3GPP TSG-RAN WG2 Meeting #129bis R2-250xxxx

Wuhan, China, April 7th – 11th, 2025

**Agenda item:** 8.9.3

**Source:** MediaTek Inc.

**Title:** Report of [Post129][307][R19 IoT NTN] CB-msg3/CB-msg4 (Mediatek)

**Document for:**  Discussion and decision

# Introduction

This is the report of the following offline discussion from RAN2#129:

* **[Post129][307][R19 IoT NTN] CB-msg3/CB-msg4 (Mediatek)**

Scope: discuss CB-msg3 resource configuration parameters and CB-msg4 monitoring window and RNTI design

Intended outcome: summary of the email discussion

Deadline: long

The deadline for long email discussion is March 21st 10:00 UTC.

# Contact Points

Respondents to the email discussion are kindly asked to fill in the following table.

|  |  |  |
| --- | --- | --- |
| Company | Name | Email Address |
| MediaTek (Rapp) | Felix Tsai | chun-fan.tsai@mediatek.com |
| Nokia | Ping Yuan | Ping.1.yuan@nokia-sbell.com |
| NEC | Yue Zhou | zhou\_yue@nec.cn |
| ZTE | Lu Ting | lu.ting@zte.com.cn |
| Qualcomm | Bharat | bshrestha@qti.qualcomm.com |
| Ericsson | Robert | robert.s.karlsson AT ericsson.com |
| CATT | Xiao XIAO | xiaoxiao@catt.cn |
| Huawei, HiSilicon | Xubin | xubin10@huawei.com |
| vivo | Mo Yitao (Stephen) | Yitao.mo@vivo.com |
| ETRI | Seungkwon Baek | skback@etri.re.kr |
| Xiaomi | Haitao Li | Lihaitao11@xiaomi.com |
| Apple | Yuqin Chen | yuqin\_chen@apple.com |
|  |  |  |

# Discussion

## CB-Msg3 resource configuration parameters

Regarding the CB-Msg3 resource configuration, the following agreements have been reached during the RAN2#129 meeting.

RAN2 #129 Agreements:

1. RAN2 assumes that at least the following will be part of the shared resources configuration for CB-msg3 (FFS on other aspects)

- Time domain resources for (N)PUSCH occasions: periodicity and start time (e.g., start subframe, start SFN)

- Frequency domain resources for (N)PUSCH occasions

- repetition number

- (N)PDCCH resource

- MCS

2. For CB-msg3 transmission, for eMTC NTN, up to three separate RSRP thresholds (on top of the minimum RSRP threshold and possibly different from the thresholds for PRACH) can be supported for achieving at most 4 CE levels; for NB-IoT NTN, up to two separate RSRP thresholds (on top of the minimum RSRP threshold possibly different from the thresholds for PRACH) can be supported for achieving at most 3 repetition levels.

3. The CB EDT Config has one minimum RSRP threshold (as agreed in RAN2#128) to use CB EDT.

4. The CB EDT Config has two RSRP thresholds for NB-IoT for the three CE levels.

5. CB EDT Config has three RSRP thresholds for eMTC for the four CE levels.

6. As Signalling design Baseline RAN2 assumes the PUR config and the NPRACH config for shared (N)PUSCH config can be used and some of the parameters can be included in a new CB EDT config.

7. RAN2 consider a new CBEDT-ConfigSIB-NB IE for configuring the CB EDT feature

Working assumption:

1. For CB-MSG3, the Transmission window is configured by the network with a starting point (e.g. H-SFN offset), a window length, and a window periodicity (window length and periodicity could be the same). For k=1 the window length can be equal to 1: same behaviour as today

TP for CB-Msg3 resource parameters for both eMTC and NB-IoT are provided in next two sessions. The TP is based on current agreements and mainly reuse from **PRACH** and **PUR configuration** parameters. In the TP, the IEs that are introduced based on RAN2 agreement or work assumption are marked with green; the IEs copied from PUR configuration are marked with yellow; the IEs copied from PRACH configuration are marked with cyan. The current PRACH and PUR parameters are also copied for reference.

### eMTC

#### – *PUR-Config*

The IE *PUR-Config* is used to specify the PUR configuration.

*PUR-Config* information element

-- ASN1START

PUR-Config-r16 ::= SEQUENCE {

pur-ConfigID-r16 PUR-ConfigID-r16 OPTIONAL, -- Need OR

pur-ImplicitReleaseAfter-r16 ENUMERATED {n2, n4, n8, spare} OPTIONAL, -- Need OR

pur-StartTimeParameters-r16 SEQUENCE {

periodicityAndOffset-r16 PUR-PeriodicityAndOffset-r16,

startSFN-r16 INTEGER (0..1023),

startSubFrame-r16 INTEGER (0..9),

hsfn-LSB-Info-r16 BIT STRING (SIZE(1))

} OPTIONAL, --Need ON

pur-NumOccasions-r16 ENUMERATED {one, infinite},

pur-RNTI-r16 C-RNTI OPTIONAL, -- Need ON

pur-TimeAlignmentTimer-r16 INTEGER (1..8) OPTIONAL, -- Need OR

pur-RSRP-ChangeThreshold-r16 SetupRelease {PUR-RSRP-ChangeThreshold-r16} OPTIONAL, -- Need ON

pur-ResponseWindowTimer-r16 ENUMERATED {sf240, sf480, sf960, sf1920, sf3840, sf5760, sf7680, sf10240} OPTIONAL, -- Need ON

pur-MPDCCH-Config-r16 PUR-MPDCCH-Config-r16 OPTIONAL, -- Need ON

pur-PDSCH-FreqHopping-r16 BOOLEAN,

pur-PUCCH-Config-r16 PUR-PUCCH-Config-r16 OPTIONAL, -- Need ON

pur-PUSCH-Config-r16 PUR-PUSCH-Config-r16 OPTIONAL, -- Need ON

...,

[[ pur-PDSCH-maxTBS-r17 BOOLEAN OPTIONAL -- Need ON

]]

}

PUR-MPDCCH-Config-r16 ::= SEQUENCE {

mpdcch-FreqHopping-r16 BOOLEAN,

mpdcch-Narrowband-r16 INTEGER (1..maxAvailNarrowBands-r13),

mpdcch-PRB-PairsConfig-r16 SEQUENCE{

numberPRB-Pairs-r16 ENUMERATED {n2, n4, n6, spare1},

resourceBlockAssignment-r16 BIT STRING (SIZE(4))

},

mpdcch-NumRepetition-r16 ENUMERATED {r1, r2, r4, r8, r16, r32, r64, r128, r256},

mpdcch-StartSF-UESS-r16 CHOICE {

fdd ENUMERATED {v1, v1dot5, v2, v2dot5, v4, v5, v8, v10},

tdd ENUMERATED {v1, v2, v4, v5, v8, v10, v20, spare1}

},

mpdcch-Offset-PUR-SS-r16 ENUMERATED {zero, oneEighth, oneQuarter,

threeEighth, oneHalf, fiveEighth,

threeQuarter, sevenEighth}

}

PUR-PUCCH-Config-r16 ::= SEQUENCE {

n1PUCCH-AN-r16 INTEGER (0..2047) OPTIONAL, -- Need ON

pucch-NumRepetitionCE-Format1-r16 ENUMERATED {n1, n2, n4, n8} OPTIONAL -- Need ON

}

PUR-PUSCH-Config-r16 ::= SEQUENCE {

pur-GrantInfo-r16 CHOICE {

ce-ModeA SEQUENCE {

numRUs-r16 BIT STRING (SIZE(2)),

prb-AllocationInfo-r16 BIT STRING (SIZE(10)),

mcs-r16 BIT STRING (SIZE(4)),

numRepetitions-r16 BIT STRING (SIZE(3))

},

ce-ModeB SEQUENCE {

subPRB-Allocation-r16 BOOLEAN,

numRUs-r16 BOOLEAN,

prb-AllocationInfo-r16 BIT STRING (SIZE(8)),

mcs-r16 BIT STRING (SIZE(4)),

numRepetitions-r16 BIT STRING (SIZE(3))

}

} OPTIONAL, -- Need ON

pur-PUSCH-FreqHopping-r16 BOOLEAN,

p0-UE-PUSCH-r16 INTEGER (-8..7),

alpha-r16 Alpha-r12,

pusch-CyclicShift-r16 ENUMERATED {n0, n6},

pusch-NB-MaxTBS-r16 BOOLEAN,

locationCE-ModeB-r16 INTEGER (0..5) OPTIONAL -- Cond SubPRB

}

PUR-RSRP-ChangeThreshold-r16 ::= SEQUENCE {

increaseThresh-r16 RSRP-ChangeThresh-r16,

decreaseThresh-r16 RSRP-ChangeThresh-r16 OPTIONAL --Need OP

}

RSRP-ChangeThresh-r16 ::= ENUMERATED {dB4, dB6, dB8, dB10, dB14, dB18, dB22, dB26, dB30, dB34, spare6, spare5, spare4, spare3, spare2, spare1}

-- ASN1STOP

#### – *PRACH-Config*

The IE *PRACH-ConfigSIB* and IE *PRACH-Config* are used to specify the PRACH configuration in the system information and in the mobility control information, respectively.

