3GPP TSG-RAN2 Meeting #116e-bis R2-210xxxx

e-Meeting, 16th-27th August 2021

Source: Email discussion Rapporteur (Ericsson)

Title: [Post116-e][514][RACH partitioning] Signaling design (Ericsson)

Agenda item: x.y.x

Document for: Discussion and Decision

# Introduction

This document contains summary of **detailed** input to the email discussion as detailed below:

* [Post116-e][514][RACH partitioning] Signaling design (Ericsson)
* Discussion points on details/principles yet to be defined
* A running CR (based on current status) that also incorporates the above when some kind of direction can be made; maybe with a few, preferable very few, alternatives.
* Deadline: Long

Deadline for company comments and responses to technical isses: 15th December, 12:00 am UTC.

The Rapporteur will attempt to summarize and make updates to the running CR based on progress and consensus.

# General

In a draft running CR for RA partitioning a first draft proposed signalling structure have been implemented.

In the below table, please fill in applicable **detailed** comments, if any, on the draft running CR for RA partitioning.

Note that current agreements made for RA partitioning can be found as an Annex in the draft running CR.

🡪 *Minor comments related to the running CR can be made directly in the CR itself*. However, for discussion points where companies would to present more complex comments or suggestions, the below table should be used.

Suggestion: In the below companies should aim for a technical discussion, not only single view input in order to progress outstanding design principals/details.

|  |
| --- |
|   |
| Company | Section/Item or IE, etc | Detailed comment with motivation | Example code, procedure |
| Intel#1 | RACH-ConfigCommon | We noticed tha feaureCombination-r17 is in 2 places: One in the RACH-ConfigCommon-r17 and another in the featureCombinationPreambles-r17. We think that only the one in the featureCombinationPreambles-r17 is needed when the RACH resource configured is for both RO sharing case and non-RO sharing cases.For the non-RO sharing case, the featureCombinationPreambles-r17 just provides the featureCombination-r17 | Remove the IE featureCombination-r17 in the RACH-ConfigCommon-r17 |
| Intel#2 | FeatureCombinationPreambles-r17 | With the change in Intel#1, the featureCombination-r17 can be made mandatory in FeatureCombinationPreambles-r17. Currently, it is unclear the expected behaviour if this field is absent with the configuration of featureCombinationPreambles-r17 | Remove OPTIONAL for featureCombination-r17 |
| Intel#3 | featureCombinationPreambles-r17 added to RACH-ConfigCommon and RACH-ConfigCommonTwoStepRA | Our understanding is that this new addition can also be applied to legacy 2-step and 4-step RACH configuration. Is this the intended/correct understanding?If so, the preamble partitioning for the other feature combination has to come after the end of the preamble partitioning of the legacy 2-step RACH. If this still needs further discussion, please add an editor note. |  |
| Intel#4 | featureCombinationPreambles-r17 added to RACH-ConfigCommon and RACH-ConfigCommonTwoStepRA | For the additional RACH configuration, we assume that the preamble partitioning will start with the 4-step RACH and will be partitioned with the group A + B of the first feature combination to the group A + B of the last feature combination configured for the 4-step RACH configuration. The preamble partitioning for 2-step RACH will then start from the end of the of the last feature combination configured for the 4-step RACH configuration and the preamble partitioning for the feature combination configured for 2-step RACH will follow as like the 4-step case (i.e. partitioned with the group A + B of the first feature combination to the group A + B of the last feature combination configured for the 2-step RACH configuration).If this still needs further discussion, please add an editor note. | Summarizing the RACH partitioning in the case of shared RO among feature/feature combinations of e.g. F1, F2, F1+F2 with 2-step and 4-step RACH also sharing the RO will be like the following: |
| ZTE#1 | FeatureCombination  | The detailed meaning of each feature bit should be captured somewhere. For example, if the slicing info is not included, then the feature combination is available to all slices. We are fine to capture it in either MAC or RRC. If we capture this in MAC, then a reference to MAC should be added here. For example, …is one of the features of this feature combination as specified in TS 38.321 [3] | ***If we want capture the meaning of each feature bit in RRC, then the following principles can be considered (e.g. the detailed description in the RRC field description):***1. if REDCAP indication is configured for the partition, then the RACH partition is only applicable to the RACH procedure triggered for REDCAP UE where Msg1 identification is required. Otherwise, if REDCAP indication is not configured, then the RACH partition is applicable to non-REDCAP UE and REDCAP UE where Msg1 identification is not required. (FFS how to determine whether Msg1 identification is required or not)
2. if slice info is configured for the partition,then the RACH partition is only applicable to the RACH procedure triggered for the slice. Otherwise, if the slice info is not configured, then the RACH partition is applicable to all slices.
3. if SDT indication is configured, then the RACH partition is only applicable to the RACH procedure triggered for SDT. Otherwise, if SDT indication is not configured, then the RACH partition is applicable to the RACH procedure not triggered for SDT.
4. if CE indication is configured, then the RACH partition is only applicable to the RACH procedure where CE is required. Otherwise, if CE indication is not configured, then the RACH partition is applicable to the RACH procedure where CE is not required. (if CE is considered as part of feature combination)

