE (1..10)) OF SearchSpaceId OPTIONAL, -- Need N

 downlinkPreemption SetupRelease { DownlinkPreemption } OPTIONAL, -- Need M

 tpc-PUSCH SetupRelease { PUSCH-TPC-CommandConfig } OPTIONAL, -- Need M

 tpc-PUCCH SetupRelease { PUCCH-TPC-CommandConfig } OPTIONAL, -- Need M

 tpc-SRS SetupRelease { SRS-TPC-CommandConfig} OPTIONAL, -- Need M

 ...

}

-- TAG-PDCCH-CONFIG-STOP

-- ASN1STOP

|  |
| --- |
| ***PDCCH-Config* field descriptions** |
| ***controlResourceSetToAddModList***List of UE specifically configured Control Resource Sets (CORESETs) to be used by the UE. The network configures at most 3 CORESETs per BWP per cell (including UE-specific and common CORESETs). In case network reconfigures control resource set with the same *ControlResourceSetId* as used for *commonControlResourceSet* configured via *PDCCH-ConfigCommon*, the configuration from *PDCCH-Config* always takes precedence and should not be updated by the UE based on *servingCellConfigCommon*. |
| ***controlResourceSetToReleaseList***List of UE specifically configured Control Resource Sets (CORESETs) to be released by the UE. This field only applies to CORESETs configured by *controlResourceSetToAddModList* and does not release the field *commonControlResourceSet* configured by *PDCCH-ConfigCommon*. |
| ***downlinkPreemption***Configuration of downlink preemption indications to be monitored in this cell (see TS 38.213 [13], clause 11.2). |
| ***searchSpacesToAddModList***List of UE specifically configured Search Spaces. The network configures at most 10 Search Spaces per BWP per cell (including UE-specific and common Search Spaces). |
| ***tpc-PUCCH***Enable and configure reception of group TPC commands for PUCCH. |
| ***tpc-PUSCH***Enable and configure reception of group TPC commands for PUSCH. |
| ***tpc-SRS***Enable and configure reception of group TPC commands for SRS. |

*Next Modified Subclause*

# A.3 PDU specification

<UNNECESSARY PARTS OMITTED>

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 INTEG ER (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. Note that the release of parent field also releases all of the child fields, regardless of whether they have been added via AddModList or as normal fields.

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.