*PRACH-Config* information elements

-- ASN1START

PRACH-ConfigSIB ::= SEQUENCE {

rootSequenceIndex INTEGER (0..837),

prach-ConfigInfo PRACH-ConfigInfo

}

PRACH-ConfigSIB-v1310 ::= SEQUENCE {

rsrp-ThresholdsPrachInfoList-r13 RSRP-ThresholdsPrachInfoList-r13,

mpdcch-startSF-CSS-RA-r13 CHOICE {

fdd-r13 ENUMERATED {v1, v1dot5, v2, v2dot5, v4, v5, v8,

v10},

tdd-r13 ENUMERATED {v1, v2, v4, v5, v8, v10, v20, spare}

} OPTIONAL, -- Cond MP

prach-HoppingOffset-r13 INTEGER (0..94) OPTIONAL, -- Need OR

prach-ParametersListCE-r13 PRACH-ParametersListCE-r13

}

PRACH-ConfigSIB-v1530 ::= SEQUENCE {

edt-PRACH-ParametersListCE-r15 SEQUENCE (SIZE(1..maxCE-Level-r13)) OF EDT-PRACH-ParametersCE-r15

}

PRACH-Config ::= SEQUENCE {

rootSequenceIndex INTEGER (0..837),

prach-ConfigInfo PRACH-ConfigInfo OPTIONAL -- Need ON

}

PRACH-Config-v1310 ::= SEQUENCE {

rsrp-ThresholdsPrachInfoList-r13 RSRP-ThresholdsPrachInfoList-r13 OPTIONAL, -- Cond MP

mpdcch-startSF-CSS-RA-r13 CHOICE {

fdd-r13 ENUMERATED {v1, v1dot5, v2, v2dot5, v4, v5, v8,

v10},

tdd-r13 ENUMERATED {v1, v2, v4, v5, v8, v10, v20, spare}

} OPTIONAL, -- Cond MP

prach-HoppingOffset-r13 INTEGER (0..94) OPTIONAL, -- Need OR

prach-ParametersListCE-r13 PRACH-ParametersListCE-r13 OPTIONAL, -- Cond MP

initial-CE-level-r13 INTEGER (0..3) OPTIONAL -- Need OR

}

PRACH-Config-v1430 ::= SEQUENCE {

rootSequenceIndexHighSpeed-r14 INTEGER (0..837),

zeroCorrelationZoneConfigHighSpeed-r14 INTEGER (0..12),

prach-ConfigIndexHighSpeed-r14 INTEGER (0..63),

prach-FreqOffsetHighSpeed-r14 INTEGER (0..94)

}

PRACH-ConfigSCell-r10 ::= SEQUENCE {

prach-ConfigIndex-r10 INTEGER (0..63)

}

PRACH-ConfigInfo ::= SEQUENCE {

prach-ConfigIndex INTEGER (0..63),

highSpeedFlag BOOLEAN,

zeroCorrelationZoneConfig INTEGER (0..15),

prach-FreqOffset INTEGER (0..94)

}

PRACH-ParametersListCE-r13 ::= SEQUENCE (SIZE(1..maxCE-Level-r13)) OF PRACH-ParametersCE-r13

PRACH-ParametersCE-r13 ::= SEQUENCE {

prach-ConfigIndex-r13 INTEGER (0..63),

prach-FreqOffset-r13 INTEGER (0..94),

prach-StartingSubframe-r13 ENUMERATED {sf2, sf4, sf8, sf16, sf32, sf64, sf128,

sf256} OPTIONAL, -- Need OP

maxNumPreambleAttemptCE-r13

ENUMERATED {n3, n4, n5, n6, n7, n8, n10} OPTIONAL, -- Need OP

numRepetitionPerPreambleAttempt-r13 ENUMERATED {n1,n2,n4,n8,n16,n32,n64,n128},

mpdcch-NarrowbandsToMonitor-r13 SEQUENCE (SIZE(1..2)) OF

INTEGER (1..maxAvailNarrowBands-r13),

mpdcch-NumRepetition-RA-r13 ENUMERATED {r1, r2, r4, r8, r16,

r32, r64, r128, r256},

prach-HoppingConfig-r13 ENUMERATED {on,off}

}

EDT-PRACH-ParametersCE-r15 ::= SEQUENCE {

edt-PRACH-ParametersCE-r15 SEQUENCE {

prach-ConfigIndex-r15 INTEGER (0..63),

prach-FreqOffset-r15 INTEGER (0..94),

prach-StartingSubframe-r15 ENUMERATED {sf2, sf4, sf8, sf16, sf32, sf64, sf128, sf256} OPTIONAL, -- Need OP

mpdcch-NarrowbandsToMonitor-r15 SEQUENCE (SIZE(1..2)) OF INTEGER (1..maxAvailNarrowBands-r13)

} OPTIONAL -- Need OR

}

RSRP-ThresholdsPrachInfoList-r13 ::= SEQUENCE (SIZE(1..3)) OF RSRP-Range

PRACH-TxDuration-r17::= SEQUENCE {

prach-TxDuration-r17 ENUMERATED {n1, n2, n4, n8, n16, n32, n64, n128}

}

-- ASN1STOP

Here is the TP for eMTC CB-Msg3 configuration parameters.

#### – CB-Msg3*-ConfigSIB*

The IE *CB-Msg3-Config* is used to specify the CB-Msg3 configuration.

*CB-Msg3-ConfigSIB* information element

-- ASN1START

CB-Msg3-ConfigSIB-r19 ::= SEQUENCE {

cb-Msg3-MinRSRP-Threshold-r19 NRSRP-Range OPTIONAL, --Need OR

cb-Msg3-RSRP-ThresholdList-r19 CB-Msg3-RSRP-ThresholdList-r19 OPTIONAL, --Need OP

cb-Msg3-ParametersList-r19 CB-Msg3-ParametersList-r19,

cb-Msg3-ResponseWindowTimer-r19 ENUMERATED {FFS},

}

CB-Msg3-ParametersList-r19 ::= SEQUENCE (SIZE (1.. maxCB-Msg3-Resources-r19)) OF

CB-Msg3-Parameters-r19

CB-Msg3-Parameters-r19 ::= SEQUENCE {

cb-Msg3-NumReplicas INTEGER(1..4),

cb-Msg3-DSATransmissionWindow-r19 SEQUENCE {

startSFN-19 INTEGER (0.. 1023),

windowSize-19 ENUMERATED {FFS},

windowPeriodicity-r19 ENUMERATED {FFS}

} OPTIONAL, --Need OP

cb-Msg3-StartTimeParameters-r19 SEQUENCE {

pusch-periodicity-r19 ENUMERATED {FFS},

pusch-startTime-r19 INTEGER (0..1023),

pusch-startSubframe-r19 INTEGER (0..9)

}

cb-Msg3-MPDCCH-Config-r19 CB-Msg3-MPDCCH-Config-r19,

cb-Msg3-PUCCH-Config-r19 CB-Msg3-PUCCH-Config-r19,

cb-Msg3-PUSCH-Config-r19 CB-Msg3-PUSCH-Config-r19,

cb-Msg3-PDSCH-Config-r19 CB-Msg3-PDSCH-Config-r19,

cb-Msg3-TBS-r19 ENUMERATED {FFS},

...

}

CB-MSG3-MPDCCH-Config-r19 ::= SEQUENCE {

mpdcch-FreqHopping-r19 BOOLEAN,

mpdcch-Narrowband-r19 INTEGER (1..maxAvailNarrowBands-r13),

mpdcch-PRB-PairsConfig-r19 SEQUENCE{

numberPRB-Pairs-r19 ENUMERATED {n2, n4, n6, spare1},

resourceBlockAssignment-r19 BIT STRING (SIZE(4))

},

mpdcch-NumRepetition-r19 ENUMERATED {r1, r2, r4, r8, r16, r32, r64, r128, r256},

mpdcch-StartSF-UESS-r19 ENUMERATED {v1, v1dot5, v2, v2dot5, v4, v5, v8, v10},

mpdcch-Offset-SS-r19 ENUMERATED {zero, oneEighth, oneQuarter,

threeEighth, oneHalf, fiveEighth,

threeQuarter, sevenEighth}

}

CB-Msg3-PUCCH-Config-r19 ::= SEQUENCE {

n1PUCCH-AN-r19 INTEGER (0..2047), OPTIONAL, -- Need ON

pucch-NumRepetitionCE-Format1-r19 ENUMERATED {n1, n2, n4, n8} OPTIONAL -- Need ON

}

CB-Msg3-PUSCH-Config-r19 ::= SEQUENCE {

numRUs-r19 BIT STRING (SIZE(2)),

prb-AllocationInfo-r19 BIT STRING (SIZE(10)),

mcs-r19 BIT STRING (SIZE(4)),

numRepetitions-r19 BIT STRING (SIZE(3))

PUSCH-FreqHopping-r19 BOOLEAN,

p0-UE-PUSCH-r19 INTEGER (-8..7),

alpha-r19 Alpha-r12

}

CB-Msg3-PDSCH-Config-r19 ::= SEQUENCE {

cb-Msg3-PDSCH-FreqHopping-r19 BOOLEAN,

cb-Msg3-PDSCH-maxTBS-r19 BOOLEAN

}

CB-Msg3-RSRP-ThresholdList-r19 ::= SEQUENCE (SIZE (1..3)) OF RSRP-Range

maxCB-Msg3-Resources-r19 INTEGER ::= 4 -- Maximum number of CB-Msg3 resources

-- ASN1STOP

Notes

* CB-Msg3-ConfigSIB: This IE is assumed to be in **SIB2**.
* cb-Msg3-MinRSRP-Threshold-r19: It is assumed as an optional IE. If it is absent, no minimal threshold is used.
* cb-Msg3-RSRP-ThresholdList-r19: It is assumed as an optional IE. If this IE is absent, the thresholds list of PRACH is used.
* cb-Msg3-DSATransmissionWindow-r19: It is assumed as an optional IE. If the number of the replicas is one, the DSA transmission window is not needed.
  + startSFN-19 in cb-Msg3-DSATransmissionWindow-r19: Although the term *H-SFN offset* is used in the agreement, the rapporteur believes it should be a SFN offset during the H-SFN duration.
  + *hsfn-LSB-Info-r16* in PUR-Config: This IE is not used as the periodicity of CB-Msg3 resource is assumed shorted than a H-SFN duration (i.e., 10.24s).
* CB-MSG3-MPDCCH-Config-r19:
  + mpdcch-StartSF-UESS-r19: The configuration for TDD is not used, as only FDD eMTC is supported for IoT NTN.
  + Other parameters in are copied from IE *PUR-MPDCCH-Config-r16*.
* CB-Msg3-PUSCH-Config-r19:
  + According to the RAN1 reply LS [R1-2407548](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_118/Docs/R1-2407548.zip) on TA validation, only CE mode A is confirmed. **Whether the CE mode B is supported is still FFS**. Therefore, only the PUSCH configuration for only CE mode A is introduced for now.
  + *pusch-CyclicShift-r16*: This parameter is not used because this parameter is specified for PUR only.
  + *pusch-NB-MaxTBS-r16*: This parameter is not used as a maximum TBS is provided.
  + *locationCE-ModeB-r16*: This parameter is not used as support of CE mode B is FFS.
* ~~CB-Msg3-PDSCH-Config-r19~~ CB-Msg3-PUCCH-Config-r19
  + Two fields copied from *PUR-PUCCH-Config-r16*. ~~Note it is assumed that the fields are mandatory as there is no dedicated configuration from connected mode in CB-MSG3 procedure~~.
* cb-Msg3-TBS-r19: Whether the maximum TBS is CE level specific is FFS. Rapporteur assumes it is CE level specific.