***If we want to capture it in MAC, then a reference can be added here. For example:******redCap***If present, this field indicates that RedCap is one of the features of this feature combination as specified in TS 38.321 [3].  |
| ZTE#2 | Generic for RO sharingRACH-ConfigCommon-r17FeatureCombinationPreambles-r17 | For RO sharing, we think the following three cases shall be considered:* Case 1: RA resource in R17 RA partition shares the RO with legacy RA resource.
* Case 2: Different types of RA resource within one RA partition share the RO with each other
* Case 3: RA resource in one RA partition share the RO with RA resource from another RA partition

In general, we think all the three cases above shall be supported, and a common structure is preferred.For the detail structure, since RACH-ConfigGeneric is mandatory present in RACH-ConfigCommon. we prefer to introduce a new structure RACH-ConfigCommon-r17 instead of the RACH-ConfigCommon.With the common structure proposed, the FeatureCombinationPreambles is not needed. | RACHPartition-ConfigCommon-r17 ::= SEQUENCE { rachPartition-ConfigID-r17 INTEGER(1..maxRACHAdditionalRACH-r17) rach-ConfigCommon-r17 RACH-ConfigCommon-r17 OPTIONAL, -- Need M msgA-ConfigCommon-r16 MsgA-ConfigCommon-r16 OPTIONAL, -- Cond SpCellOnly2 featureCombination-r17 FeatureCombination-r17 OPTIONAL}RACH-ConfigCommon-r17 ::= SEQUENCE { occasions CHOICE { shared-RO-r17 Shared-RO-r17, separate-RO-r17 Separate-RO-r17 }  groupBconfigured SEQUENCE { ra-Msg3SizeGroupA ENUMERATED {b56, b144, b208, b256, b282, b480, b640, b800, b1000, b72, spare6, spare5,spare4, spare3, spare2, spare1}, messagePowerOffsetGroupB ENUMERATED { minusinfinity, dB0, dB5, dB8, dB10, dB12, dB15, dB18}, numberOfRA-PreamblesGroupA INTEGER (1..64)} OPTIONAL, -- Need R ra-ContentionResolutionTimer ENUMERATED { sf8, sf16, sf24, sf32, sf40, sf48, sf56, sf64}, rsrp-ThresholdSSB RSRP-Range OPTIONAL, -- Need Rrsrp-ThresholdSSB-SUL RSRP-Range OPTIONAL, -- Cond SUL ra-PrioritizationForAccessIdentity-r16 SEQUENCE { ra-Prioritization-r16 RA-Prioritization, ra-PrioritizationForAI-r16 BIT STRING (SIZE (2))},...}Shared-RO-r17 ::= SEQUENCE { rachPartition-ConfigID-r17 INTEGER (1.. maxRACHAdditionalRACH-r17) OPTIONAL, -- Need S shared-RACH-resource ENUMERATED {ra4step,ra2step,raCE,spare} OPTIONAL, -- Need S}Separate-RO-r17 ::= SEQUENCE { rach-ConfigGeneric RACH-ConfigGeneric, totalNumberOfRA-Preambles INTEGER (1..63) OPTIONAL, -- Need S ssb-perRACH-OccasionAndCB-PreamblesPerSSB CHOICE { oneEighth ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64}, oneFourth ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64}, oneHalf ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64}, one ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64}, two ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32}, four INTEGER (1..16), eight INTEGER (1..8), sixteen INTEGER (1..4)} }For MsgA, we can have a similar structure above, or we can consider the following one as a simplified version.RACH-ConfigGenericTwoStepRA-r16 ::= SEQUENCE { msgA-PRACH-ConfigurationIndex-r16 INTEGER (0..262) OPTIONAL, -- Cond 2StepOnly msgA-RO-FDM-r16 ENUMERATED {one, two, four, eight} OPTIONAL, -- Cond 2StepOnly msgA-RO-FrequencyStart-r16 INTEGER (0..maxNrofPhysicalResourceBlocks-1) OPTIONAL, -- Cond 2StepOnly msgA-ZeroCorrelationZoneConfig-r16 INTEGER (0..15) OPTIONAL, -- Cond 2StepOnly msgA-PreamblePowerRampingStep-r16 ENUMERATED {dB0, dB2, dB4, dB6} OPTIONAL, -- Cond 2StepOnlyNoCFRA msgA-PreambleReceivedTargetPower-r16 INTEGER (-202..-60) OPTIONAL, -- Cond 2StepOnlyNoCFRA msgB-ResponseWindow-r16 ENUMERATED {sl1, sl2, sl4, sl8, sl10, sl20, sl40, sl80, sl160, sl320} OPTIONAL, -- Cond NoCFRA preambleTransMax-r16 ENUMERATED {n3, n4, n5, n6, n7, n8, n10, n20, n50, n100, n200} OPTIONAL, -- Cond 2StepOnlyNoCFRA...,[[ shared-RachPartition-index INTEGER (0.. maxNrofRACHResourcePool-1) OPTIONAL, -- Need S shared-RACH-resource ENUMERATED {ra4step,ra2step,raCE,spare} OPTIONAL, -- Need S]]} |