Companies are invited to provide comments on the above TP including but not limited to below discussion points.

* Which SIB should be used for IE CB-Msg3*-ConfigSIB* ?
* Any parameter is missing in the TP and why this should be added?
* Any parameter should be removed from the TP?
* Most value range of the parameters are copied from PUR parameters, any further change needed?

**Q1: Any comments on the TP of *CB-Msg3-ConfigSIB?***

|  |  |
| --- | --- |
| **Company** | **Comments** |
| Nokia | On the TP itself, we have below comments:   1. For the IE *cb-Msg3-MinRSRP-Threshold-r19*, for modelling simplicity, it can be merged into the list of *cb-Msg3-RSRP-ThresholdList-r19 (e.g.,* as the entry in the list to determine the most robust CE level). i.e., the number of elements in the list indicates the number of supported CE level. The value of each element is the RSRP threshold for the corresponding CE level. 2. For the “NRSRP-Range”, it should be “RSRP-Range” since the IE is for eMTC instead of NB-IoT.   [ZTE comments] Agree   1. For the “CB-Msg3-ParametersList-r19 ::= SEQUENCE (SIZE (1.. *maxCB-Msg3-Resources-r19*))”, should it be (1…*maxCE-Level*)? We understand the CB-Msg3-Parameters-r19 is a per-CE level configuration.   [ZTE comments] Agree, *maxCE-Level-r13* is for eMTC and can be reused.   1. For the “cb-Msg3-NumReplicas INTEGER(1..4)”, in principle it can be an optional parameter. If absent, it then means 1 (stand-alone case).   [ZTE comments] Agree with Nokia to define *cb-Msg3-NumReplicas* as optional and it can be absent.  [Qualcomm] Disagree, this does not save signaling, Just 2 bits mandatory is better.   1. For the MPDCCH search space for CB-Msg4, we think it should be checked and confirmed by RAN1. 2. For the field description on “cb-Msg3-DSATransmissionWindow-r19:... If the number of the replicas is one, the DSA transmission window *is not needed*”, the wording “not needed” should be “not applied”.   [ZTE comments] Agree. Furthermore, we also have a general suggestion to organize all the DSA related parameters into a separate IE, e.g., *cb-Msg3-DSAConfig-R19* which can only be present when cb-Msg3-NumReplicas is present or *cb-Msg3-NumReplicas* is larger than 1.  Additional parameters:   * The parameters for CB-Msg3 fallback to 4-step RACH/EDT can be considered. Furthermore, the power ramping step parameters can be considered if the UE transmits CB-Msg3 multiple times.   [ZTE comments] We also think some configuration for backoff and/or fallback schemes may be needed. But firstly we need to discuss these two schemes. |
| NEC | We are generally fine.   1. Typographical Errors: In the IE *cb-Msg3-DSATransmissionWindow-r19*, the parameters *startSFN-19* and *windowSize-19* should be revised to ***startSFN-r19*** and ***windowSize-r19***, respectively 2. *cb-Msg3-DSATransmissionWindow-r19*: We agree that *cb-Msg3-DSATransmissionWindow-r19* is optional. Since a UE can derive *windowSize-r19* and *windowPeriodicity-r19* from the configured *cb-Msg3-NumReplicas* and *cb-Msg3-StartTimeParameters-r19*, these parameters **MAY** be optional. The final determination of *windowSize-r19* and *windowPeriodicity-r19* remains **FFS for RAN2**. |
| ZTE | We have the following further comments for eMTC CB-Msg3 configuration:   1. We slightly prefer a separate *cb-Msg3-MinRSRP-Threshold-r19* from *cb-Msg3-RSRP-ThresholdList-r19* (naming can be improved, if needed). The main consideration is that *cb-Msg3-MinRSRP-Threshold-r19* may be used for condition checking for initiating transmission using CB-Msg3 which can be captured in RRC. Meanwhile, *cb-Msg3-RSRP-ThresholdList-r19* may be used in another section for CB-Msg3 resources selection (either in RRC or in MAC spec). 2. For “*cb-Msg3-DSATransmissionWindow-r19*”, we assume this time window is divided continuously along the timeline. So it only needs one of these two parameters, *windowSize-19* and *windowPeriodicity-r19*. We slightly prefer to keep *windowSize-19.* Moreover, we see it’s simple to align the start of DSA transmission window with the start of time-domain resources. In other word, we see no clear motivation to configure the start of DSA transmission window in the middle of a cycle of time-domain resources. So it may be possible to skip *startSFN-19* and make *pusch-startTime-r19* applied*.* Finally, we also see it’s feasible or beneficial to define *windowSize-19* with the unit of *pusch-periodicity-r19*, e.g., *windowSize-19* equals to one or more *pusch-periodicity-r19*. 3. For *CB-Msg3-PUSCH-Config-r19*, we understand it only focus on the frequency-domain resource configuration for PUSCH. Please not the time-domain resource configuration for PUSCH is in *cb-Msg3-StartTimeParameters*. Generally, we think *cb-Msg3-StartTimeParameters* can also be included in the *CB-Msg3-PUSCH-Config-r19.* Moreover, current definition way for PUSCH frequency-domain resource configuration follows the PUR configuration. However, PUR-Config is dedicated and UE-specific resource configuration and the eNB generally needs to know UE capability before it can provide such configuration. Therefore, the current definition way for PUSCH frequency-domain resource configuration is not suitable to CB-Msg3 PUSCH. Based on our understanding, at least *prb-AllocationInfo* can be defined as a “set” format with intention to provide a set of or shared frequency-domain resources. The following alternative can be considered:   ......  cb-Msg3-MPDCCH-Config-r19 CB-Msg3-MPDCCH-Config-r19,  cb-Msg3-PUCCH-Config-r19 CB-Msg3-PUCCH-Config-r19,  cb-Msg3-PUSCH-Config-r19 CB-Msg3-PUSCH-Config-r19,  cb-Msg3-PDSCH-Config-r19 CB-Msg3-PDSCH-Config-r19,  ........  CB-Msg3-PUSCH-Config-r19 ::= SEQUENCE {  cb-Msg3-StartTimeParameters-r19 SEQUENCE {  pusch-periodicity-r19 ENUMERATED {FFS},  pusch-startTime-r19 INTEGER (0..1023),  pusch-startSubframe-r19 INTEGER (0..9)  },  cb-Msg3-PUSCH-FreqConfig-r19 CB-Msg3-PUSCH-FreqConfig-r19  }  CB-Msg3-PUSCH-FreqConfig-r19::= SEQUENCE {  numRUs-r19 BIT STRING (SIZE(2)),  prb-AllocationInfoSet-r19 SEQUENCE (SIZE(1..FFS)) OF BIT STRING (SIZE(10)),  mcs-r19 BIT STRING (SIZE(4)),  numRepetitions-r19 BIT STRING (SIZE(3))  PUSCH-FreqHopping-r19 BOOLEAN,  p0-UE-PUSCH-r19 INTEGER (-8..7),  alpha-r19 Alpha-r12  }   1. For *CB-MSG3-MPDCCH-Config-r19*, we think it's also not suitable to configure only one narrow band. In other word, the narrow bands for monitoring MPDCCH for Msg4 also needs to be configured as a "set" format, similar as the MPDCCH configuration for RAR in PRACH-Config. An example is as below:   CB-MSG3-MPDCCH-Config-r19 ::= SEQUENCE {  mpdcch-FreqHopping-r19 BOOLEAN,    mpdcch-NarrowbandSet-r19 SEQUENCE (SIZE(1..FFS)) OF INTEGER (1..maxAvailNarrowBands-r13),  ......  How the UE selects the corresponding narrow band to monitor Msg4 can adopt a similar approach to how the UE selects the narrowband for monitoring MPDCCH for RAR. |
| Qualcomm | Same as EDT, this can be in SIB2.  For now we should confirm the parameters we need. Whether configurations for different narrowband are supported can be checked with RAN1. We should also check with RAN1 on any missing parameter of any parameters that are not needed.  We think this cb-Msg3-DSATransmissionWindow-r19 should not be optional as per working assumption. The way two periodicities one for window and one for PUSCH are defined is little confusing as the window periodicity should not be smaller than window size leading to overlapping windows. In addition, there may be PUSCH resource configured outside window and wasted. I think we need to configure this efficient way. |
| Samsung | 1. Agree with ZTE that the condition to enter CB-EDT should be separate from the list of CE thresholds. It would be odd to have a list of thresholds, where the thresholds are actually used for separate parts of the procedure. I would also drop the name “min” from the ASN1 name.  2. For the following:  p0-UE-PUSCH-r19 INTEGER (-8..7),  alpha-r19 Alpha-r12  These are power control parameters copied straight from PUR. We have not discussed power control yet, but our understanding is that since it is contention-based, then it should be possible to configure power ramping. Otherwise we risk certain UEs being starved.  3. The current encoding seems to be only for CE mode A, and does not include a CE mode B option as in PUR. We probably need to make an agreement on excluding CE mode B.  4. Besides that:  - SIB2 is used, just like EDT. No new SIB is introduced for this (would be happy to exclude this)  - We expect the value ranges will have to be checked by RAN1. |
| Ericsson | The minimum RSRP threshold shall not be merged into the list for CE levels, because the minimum is likely to be captured in the triggering for EDT in 36.331 5.3.3.1b while the CE levels will be captured in MAC.  This  cb-Msg3-DSATransmissionWindow-r19: It is assumed as an optional IE. If the number of the replicas is one, the DSA transmission window is not needed.  Is not optional and is always needed. The resource allocation for CB PUSCH resources shall be a separate allocation. Then DSA window just divide the full set of CB PSUCH resources in to DSA windows. Then SA can have the periodicity of 1 – which will make all spec text in MAC easier as there will be no difference between SA and DSA.  Further, we think we must have H-SFH in the DSA window in case the config is received close in time to a SFN wrap-around and the window do not evenly divide 1024 (H-SFN wrap around maybe also needs to be considered?).  Many parameters have not been discussed yet, and they need to be before including in a CR.  SIB2 is fine. |
| CATT | 1. We are open to discuss the way as commented by Nokia, i.e. no need to introduce a separate field. This implies revisit of agreement “The CB EDT Config has one minimum RSRP threshold (as agreed in RAN2#128) to use CB EDT.”. 2. For the parameter *maxCB-Msg3-Resources-r19*, we have the same view with Nokia, i.e. this parameter should just be the *maxCE-Level* instead. We have the agreement of “CB-msg3 EDT cell specific PUSCH resources for Msg3 transmission are provided **per CE level**”, and each CB-Msg3-Parameters-r19 includes the set of resources and the associated transmission parameters used for CB-Msg3 EDT. The resources and associated transmission parameters should all be configured at per CE level.   For the fallback case mentioned by Nokia above, we think we can consider this after we achieve necessary agreements, e.g. for the power ramping case, similar mechanism in eMTC and NB-IoT can be considered. |
| Huawei, HiSilicon | We are generally fine.  In addition, we agree with the fourth comment by Nokia and the 2nd comment by ZTE.  Besides，one more general comment: we assume this is still an initial discussion on the TP/signalling structure, we should focus on the essential parameters first and they have to be confirmed by RAN1. So for the parameters without being supported by discussion or agreement, we need to further discuss. |
| vivo | We are also generally fine with the proposed TP and aslo think that power ramping parameters and the maximum number of CB-Msg3 attempts are needed if reattempt and fallback are supported. |
| Apple | We are generally fine but also agree this should be confirmed by RAN1. The fallback issue mentioned by Nokia should be first discussed in RAN2, though we have a sympathy for it. |
|  |  |
|  |  |

### NB-IoT part

#### – *PUR-Config-NB*

The IE *PUR-Config-NB* is used to specify PUR configuration.

*PUR-Config-NB* information element

-- ASN1START

PUR-Config-NB-r16 ::= SEQUENCE {

pur-ConfigID-r16 PUR-ConfigID-NB-r16 OPTIONAL, --Need OR

pur-TimeAlignmentTimer-r16 INTEGER (1..8) OPTIONAL, --Need OR

pur-NRSRP-ChangeThreshold-r16 SetupRelease {PUR-NRSRP-ChangeThreshold-NB-r16}

OPTIONAL, --Need ON

pur-ImplicitReleaseAfter-r16 ENUMERATED {n2, n4, n8, spare} OPTIONAL, --Need OR

pur-RNTI-r16 C-RNTI OPTIONAL, --Need ON

pur-ResponseWindowTimer-r16 ENUMERATED {pp1, pp2, pp3, pp4, pp8, pp16, pp32, pp64}

OPTIONAL, --Need ON

pur-StartTimeParameters-r16 SEQUENCE {

periodicityAndOffset-r16 PUR-PeriodicityAndOffset-NB-r16,

startSFN-r16 INTEGER (0..1023),

startSubframe-r16 INTEGER (0..9),

hsfn-LSB-Info-r16 BIT STRING (SIZE(1))

} OPTIONAL, --Need ON

pur-NumOccasions-r16 ENUMERATED {one, infinite},

pur-PhysicalConfig-r16 SEQUENCE {

carrierConfig-r16 CarrierConfigDedicated-NB-r13,

npusch-NumRUsIndex-r16 INTEGER (0..7),

npusch-NumRepetitionsIndex-r16 INTEGER (0..7),

npusch-SubCarrierSetIndex-r16 CHOICE {

khz15 INTEGER (0..18),

khz3dot75 INTEGER (0..47)

},

npusch-MCS-r16 CHOICE {

singleTone INTEGER (0..10),

multiTone INTEGER (0..13)

},

p0-UE-NPUSCH-r16 INTEGER (-8..7),

alpha-r16 ENUMERATED {al0, al04, al05, al06,

al07, al08, al09, al1},

npusch-CyclicShift-r16 ENUMERATED {n0, n6},

npdcch-Config-r16 NPDCCH-ConfigDedicated-NB-r13

} OPTIONAL, -- Need ON

...,

[[

pur-PhysicalConfig-v1650 SEQUENCE {

ack-NACK-NumRepetitions-r16 ACK-NACK-NumRepetitions-NB-r13

} OPTIONAL --Need ON

]],

[[

pur-PhysicalConfig-v1700 SEQUENCE {

pur-UL-16QAM-Config-r17 SetupRelease {PUR-UL-16QAM-Config-NB-r17} OPTIONAL, -- Need ON

pur-DL-16QAM-Config-r17 SetupRelease {NPDSCH-16QAM-Config-NB-r17} OPTIONAL -- Need ON

} OPTIONAL -- Need ON

]]

}

PUR-NRSRP-ChangeThreshold-NB-r16 ::= SEQUENCE {

increaseThresh-r16 NRSRP-ChangeThresh-NB-r16,

decreaseThresh-r16 NRSRP-ChangeThresh-NB-r16 OPTIONAL --Need OP

}

PUR-UL-16QAM-Config-NB-r17 ::= SEQUENCE {

uplinkPowerControlDedicated-r17 UplinkPowerControlDedicated-NB-v1700

}

NRSRP-ChangeThresh-NB-r16 ::= ENUMERATED {dB4, dB6, dB8, dB10, dB14, dB18, dB22, dB26, dB30, dB34, spare6, spare5, spare4, spare3, spare2, spare1}

-- ASN1STOP

#### – *NPRACH-ConfigSIB-NB*

The IE *NPRACH-ConfigSIB-NB* is used to specify the NPRACH configuration for the anchor and non-anchor carriers.

*NPRACH-ConfigSIB-NB* information elements

-- ASN1START

NPRACH-ConfigSIB-NB-r13 ::= SEQUENCE {

nprach-CP-Length-r13 ENUMERATED {us66dot7, us266dot7},

rsrp-ThresholdsPrachInfoList-r13 RSRP-ThresholdsNPRACH-InfoList-NB-r13 OPTIONAL, -- Need OR

nprach-ParametersList-r13 NPRACH-ParametersList-NB-r13

}

NPRACH-ConfigSIB-NB-v1330 ::= SEQUENCE {

nprach-ParametersList-v1330 NPRACH-ParametersList-NB-v1330

}

NPRACH-ConfigSIB-NB-v1450 ::= SEQUENCE {

maxNumPreambleAttemptCE-r14 ENUMERATED {n3, n4, n5, n6, n7, n8, n10, spare1}

}

NPRACH-ConfigSIB-NB-v1530 ::= SEQUENCE {

tdd-Parameters-r15 SEQUENCE {

nprach-PreambleFormat-r15 ENUMERATED {

fmt0, fmt1, fmt2, fmt0-a, fmt1-a},

dummy ENUMERATED {

n1, n2, n4, n8, n16, n32, n64, n128,

n256, n512, n1024},

nprach-ParametersListTDD-r15 NPRACH-ParametersListTDD-NB-r15

} OPTIONAL, -- Cond TDD

fmt2-Parameters-r15 SEQUENCE {

nprach-ParametersListFmt2-r15 NPRACH-ParametersListFmt2-NB-r15 OPTIONAL, -- Need OR

nprach-ParametersListFmt2EDT-r15 NPRACH-ParametersListFmt2-NB-r15 OPTIONAL -- Cond EDT2

} OPTIONAL, -- Need OR

edt-Parameters-r15 SEQUENCE {

edt-SmallTBS-Subset-r15 ENUMERATED {true} OPTIONAL, -- Need OR

edt-TBS-InfoList-r15 EDT-TBS-InfoList-NB-r15,

nprach-ParametersListEDT-r15 NPRACH-ParametersList-NB-r14 OPTIONAL -- Need OR

} OPTIONAL -- Cond EDT1

}

NPRACH-ConfigSIB-NB-v1550 ::= SEQUENCE {

tdd-Parameters-v1550 SEQUENCE {

nprach-ParametersListTDD-v1550 NPRACH-ParametersListTDD-NB-v1550

}

}

NPRACH-ParametersList-NB-r13 ::= SEQUENCE (SIZE (1.. maxNPRACH-Resources-NB-r13)) OF NPRACH-Parameters-NB-r13

NPRACH-ParametersList-NB-v1330 ::= SEQUENCE (SIZE (1.. maxNPRACH-Resources-NB-r13)) OF NPRACH-Parameters-NB-v1330