| CATT#1 | redundant definition of featureCombination-r17 | Field featureCombinationPreambles-r17 which is under RACH-ConfigCommon, includes field featureCombination-r17. While RACH-ConfigCommon-r17 also includes featureCombination-r17. We are wondering whether featureCombination-r17 is redundant defined. | Remove featureCombination-r17 in RACH-ConfigCommon-r17. |
| CATT#2 | The naming of RACH-ConfigCommon-r17 | This IE name seems to be a critical extension of RACH-ConfigCommon. In order to avoid misunderstanding, we suggest to use a new name, such as RACHPartitioning-Config. | Use a new name to avoid the misunderstanding. |
| CATT#3 | legacy-CB-PreamblesPerSSB-PerSharedRO-r16 | This is a new field without further clarification. Maybe it is better to add the description about this field. |  |
| Samsung #1 | featureCombination-r17 in FeatureCombinationPreambles-r17 | featureCombination-r17 is not needed in FeatureCombinationPreambles-r17 as RACH-ConfigCommon-r17 already incudes featureCombination-r17.It should be replaced by rach-ConfigID-r17 to indicate the other RACH configurations with which this RACH configuration shares ROs. | FeatureCombinationPreambles-r17 ::= SEQUENCE { ~~featureCombination-r17 FeatureCombination-r17 OPTIONAL,~~ rach-ConfigID-r17 INTEGER(1..maxRACHAdditionalRACH-r17) legacy-CB-PreamblesPerSSB-PerSharedRO-r16 INTEGER (1..64) OPTIONAL, msgA-CB-PreamblesPerSSB-PerSharedRO-r16 INTEGER (1..64) OPTIONAL, SSB-SharedRO-MaskIndex-r16 INTEGER (1..15) OPTIONAL, groupBconfigured SEQUENCE { ra-Msg3SizeGroupA ENUMERATED {b56, b144, b208, b256, b282, b480, b640, b800, b1000, b72, spare6, spare5,spare4, spare3, spare2, spare1}, messagePowerOffsetGroupB ENUMERATED { minusinfinity, dB0, dB5, dB8, dB10, dB12, dB15, dB18}, numberOfRA-PreamblesGroupA INTEGER (1..64) } groupB-ConfiguredTwoStepRA-r16 GroupB-ConfiguredTwoStepRA-r16} |
| Sasmung #2 | General RO sharing Scenario | A RACH configuration configured by rach-ConfigCommonToAddModList-r17 may share ROs with ROs configured by legacy ROs (i.e. configured by rach-ConfigCommon in BWP-UplinkCommon)rach-ConfigID 0 in FeatureCombinationPreambles-r17 can be used indicate this. |  |
| Samsung #3 | Group B preamble configuration in FeatureCombinationPreambles-r17 | Group configuration is not always needed. So these should be optional |  |
| Samsung #4 | RACH configuration with only 2 step RA  | Also one RACH configuration may include only 2 step configuration and not 4 step configuration. In this case RACH-ConfigCommon will not be present. How to indicate the RACH configuration is shared with another RACH configuration in this case?featureCombinationPreambles-r17 seems needed in MsgA-ConfigCommon-r16 also. |  |
| Samsung #5 | Premables usage in shared RO case | How to decide which feature uses which preambles in shared RO case needs to be further discussed. Add a Editors’s note for the same. |  |
| Samsung#6 | CB-PreamblesPerSSB-PerSharedRO-r16 | legacy-CB-PreamblesPerSSB-PerSharedRO-r16 should be renamed to CB-PreamblesPerSSB-PerSharedRO-r16We do not see need to add ‘legacy’ |  |
| LGE#1 | FeatureCombinationPreambles-r17 | There are duplicated feature/feature combination indication in two places: in RACH-ConfigCommon-r17 and in FeatureCombinationPreambles-r17. Since the list of ‘RACH-ConfigCommon-r17’ is configured in BWP-UplinkCommon, one RACH-ConfigCommon-r17 indicates 2-step and/or 4-step RA configuration for specific feature /feature combination. Given that FeatureCombinationPreambles-r17 is included in both RACH-ConfigCommon and RACH-ConfigCommonTwoStepRA, duplicated indication ‘FeatureCombination-r17’ is not needed(i.e. one common indication in RACH-ConfigCommon-r17 is enough). | Remove ‘FeatureCombination-r17’ from ‘FeatureCombinationPreambles-r17’ |