NPRACH-Parameters-NB-r13::= SEQUENCE {

nprach-Periodicity-r13 ENUMERATED {ms40, ms80, ms160, ms240,

ms320, ms640, ms1280, ms2560},

nprach-StartTime-r13 ENUMERATED {ms8, ms16, ms32, ms64,

ms128, ms256, ms512, ms1024},

nprach-SubcarrierOffset-r13 ENUMERATED {n0, n12, n24, n36, n2, n18, n34, spare1},

nprach-NumSubcarriers-r13 ENUMERATED {n12, n24, n36, n48},

nprach-SubcarrierMSG3-RangeStart-r13 ENUMERATED {zero, oneThird, twoThird, one},

maxNumPreambleAttemptCE-r13 ENUMERATED {n3, n4, n5, n6, n7, n8, n10, spare1},

numRepetitionsPerPreambleAttempt-r13 ENUMERATED {n1, n2, n4, n8, n16, n32, n64, n128},

npdcch-NumRepetitions-RA-r13 ENUMERATED {r1, r2, r4, r8, r16, r32, r64, r128,

r256, r512, r1024, r2048,

spare4, spare3, spare2, spare1},

npdcch-StartSF-CSS-RA-r13 ENUMERATED {v1dot5, v2, v4, v8, v16, v32, v48, v64},

npdcch-Offset-RA-r13 ENUMERATED {zero, oneEighth, oneFourth, threeEighth}

}

NPRACH-Parameters-NB-v1330 ::= SEQUENCE {

nprach-NumCBRA-StartSubcarriers-r13 ENUMERATED {n8, n10, n11, n12, n20, n22, n23, n24,

n32, n34, n35, n36, n40, n44, n46, n48}

}

NPRACH-ParametersList-NB-r14 ::= SEQUENCE (SIZE (1.. maxNPRACH-Resources-NB-r13)) OF

NPRACH-Parameters-NB-r14

NPRACH-Parameters-NB-r14 ::= SEQUENCE {

nprach-Parameters-r14 SEQUENCE {

nprach-Periodicity-r14 ENUMERATED {ms40, ms80, ms160, ms240,

ms320, ms640, ms1280, ms2560}

OPTIONAL, -- NEED OP

nprach-StartTime-r14 ENUMERATED {ms8, ms16, ms32, ms64,

ms128, ms256, ms512, ms1024}

OPTIONAL, -- NEED OP

nprach-SubcarrierOffset-r14 ENUMERATED {n0, n12, n24, n36, n2, n18, n34, spare1}

OPTIONAL, -- NEED OP

nprach-NumSubcarriers-r14 ENUMERATED {n12, n24, n36, n48}

OPTIONAL, -- NEED OP

nprach-SubcarrierMSG3-RangeStart-r14 ENUMERATED {zero, oneThird, twoThird, one}

OPTIONAL, -- NEED OP

npdcch-NumRepetitions-RA-r14 ENUMERATED {r1, r2, r4, r8, r16, r32, r64, r128,

r256, r512, r1024, r2048,

spare4, spare3, spare2, spare1}

OPTIONAL, -- NEED OP

npdcch-StartSF-CSS-RA-r14 ENUMERATED {v1dot5, v2, v4, v8, v16, v32, v48, v64}

OPTIONAL, -- NEED OP

npdcch-Offset-RA-r14 ENUMERATED {zero, oneEighth, oneFourth, threeEighth}

OPTIONAL, -- NEED OP

nprach-NumCBRA-StartSubcarriers-r14 ENUMERATED {n8, n10, n11, n12, n20, n22, n23, n24,

n32, n34, n35, n36, n40, n44, n46, n48}

OPTIONAL, -- NEED OP

npdcch-CarrierIndex-r14 INTEGER (1..maxNonAnchorCarriers-NB-r14)

OPTIONAL, -- Need OP

...

} OPTIONAL -- Need OR

}

NPRACH-ParametersListTDD-NB-r15 ::= SEQUENCE (SIZE (1.. maxNPRACH-Resources-NB-r13)) OF

NPRACH-ParametersTDD-NB-r15

NPRACH-ParametersTDD-NB-r15 ::= SEQUENCE {

nprach-Parameters-r15 SEQUENCE {

nprach-Periodicity-r15 ENUMERATED {ms80, ms160, ms320, ms640,

ms1280, ms2560, ms5120, ms10240}

OPTIONAL, -- NEED OP

nprach-StartTime-r15 ENUMERATED {ms10, ms20, ms40, ms80,

ms160, ms320, ms640, ms1280,

ms2560, ms5120, spare6, spare5,

spare4, spare3, spare2, spare1}

OPTIONAL, -- NEED OP

nprach-SubcarrierOffset-r15 ENUMERATED {n0, n12, n24, n36, n2, n18, n34, spare1}

OPTIONAL, -- NEED OP

nprach-NumSubcarriers-r15 ENUMERATED {n12, n24, n36, n48}

OPTIONAL, -- NEED OP

nprach-SubcarrierMSG3-RangeStart-r15 ENUMERATED {zero, oneThird, twoThird, one}

OPTIONAL, -- NEED OP

npdcch-NumRepetitions-RA-r15 ENUMERATED {r1, r2, r4, r8, r16, r32, r64, r128,

r256, r512, r1024, r2048,

spare4, spare3, spare2, spare1}

OPTIONAL, -- NEED OP

npdcch-StartSF-CSS-RA-r15 ENUMERATED {v4, v8, v16, v32, v48, v64, v96, v128}

OPTIONAL, -- NEED OP

npdcch-Offset-RA-r15 ENUMERATED {zero, oneEighth, oneFourth, threeEighth}

OPTIONAL, -- NEED OP

nprach-NumCBRA-StartSubcarriers-r15 ENUMERATED {n8, n10, n11, n12, n20, n22, n23, n24,

n32, n34, n35, n36, n40, n44, n46, n48}

OPTIONAL, -- NEED OP

...

} OPTIONAL -- Need OR

}

NPRACH-ParametersListTDD-NB-v1550 ::= SEQUENCE (SIZE (1.. maxNPRACH-Resources-NB-r13)) OF

NPRACH-ParametersTDD-NB-v1550

NPRACH-ParametersTDD-NB-v1550 ::= SEQUENCE {

maxNumPreambleAttemptCE-v1550 ENUMERATED {n3, n4, n5, n6, n7, n8, n10, spare1},

numRepetitionsPerPreambleAttempt-v1550 ENUMERATED {n1, n2, n4, n8, n16, n32, n64, n128,

n256, n512, n1024}

}

NPRACH-ParametersListFmt2-NB-r15 ::= SEQUENCE (SIZE (1.. maxNPRACH-Resources-NB-r13)) OF NPRACH-ParametersFmt2-NB-r15

NPRACH-ParametersFmt2-NB-r15 ::= SEQUENCE {

nprach-Parameters-r15 SEQUENCE {

nprach-Periodicity-r15 ENUMERATED {ms40, ms80, ms160, ms320,

ms640, ms1280, ms2560, ms5120}

OPTIONAL, -- NEED OP

nprach-StartTime-r15 ENUMERATED {ms8, ms16, ms32, ms64,

ms128, ms256, ms512, ms1024}

OPTIONAL, -- NEED OP

nprach-SubcarrierOffset-r15 ENUMERATED {n0, n36, n72, n108, n6, n54, n102, n42,

n78, n90, n12, n24, n48, n84, n60, n18}

OPTIONAL, -- NEED OP

nprach-NumSubcarriers-r15 ENUMERATED {n36, n72, n108, n144}

OPTIONAL, -- NEED OP

nprach-SubcarrierMSG3-RangeStart-r15 ENUMERATED {zero, oneThird, twoThird, one}

OPTIONAL, -- NEED OP

npdcch-NumRepetitions-RA-r15 ENUMERATED {r1, r2, r4, r8, r16, r32, r64, r128,

r256, r512, r1024, r2048,

spare4, spare3, spare2, spare1}

OPTIONAL, -- NEED OP

npdcch-StartSF-CSS-RA-r15 ENUMERATED {v1dot5, v2, v4, v8, v16, v32, v48, v64}

OPTIONAL, -- NEED OP

npdcch-Offset-RA-r15 ENUMERATED {zero, oneEighth, oneFourth, threeEighth}

OPTIONAL, -- NEED OP

nprach-NumCBRA-StartSubcarriers-r15 ENUMERATED {

n24, n30, n33, n36, n60, n66, n69, n72,

n96, n102, n105, n108, n120, n132, n138, n144}

OPTIONAL, -- NEED OP

npdcch-CarrierIndex-r15 INTEGER (1..maxNonAnchorCarriers-NB-r14)

OPTIONAL, -- Need OP

...

} OPTIONAL -- Need OR

}

NPRACH-TxDurationFmt01-NB-r17 ::= SEQUENCE {

nprach-TxDurationFmt01-r17 ENUMERATED {n2, n4, n8, n16, n32, n64}

}

NPRACH-TxDurationFmt2-NB-r17 ::= SEQUENCE {

nprach-TxDurationFmt2-r17 ENUMERATED {n1, n2, n4, n8, n16}

}

RSRP-ThresholdsNPRACH-InfoList-NB-r13 ::= SEQUENCE (SIZE(1..2)) OF RSRP-Range

EDT-TBS-InfoList-NB-r15 ::= SEQUENCE (SIZE (1.. maxNPRACH-Resources-NB-r13)) OF EDT-TBS-NB-r15

EDT-TBS-NB-r15 ::= SEQUENCE {

edt-SmallTBS-Enabled-r15 BOOLEAN,

edt-TBS-r15 ENUMERATED {b328, b408, b504, b584, b680, b808, b936, b1000}

}

-- ASN1STOP

Here is the TP for NB-IoT CB-Msg3 configuration parameters.

#### – *CB-Msg3-ConfigSIB-NB*

The IE *CB-Msg3-ConfigSIB-NB* is used to specify CB-Msg3 configuration.

*CB-Msg3-ConfigSIB-NB* information element

-- ASN1START

CB-Msg3-ConfigSIB-NB-r19 ::= SEQUENCE {

cb-Msg3-MinRSRP-Threshold-r19 NRSRP-Range-NB-r14 OPTIONAL, --Need OR

cb-Msg3-RSRP-ThresholdList-r19 CB-Msg3-RSRP-ThresholdList-NB-r19 OPTIONAL, --Need OP

cb-Msg3-ParametersList-r19 CB-Msg3-ParametersList-NB-r19,

cb-Msg3-ResponseWindowTimer-r19 ENUMERATED {FFS}

}

CB-Msg3-ParametersList-NB-r19 ::= SEQUENCE (SIZE (1.. maxCB-Msg3-Resources-NB-r19)) OF

CB-Msg3-Parameters-NB-r19

CB-Msg3-Parameters-NB-r19 ::= SEQUENCE {

cb-Msg3-NumReplicas INTEGER(1..4),

cb-Msg3-DSATransmissionWindow-r19 SEQUENCE {

startSFN-19 INTEGER (0.. 1023),

windowSize-19 ENUMERATED {FFS},

windowPeriodicity-r19 ENUMERATED {FFS}

} OPTIONAL, --Need OP

cb-Msg3-StartTimeParameters-r19 SEQUENCE {

npusch-periodicity-r19 ENUMERATED {FFS},

npusch-startTime-r19 INTEGER (0..1023),

npusch-startSubframe-r19 INTEGER (0..9)

}

cb-Msg3-PhysicalConfig-r19 ::= SEQUENCE {

npusch-NumRUsIndex-r19 INTEGER (0..7),

npusch-NumRepetitionsIndex-r19 INTEGER (0..7),

npusch-SubCarrierIndex-r19 INTEGER (0..47),

npusch-MCS-r19 INTEGER (0..10),

p0-UE-NPUSCH-r19 INTEGER (-8..7),

alpha-r19 ENUMERATED {al0, al04, al05, al06,

al07, al08, al09, al1},

npdcch-NumRepetitions-r19 ENUMERATED {r1, r2, r4, r8, r16, r32, r64, r128,

r256, r512, r1024, r2048,

spare4, spare3, spare2, spare1},

npdcch-StartSF-CSS-r19 ENUMERATED {v1dot5, v2, v4, v8, v16, v32, v48, v64},

npdcch-Offset-r19 ENUMERATED {zero, oneEighth, oneFourth, threeEighth}

npdcch-CarrierIndex-r19 INTEGER (1..maxNonAnchorCarriers-NB-r14)

OPTIONAL -- Need OP

}

cb-Msg3-TBS-NB-r19 ENUMERATED {FFS},

...

}

CB-Msg3-RSRP-ThresholdList-NB-r19 ::= SEQUENCE (SIZE(1..2)) OF RSRP-Range

maxCB-Msg3-Resources-NB-r19 INTEGER ::= 3 -- Maximum number of CB-Msg3 resources for NB-IoT

-- ASN1STOP

Notes

* CB-Msg3-ConfigSIB-NB:This IE is used in the **SIB2-NB** for anchor carrier and **SIB22-NB** for non-anchor carrier.
* cb-Msg3-MinRSRP-Threshold-r19: It is assumed as an optional IE. If it is absent, no minimal threshold is used.
* CB-Msg3-RSRP-ThresholdList-NB-r19: It is assumed as an optional IE. If this IE is absent, the thresholds list of PRACH is used.
* cb-Msg3-DSATransmissionWindow-r19: It is assumed as an optional IE. If the number of the replicas is one, the DSA transmission window is not needed.
  + startSFN-19 in cb-Msg3-DSATransmissionWindow-r19: Although the term *H-SFN offset* is used in the agreement, the rapporteur believes it should be a SFN offset during the H-SFN duration.
  + *hsfn-LSB-Info-r16* in PUR-Config-NB: This IE is not used as the periodicity of CB-Msg3 resource is assumed shorted than a H-SFN duration (i.e., 10.24s).
* cb-Msg3-PhysicalConfig-r19
  + npusch-SubCarrierIndex-r19 / npusch-MCS-r19: According to the RAN1 reply LS [R1-2407548](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_118/Docs/R1-2407548.zip) on TA validation, only 3.75kHz SCS is confirmed. **Whether the 15kHz SCS NPUSCH is supported is still FFS**. Therefore, only parameters for 3.75kHz SCS is introduced for now.
  + p0-UE-NPUSCH-r19, alpha-r19: The parameters for NPUSCH transmission power are assumed as CE level specific.
  + *npusch-CyclicShift-r16* in PUR-Config-NB: This IE is not used because this parameter is specified for PUR only.
  + npdcch-NumRepetitions-r19/npdcch-StartSF-CSS-r19/npdcch-Offset-r19: 3 parameters copied from *NPDCCH-ConfigDedicated-NB-r13.* Note hear it should be CSS instead of USS.
  + npdcch-CarrierIndex-r19: The non-anchor carrier index for monitoring Msg4. If this IE is absent, anchor carrier is assumed to be used.
  + *ack-NACK-NumRepetitions-r16* in PUR-Config-NB: This IE is not used because the same IE in Msg4 can be used.
* cb-Msg3-TBS-NB-r19: Whether the maximum TBS is CE level specific is FFS. Rapporteur assumes it is CE level specific.

Companies are invited to provide comments on the above TP including but not limited to below discussion points.

* Which SIB should be used for IE CB-Msg3*-ConfigSIB* ?
* Any parameter is missing in the TP and why this should be added?
* Any parameter should be removed from the TP?
* Most value range of the parameters are copied from PUR parameters, any further change needed?

**Q2: Any comments on the TP of *CB-Msg3-ConfigSIB-NB?***

|  |  |
| --- | --- |
| **Company** | **Comments** |
| Nokia | Same comments as Q1 (except the one for NRSRP-Range).  Furthermore, for NB-IoT, we think it is too early to conclude “the periodicity of CB-Msg3 resource is assumed shorted than a H-SFN duration (i.e., 10.24s).” E.g., when considering the largest number of repetitions, number of resources units (RU), and number of slots in RU, the maximum transmission time for one replica could be 10.24 s for 15kHz SCS and 40.96 s for 3.75kHz SCS. |
| NEC | Most comments align with the eMTC portion.   1. Typographical Errors: In the *IE cb-Msg3-DSATransmissionWindow-r19*, the parameters *startSFN-19* and *windowSize-19* should be revised to ***startSFN-r19***and ***windowSize-r19***, respectively 2. *cb-Msg3-DSATransmissionWindow-r19*: We agree that *cb-Msg3-DSATransmissionWindow-r19* is optional.    1. When CB-Msg3-DSA is enabled and configured by the network, the network **MUST** set the window’s starting point **in accordance with RAN2#129 agreements**. For NB-IoT, due to the 3.75 kHz subcarrier spacing (SCS), RU duration (16 slots, 32 ms), the number of RUs per NPUSCH transmission, and required repetitions, the CB-Msg3 resource or DSA window **MAY exceed 10.24 s**. Consequently, it is **FFS** whether both ***startHSFN-r19*** and ***startSFN-r19*** should define the DSA window start.    2. Since a UE can derive *windowSize-r19* and *windowPeriodicity-r19* from the configured *cb-Msg3-NumReplicas* and *cb-Msg3-StartTimeParameters-r19*, these parameters **MAY** be optional. The final determination of *windowSize-r19* and *windowPeriodicity-r19* remains **FFS for RAN2**. 3. Unlike PUR, NB-IoT UEs **MAY** transmit CB-Msg3 on different subcarriers. NPRACH defines subcarrier resources via *nprach-SubcarrierOffset-r15* (starting point) and *nprach-NumSubcarriers-r15* (frequency range). Whether the network should configure CB-Msg3 on **contiguous or non-contiguous** subcarriers has not been discussed in RAN2 and is **FFS**. If contiguous CB-Msg3 resources are desired, NPRACH configurations could serve as a baseline. For example:   npusch-SubcarrierOffset-r19 ENUMERATED {FFS},  npusch-NumSubcarriers-r19 ENUMERATED {FFS}, |
| ZTE | Similar comments as Q1.  For example, *npusch-SubCarrierIndex-r19* should be with a “set” format instead of configuring only one specific value. An example is as below:  cb-Msg3-PhysicalConfig-r19 ::= SEQUENCE {  npusch-NumRUsIndex-r19 INTEGER (0..7),  npusch-NumRepetitionsIndex-r19 INTEGER (0..7),    npusch-SubCarrierSet-r19 SEQUENCE (SIZE (1.. FFS)) OF INTEGER (0..47),  ....... |
| Qualcomm | Same as EDT, this can be in SIB2.  For now we should confirm the parameters we need. Whether anchor and non-anchor configurations are supported can be checked with RAN1. We should also check with RAN1 on any missing parameter of any parameters that are not needed. |
| CATT | Basically similar comments as to Q1. |
| Huawei, HiSilicon | Similar comments as Q1. |
| vivo | Same as comments to Q1. |
| Apple | Same as comments to Q1. |
|  |  |
|  |  |
|  |  |
|  |  |

## Msg4 monitoring window and RNTI

Regarding the Msg4 monitoring window and RNTI, the following agreements have been reached:

RAN2#127bis agreement:

1. The RNTI used at least to schedule Msg4 transmission is derived based on the resource associated to the PUSCH occasion used for contention based Msg3 EDT transmission (FFS on the details. FFS how this is impacted by DSA)

RAN2#128 agreement:

1. For SA case (single replica), after the end of all repetition of CB-Msg3 PUSCH transmission, UE starts a window for response reception taking UE-eNB RTT into account. FFS if we need to consider additional delay e.g. for the processing time

2. For DSA case, FFS if we only have one or multiple PDCCH monitoring window(s) (i.e. one window per each replica) for response reception. FFS when the window(s) is/are started (or restarted) and stopped. FFS on the window length. FFS if the UE needs to monitor only one RNTI or multiple RNTIs)

During the online discussion on CB-Msg3 in RAN2#129, it was found that the DSA transmission window, the Msg4 monitoring window, and the RNTI for Msg4 are closely related. An offline discussion was initiated to narrow down the combinations of these three aspects.