| Nokia#1 | FeatureCombinationPreambles-r17 | feaureCombination-r17 in in the RACH-ConfigCommon-r17 and in the featureCombinationPreambles-r17 seems redundant. Should be placed in one structure |  |
| Nokia#2 | rach-ConfigCommonToAddModListrach-ConfigCommonToReleaseList- | Prefer to use in the name a reference to FeatureCOmbination,  |  |
| Nokia#3 | maxRACHAdditionalRACH | Prefer to use in the name a reference to FeatureCOmbination, “additional” RACH is confusing, Rel-17 partitions will still be part of the total RACH resources |  |
| Nokia#4 | FeatureCombination IE | Prefer to have future-proof approach and use:FeatureCombination ::= SEQUENCE { redCap ENUMERATED {true} OPTIONAL, smallData ENUMERATED {true} OPTIONAL, slicing ENUMERATED {true} OPTIONAL, covEnh ENUMERATED {true} OPTIONAL, spare BIT STRING (SIZE(4))}Or FeatureCombination ::= SEQUENCE { redCap ENUMERATED {true} OPTIONAL, smallData ENUMERATED {true} OPTIONAL, slicing ENUMERATED {true} OPTIONAL, covEnh ENUMERATED {true} OPTIONAL, spare1 ENUMERATED {true} OPTIONAL, spare2 ENUMERATED {true} OPTIONAL,spare3 ENUMERATED {true} OPTIONAL,spare4 ENUMERATED {true} OPTIONAL}FeaturesCombinationIndicationBitmap ::= BIT STRING (SIZE (maxNrofFeatures)) maxNrofFeatures ::= INTEGER (2..8)  |  |
| OPPO#1 | Redundant featureCombination-r17  | Similar as other company, we think this IE within FeatureCombinationPreambles-r17 is sufficient and hence should be removed |  |
| OPPO#2 | RACH-ConfigCommon-r17 | Our interpretation of the ASN.1 structure is that ROs defined with RACH-ConfigCommon-r17 and/or MsgA-ConfigCommon-r16 is additional ROs for sharing only for Rel17 feature or feature combinations for both 4-step RACH and 2-step RACH.For one specific RACH partition, if it is located within RACH-ConfigCommon-r17, relevant 2-step RACH control parameters should be borrowed from MsgA-ConfigCommon-r16 includig PUSCH resource units in the same parent IE. If it is located within MsgA-ConfigCommon-r16, relevant 4-step RACH parameters should be borrowed from ConfigCommon-r17 in the same parent IE. This is based on assumption that within one RACH partition there are preambles/ROs for 4-step and/or 2-step RACH. |  |
| OPPO#3 | msgA-ConfigCommon-r16 | msgA-ConfigCommon-r16 should be msgA-ConfigCommon-r17. |  |
| OPPO#4 | shared cases | To confirm our understanding of this CR, following 4 cases are covered:Case1: legacy ROs configured only for 4-step RACH, by adding featureCombinationPreambles-r17 in existing RACH-ConfigCommonNote: such ROs maybe already shared by 2-step RACH Case2:legacy ROs configured only for 2-step RACH, by adding featureCombinationPreambles-r17 in existing MsgA-ConfigCommon-r16 Case 3: new Rel17 ROs configured by RACH-ConfigCommon-r17 and/or MsgA-ConfigCommon-r16, which is shared by more than one feature or feature combinationCase 4: new Rel17 ROs configured by RACH-ConfigCommon-r17 and/or MsgA-ConfigCommon-r16, which is shared by only one feature or feature combination. This is called non-shared RO case. |  |
| OPPO#5 | FeatureCombinationPreambles-r17 | We have same understanding as Intel that within shared ROs, preamles should be partitioned first for 4-step RACH for all relevant feature combinations and then for 2-step RACH.For either 4-step RACH resource or 2-step RACH resource, as for starting preamble index:Since there is only number of preambles i.e. legacy-CB-PreamblesPerSSB-PerSharedRO-r16 and msgA-CB-PreamblesPerSSB-PerSharedRO-r16 are configured, it is assumed that 1st preamble for R17 feature combination is the configured legacy preamble+1. For new ROs, the starting preamble index is simply zero. This is not clear and one Editor note is neededPreamble range:Preamble range of feature combination is configured in back-to-back manner i.e. preamble of No.(n+1) feature combination starts with the last preamble +1 of No.n feature combination. This is not clear and one Editor note is needed |  |
| OPPO#6 | FeatureCombinationPreambles-r17 | groupB-ConfiguredTwoStepRA-r16 should be optional. If network doesn’t want to configure 2-step RACH resource for one specific feature or feature combination, then msgA-CB-PreamblesPerSSB-PerSharedRO-r16 and groupB-ConfiguredTwoStepRA-r16 should be optional. Even 2-step RACH resource is configured, it should be allowed not to split between group A and group B. |  |
| Vivo#01&02 | FeatureCombinationPreambles-r17 | **#01:**We think it might be clearer to have two IEs (which may help to reduce the OPTIONAL signalling overhead),One is **FeatureCombinationPreambles-r17** for 4-step RA shared ROs case,The other one is **FeatureCombinationPreamblesTwoStepRA-r17** for 2-step shared ROs case.**#02:**Besdies, rsrp-ThresholdSSB should be also included as it is agreed that separate RSRP threshold for SSB selection can be configured for CovEnh.1. A separate rsrp-ThresholdSSB threshold is introduced for requesting Msg3 repetition.
 | *FeatureCombinationPreambles*TBD.*FeatureCombinationPreambles* information element-- ASN1START-- TAG-FEATURECOMBINATIONPREAMBLES-STARTFeatureCombinationPreambles-r17 ::= SEQUENCE { featureCombination-r17 FeatureCombination-r17 OPTIONAL, legacy-CB-PreamblesPerSSB-PerSharedRO-r16 INTEGER (1..60) OPTIONAL, SSB-SharedRO-MaskIndex-r16 INTEGER (1..15) OPTIONAL, groupBconfigured SEQUENCE { ra-Msg3SizeGroupA ENUMERATED {b56, b144, b208, b256, b282, b480, b640, b800, b1000, b72, spare6, spare5,spare4, spare3, spare2, spare1}, messagePowerOffsetGroupB ENUMERATED { minusinfinity, dB0, dB5, dB8, dB10, dB12, dB15, dB18}, numberOfRA-PreamblesGroupA INTEGER (1..64)} OPTIONAL,rsrp-ThresholdSSB RSRP-Range OPTIONAL, }-- TAG-FEATURECOMBINATIONPREAMBLES-STOP-- ASN1STOP*FeatureCombinationPreamblesTwoStepRA*TBD.*FeatureCombinationPreambles* information element-- ASN1START-- TAG-FEATURECOMBINATIONPREAMBLES-STARTFeatureCombinationPreamblesTwoStepRA-r17 ::= SEQUENCE { featureCombination-r17 FeatureCombination-r17 OPTIONAL, msgA-CB-PreamblesPerSSB-PerSharedRO-r16 INTEGER (1..60) OPTIONAL, SSB-SharedRO-MaskIndex-r16 INTEGER (1..15) OPTIONAL, groupB-ConfiguredTwoStepRA-r16 GroupB-ConfiguredTwoStepRA-r16 OPTIONAL,msgA-RSRP-ThresholdSSB-r16 RSRP-Range OPTIONAL,}-- TAG-FEATURECOMBINATIONPREAMBLES-STOP-- ASN1STOP |
| Huawei#1 | FeatureCombinationPreambles IERACH-ConfigCommon IERACH-ConfigCommonTwoStepRA IE | The current structure assumes the same parameter values are used for all feature combinations that use a certain RACH configuration. However, many of the parameters were agreed to be feature specific. | 1. It should be possible to configure some parameters as feature (combination) specific parameters, e.g.* Parameters from RACH-ConfigCommon:
	+ groupBconfigured
	+ ra-conetntionResolutionTimer (?)
	+ rsrp-ThresholdSSB
	+ rsrpThresholdSSB-SUL
	+ ra-PrioritizationParameters
	+ parameters from RACH-ConfigGeneric: preambleReceivedTargetPower, powerRampingStep
* Parameters from RACH-ConfigCommonTwoStepRA
	+ msgA-RSRP-ThresholdSSB
	+ msgA-TransMax
	+ msgA-RSRP-Threshold
	+ groupB-ConfiguredTwoStepRA
	+ msgA-PreambleReceivedTargetPower
	+ msgA-PreamblePowerRampingStep
	+ ra-ContentionResolutionTimer (?)
	+ ra-PrioritizationForAccessIdentityTwoStep