* [AT129][306][R19 IoT NTN] TX and RX window for CB-msg3 (Mediatek)

Scope: discuss details of transmission window (e.g. sliding or fixed), monitoring window and whether 1 or multiple RNTIs should be considered for CB-msg3/DSA

      Intended outcome: summary of the offline discussion

      Deadline for companies' feedback:  Thursday 2025-02-20 20:00

      Deadline for rapporteur's summary (in R2-2501420):  Friday 2025-02-21 08:00

As a result of the offline discussion [1], two proposals were submitted as follows:

**Proposal 1: For CB-Msg3 DSA transmission window design, RAN2 to discuss below two options:**

* **Option 1** 
  + **Transmission window is started at the first replica that the UE randomly select from CB-MSG3 occasions. Window length is configured by network via SIB.**
* **Option 2** 
  + **Transmission window is configured by network with a starting point (e.g. H-SFN offset), a window length, and a window periodicity.**
  + **The UE first selects a nearest DSA transmission window and then randomly select K replicas inside the window.**

**Proposal 2: For CB-Msg3 DSA monitor window and RNTI design, RAN2 to discuss below two options:**

* **Option 1** 
  + **RNTI is calculated for each Replica respectively according to the selected resource (same as SA).**
  + **After the end of all repetition of CB-Msg3 PUSCH transmission of each replica, UE starts the corresponding monitor window, taking UE-eNB RTT into account.**
  + **The UE has to monitor multiple RNTIs in multiple monitor windows.**
* **Option 2**
  + **A single RNTI is used for all replicas and it is calculated based on the location of the selected DSA transmission window.**
  + **A single monitor window is used for all replicas.**

Regarding proposal 1 (i.e., the DSA transmission window), the following working assumption have been reached.

RAN2#129 Working assumption:

1. For CB-MSG3, the Transmission window is configured by the network with a starting point (e.g. H-SFN offset), a window length, and a window periodicity (window length and periodicity could be the same). For k=1 the window length can be equal to 1: same behaviour as today

The UE first selects the next DSA transmission window and then randomly select K replicas inside the window.

P2 was briefly discussed, but no conclusion was reached. We will continue our discussion on the Msg4 monitoring window and RNTI in this offline.

Although option 2 is adopted for the DSA transmission window, companies have varied views on the Msg4 monitoring window and RNTI. Some companies mentioned that even with a fixed DSA transmission window, the Msg4 monitoring window can still be multiple. The RNTI design would also be impacted accordingly. Therefore, the rapporteur would like to decouple options on the monitoring window and RNTI.

Companies are invited to provide their views on the following questions:

**Q3: Should there be one Msg4 monitoring window or multiple windows for DSA transmission?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Multiple or Single** | **Comments** |
| Nokia | Single for Half-Duplex UEs.  Multiple for Full-Duplex eMTC UE. | In last RAN2 meeting, it was agreed that RAN2 assumes that a pointer solution is not needed in Rel-19. This means the NW is not aware of the Msg3 transmission occasions within the Msg3 transmission window.   * For half-duplex UE, there may have Msg3 and Msg4 collision if the NW schedules the Msg4 within the Msg3 transmission window. Therefore, a single Msg4 monitoring window could be started at least after the end of Msg3 transmission window hence the NW schedules Msg4 within the Msg4 monitoring window to avoid the UL/DL collision. The drawback of single Msg4 monitoring window is that, the NW has no means to respond Msg4 to UE as early as possible even if NW has decoded a Msg3 successfully. This is not only a waste of UE’s power consumption but also a waste of NW resource (which will increase the Msg3 collision rate as well). * For full-duplex UE, there is no Msg3 and Msg4 window collision issue at all. Therefore, once the eNB successfully decodes one of the multiple replicas, it may respond as early as possible (i.e., without waiting for the reception of the remaining replica(s)). The UE could stop transmitting the remaining replicas if it has received a CB-msg4 containing a matching Contention Resolution Identity. The early termination of Msg3 replicas transmission would benefit both UE and NW. In this case, a Msg4 reception window should be started after the first Msg3 replica (plus RTT) to receive the possible Msg4 response as early as possible.   On how to (re)start the Msg4 reception window, there may have two options:   * In one way (Option#1), the UE may apply the window with quite long duration to cover the whole period to receive possible Msg4 reception responding to the first or subsequent Msg3 replicas. However, this may cause UE and NW misalignment because NW may miss the first Msg3 replica reception hence UE and NW have different understanding on when the window is started and the exact window length. To avoid the NW and UE misalignment, the Msg4 reception window should be restarted after each Msg3 replica. * In the other way (Option#2), the UE may start an individual Msg4 reception window for each of the replica. (i.e., multiple Msg4 reception windows). From specification implementation point of view, Option#2 seems easy for window modelling, and it anyway needs to be specified for SA with single replica.   Therefore, we prefer the Option#2 for Full Duplex UE.  Note: PUSCH repetition early termination is a legacy feature which has been supported in legacy eMTC Full Duplex system (see *mpdcch-UL-HARQ-ACK-FeedbackConfig*). |
| NEC | Single for a unified design | Agree with Nokia.  If NB-IoT and eMTC adopt a unified design framework, a single window configuration is the only feasible solution. |
| ZTE | Multiple | We prefer to use multiple Msg4 monitoring windows.  Firstly, we are unclear if single Msg4 monitoring window is used, how long it will be? A configured window length or multiple (e.g., number of replicas) times of a configured window length?   * If it is the former, we think it infeasible, as it imposes much restrictions on network scheduling of Msg4, potentially preventing the transmission of Msg4 corresponding to the latter replicas. * If it is the latter, we think that single window scheme and multiple windows scheme have little difference for the impacts on UE power consumption (the complexity of monitoring RATI is discussed separately, e.g., in Q5).   Secondly, regarding the start of Msg4 monitoring window:   * Even for the start of a single Msg4 monitoring window, we think it is unnecessary to impose restriction that it needs to start at least after the end of Msg3 transmission window. In some configurations, such as a large RTT and a short Msg3 transmission window, this scheme also has its drawbacks, e.g., the window may start too early and the first Msg4 may arrive long after the start of Msg4 monitoring window. Shortly to say, the assumed start of single Msg4 monitoring window is not always appropriate in half-duplex scenario either. * For full-duplex scenarios, we agree with Nokia's analysis of the benefits. * Hence, we suggest not defining different Msg4 monitoring window start mechanisms for different scenarios, as this would be too complicated. A unified mechanism can be used instead. One way to avoid the issue of start of multiple Msg4 monitoring windows, e.g., UE monitors Msg4 too early before finishing all the replicas transmission, is to set an appropriate Msg3 transmission window length, e.g., now too short or too long and can be corresponding to the RTT. |
| Qualcomm | Single | Single window keeps the feature simple. We also think unified solution for eMTC and NB-IoT is better. |
| Samsung | Single | We do not think that it imposes restrictions on the scheduling of Msg4, actually it increases flexibility as multiple Msg4 can be sent in one message. With multiple windows, this cannot be done reliably.  With a single Msg4 window, PDCCH resources are also saved.  For single window, all UEs transmitting in one DSA window monitor the same window, so there is no notion of “preventing the transmission of Msg4 corresponding to the latter replicas”. We cannot see a case where there is an excessive number of UEs that need to be scheduled in a Msg4 window. In that case, there likely needs to be more DSA windows configured in order to offload UEs to more DSA windows, similar to configuring suitable PRACH resources according to the load in legacy random access procedures. This was for instance never an issue discussed for 2-step random access.  For any benefits applicable to full duplex, full duplex is not a priority for IoT, especially if we look at what is deployed in the market. Note that while there is PUSCH early termination, there is no early termination for Msg1. We think that this case is more similar to Msg1 transmissions. |
| Ericsson | Single | For simplicity. Agree with Samsung. |
| CMCC | Support multiple windows for DSA transmission | For one window, network configure only one long duration to cover all replicas, which may be lead to power consumption for UE. Then for multiple window (i.e. one window per each replica), UE (re)starts multiple shorter window for corresponding replica. And considering we have agreed that the number of replicas for DSA will be configured by the NW: 1 (SA), 2, 3, 4. The replica number is not so large, that is to say the monitoring complexity on the UE side is not so high. Therefore multiple PDCCH monitoring windows may be more suitable. |
| CATT | Multiple Msg4 monitoring windows | We understand that companies proposing to operate only one monitoring window for the half-duplex case are with the assumption that the Msg.4 monitoring window is only started after the transmission of all selected DSA occasions are finished. However, this is not in line with the following agreements in RAN2#128:   * “The UE stops the PDCCH monitoring window(s) once it receives a CB-msg4 containing a matching Contention Resolution Identity (FFS if there is no RRC message together with the CB-msg4) * Assuming that there will be scenarios where it’s possible to receive a CB-msg4 before the UE transmits some replicas, a UE stops transmitting the remaining replicas if it has received a CB-msg4 containing a matching Contention Resolution Identity (FFS if there is no RRC message together with the CB-msg4) * RAN2 understands that, for DSA, once the eNB successfully decodes one of the multiple replicas, it may respond without waiting for the remaining replica(s) (FFS when the response window(s) is/are started)   which still allow the chance for the UE to monitor the Msg.4 response of a replica before a future replica is transmitted. If companies really want to support this single monitoring window for the half-duplex case, we need to revisit the above agreement and consider introducing potential resource selection restriction w.r.t. when the monitoring window is started. |
| Huawei, HiSilicon | Multiple | 1. Independent multiple windows are much simpler following the same logic as what is defined for SA. While for single window, extra effort is needed to define the start point and the duration of the window for DSA.  2. With multiple windows, UE is able to stop monitoring RATI for power saving if there are gaps between two windows. While for single window, UE has to monitoring the RNTI for the whole time until the ending of the window.  3. With single window, UE/NW has to wait until all the replicas have been sent/received to receive/send the feedback, which will bring uncessary latency in case the first replica is already successful.  4. We don’t think the half-duplex is a big issue since the selection of replica occasions are up to UE and anyway UE can prioritize the reception of feedback by implementation. |
| vivo | Single | Unified solution is expected. |
| ETRI | Single | For simplicity and unified design. |
| Xiaomi | Single | For simplicity. |
| Apple | Multiple | We actually feel multiple window is simpler and power efficient. For example, if the single window length has to be enlarged somehow for NW to accommodate all Msg4(s), UE would waste more power in receiving Msg4. In addition, with single window, the starting point of the window would be delayed. |

**Q4: What should be the start point of Msg4 monitoring window(s) for DSA transmission?**

|  |  |
| --- | --- |
| **Company** | **Comments** |
| Nokia | * For half-duplex, since the NW is not aware the Msg3 replica occasions, to avoid UL/DL collision, the single Msg4 monitoring window starts at the end of the Msg3 transmission window plus UE-eNB RTT. * For full-duplex, the UE starts individual Msg4 reception window for **each** of the replica in the subframe containing the last PUSCH repetition of the replica plus UE-eNB RTT. |
| NEC | If NB-IoT and eMTC adopt a unified design framework, the Msg4 monitoring window MUST be configured as a single window starting after the corresponding DSA transmission window, accounting for UE-eNB RTT and processing time. |
| ZTE | Similar as that in SA, each Msg4 monitoring window can be started at the end of transmission of one replica plus UE-eNB RTT plus 4 subframes (processing time). |
| Qualcomm | The start point can be end of first replica plus UE-eNB RTT but then we should clarify that the UL transmission is prioritized over PDCCH monitoring.  For this reason, we think it is simple to define the start point of the window after the end of last replica. |
| Samsung | Agree with NEC. With a single window, then it must start after the end of the DSA window. |
| Ericsson | Agree with NEC and Samsung. |
| CMCC | UE starts corresponding Msg4 monitoring window for each replica at the end of the last repetition of the replica plus UE-eNB RTT. |
| CATT | We think multiple Msg4 monitoring windows should be used for DSA transmissions. Particularly, Msg4 monitoring window should start after each DSA transmission occasion, considering the RTT and necessary processing time. |
| Huawei, HiSilicon | Similar as SA: Msg4 monitoring window starts after the sending of the corresponding MSG3 plus UE-eNB RTT. No need to consider the processing time which can be covered by the window duration. |
| vivo | Same view as Sumsung. |
| ETRI | End of DSA transmission window + RTT between UE and eNB. |
| Xiaomi | End of transmission of first replica plus UE-eNB RTT |
| Apple | End of transmission of each replica plus UE-eNB RTT. |

**Q5: Should there be one RNTI or multiple RNTIs for DSA transmission?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Multiple or Single** | **Comments** |
| Nokia | Single | Due to up to 4 RNTIs can be derived in DSA for different replica occasions, it may burden UE to monitor 4 RNTIs for Msg4 reception at the same time. |
| NEC | Prefer multiple RNTIs | While both approaches are feasible, a single RNTI would require resolving all contention within the Msg4 PDSCH. In contrast, multiple RNTIs enable grouping contention resolution into subgroups via the Msg4 PDCCH. Given the higher decoding complexity, elevated false alarm rate, and potential missed detection rate on the NPDSCH, multiple RNTIs are advantageous for low-cost, low-power NB-IoT UEs. |
| ZTE | Multiple | Till now we see only two options for defining single RNTI:   * Alt1: A single RNTI is used for all replicas and it is calculated based on the location of the selected DSA transmission window.   + Based on the agreement on the fixed Msg3 transmission window, it’s easy to see that Alt1 will cause high conflicts for the UEs that trigger Msg3 transmission in a same resource periodicity, e.g., same Msg3 transmission window. The UEs within this window will monitor same RNTI. The more UEs performing DSA within this window, the more Msg4s each UE will demodulate that do not target to it, resulting in more unnecessary power consumption. * Alt2: A predefined/allocated common RNTI.   + The drawback is that, the UE has to decode every MAC PDU within the Msg4 monitoring window until they identity a matching UE Contention Resolution Identity. This exhaustive decoding process will also significantly increase UE power consumption.   One the other hand, in the scheme of multiple RNTIs, the RNTI is derived based on the transmission resources for the corresponding replica. We see it simple for the processing logic in the UE and can address conflicts as much as possible. |
| Qualcomm | Single | We believe number RNTIs to monitor = number replicas configured in a DSA group can also work.  But we prefer single RNTI, and UE does not have to handle multiple RNTIs.   * For UEs with successful response (i.e., response with its contention resolution ID), should either go to IDLE (not monitor anything) or receive new C-RNTI for monitoring further DL messages. * Completing procedure and moving all UEs can be done with single RNTI. * For the UE specific response, the contention resolution can be included. * Retransmission is also possible by including resource index in the retransmission command. |
| Samsung | Single | Single RNTI is needed in order to address multiple UEs with one Msg4, which is one of the main benefits of the CB-EDT procedures.  With single window which is the same for all UEs using a particular DSA window, then there is no need for multiple RNTIs.  Resolving the contention using a single RNTI is also not much of an issue as the contention resolution mechanism is already very strong. |
| Ericsson | Single | Agree with Samsung. About ZTE Alt2: most of the time very few UEs will transmit in a DSA window, thus likely only one msg4 is needed – thus most of the time there is no extra monitoring/decoding. |
| CMCC | Multiple | As we have agreed to derive it based on the resource associated to the PUSCH occasion used for contention based Msg3 EDT transmission. And it may be difficult to design the common RNTI associated with the PUSCH transmission resources of multiple replicas. Then multiple RNTI could be calculated based on corresponding replica transmission resource respectively, which is more reasonable and clear. |
| CATT | Multiple | Seems UE vendors in previous meetings did not see multiple RNTI handling as an unacceptable burden. |
| Huawei, HiSilicon | Single | 1. With single RNTI, UE can avoid monitoring multiple RNTIs at the same time, which adds complexity to IoT UEs. 2. With single RNTI, multiple MSG4 can be multiplexed into one MAC PDU, which saves PDCCH signaling. |
| vivo | Single | For multiple UEs multiplexing in Msg4, we think single RNTI shoul be used. |
| ETRI | Single | UE complexity reduction due to multiple RNTI monitoring and signalling reduction due to multiplexing of msg4. |
| Xiaomi | Multiple | Each RNTI is related to each replica transmission. |
| Apple | Multiple | If using one single RNTI and if NW needs to transmit multiple Msg4(s), the false alarm rate at UE reception would be very high. It is also understandable that if the RNTI(s) space is too large, the blind decoding of PDCCH would become heavy. Thus, a middle ground could be grouping multiple resources to a single RNTI. It might be too demanding to associate all resources in a Msg3 window to one single RNTI. |

**Q6: How is the RNTI derived for DSA transmission?**

|  |  |
| --- | --- |
| **Company** | **Comments** |
| Nokia | For the RNTI used for Msg4 monitor, the most straight-forward method is that the UE includes the RNTI information in the Msg3 content. We acknowlege the 16 bits C-RNTI may overhead Msg3 a lot. The alternative is to derive the C-RNTI based on the NW-configured/fixed Msg3 transmission window. We think RAN2 can further discuss the details on how to reduce the C-RNTI overhead or how to derive the C-RNTI from the window. |
| NEC | * **If multiple RNTIs are adopted**, the RNTI SHOULD be derived from the **time-domain resource** of the corresponding DSA transmission replica. * **If a single RNTI is adopted**, two options are proposed:   1. A **common RNTI** for all CB-Msg3 transmissions, or   2. A **window-specific RNTI** based on the selected DSA window. |
| ZTE | We prefer multiple RNTIs scheme. In this scheme, the RNTI is derived based on the transmission resources for the corresponding replica. |
| Qualcomm | This can be simple as:  CB-RNTI=1 + SFN\_id + ceil(1024/w)\*carrier\_id  Where w is the CB Msg3 transmission window size in terms of the number of SFNs. And SFN\_id is the SFN value where the CB Msg3 transmission window starts.  This means, if w = 4 SFN and non-anchor carrier with carrier\_id = 1, then any msg3 transmission window starting at SFN = 0 will result in the same RNTI value, i.e., 257. |
| Samsung | It should be based on the DSA window. I think the approach outlined by QC makes sense and can be taken as a baseline, because the DSA windows should not overlap. |
| Ericsson | Use a single RNTI. Simplest is to configure one CB EDT RNTI in SIB, then no “unused” RNTIs are reserved. |
| CMCC | Derive the RNTI based on corresponding replica transmission resource respectively. Even though the pointer solution is not supported in R19, NW is able to infer the transmission resource used by the replica after receiving the replica. |
| CATT | Derived based on the time and frequency resource for a replica transmission. Note that OCC during CB-msg.3 seems not able to be supported in this release, so OCC index does not need to be considered. |
| Huawei, HiSilicon | Since the MSG3 transmission window is fixed, the simplest way is to derive the RNTI based on a designated resource within the MSG3 transmission window or just based on NW’s explicated configuration. |
| vivo | RNTI can be calculated based on the DSA window time/frequency resource and the length of Cb-Msg4 response window length. E.g., different DSA window may mapped to different RNTIs. |
| ETRI | For multiple RNTI, time/frequency domain resources for replica transmission can be utilized.  For a single RNTI, predefined or DSA window based RNTI can be employed. |
| Xiaomi | Each RNTI is related to each replica transmission |
| Apple | Each RNTI is derived based on the resource selected for each replica. Some grouping mechanism can be considered to associate multiple resources with one single RNTI. |

# Summary

# References

[1] R2-2501420 Report of [AT129][306][R19 IoT NTN] TX and RX window for CB-msg3 Mediatek