This can be achieved by moving these parameters to FeatureCombinationPreambles. |
| Huawei#2 | RACH-ConfigCommon-r17 IE | Currently this IE contains msgA-ConfigCommon-r16 which means that msgA PUSCH conifguration is made per Rel-17 RACH configuration instead of per feature combination. It should be possible to configure msgA PUSCH per feature combination, e.g. PUSCH for SDT will normally provide a bigger grant than for non-SDT. | It should be possible to configure msgA-PUSCH-Config within FeatureCombinationPreambles IE (the name of the IE may need to be changed to better express its contents in this case, e.g. FeatureCombinationRA-Resources). |
| Huawei#3 | RACH-ConfigCommon IERACH-ConfigCommonTwoStepRA IE | As mentioned in “Huawei#1” issue, many parameters should be feature combination specific. Therefore it is unclear whether it makes sense to reuse RACH-ConfigCommon IE and RACH-ConfigCommonTwoStepRA IE at all. | Define new structure instead of reusing RACH-ConfigCommon and RACH-ConfigCommonTwoStepRA IE and move all feature combination specific parameters to FeatureCombinationPreambles IE as in “Huawei#1”. |
| Huawei#4 | FeatureCombination IE | It is already clear that this has to be configurable per a list of slice groups, please see the relevant agreement from Slice WI:* In a cell, there may be multiple slice-specific RACH configurations.
* One or more of the slice groups are linked to a slice-specific RACH configuration.
* There may be slice groups that are not linked to a slice-specific RACH configuration (they use the common RACH configuration).
* All slices of a slice group use the slice-specific RACH configuration of the slice group.

RACH prioritization parameters can be configured per slice group. | Convert *slicing* parameter to a list of slice group IDs. |
| Huawei#5 | FeatureCombination IE | It does not seem to make sense to use ‘…’ as extension mark. Legacy UEs will not be able to read anything after ‘…’ and will incorrectly use such RACH partition.  | How to do the extension should be put FFS. |
| Huawei#6 | featureCombination-r17 | This parameter is now present in both FeatureCombinationPreambles IE and in RACH-ConfigCommon-r17 and it is unclear what the purpose is to have it in RACH-ConfigCommon-r17. | featureCombination-r17 parameter should be removed from RACH-ConfigCommon-r17 IE, not from FeatureCombinationPreambles IE, as suggested by some companies. If we remove it from FeatureCombinationPreambles IE, then it will not be possible for multiple feature combinations to share the same RACH config. We would need to introduce some reference to shared RO set via index which we find cumbersome. We prefer the currently proposed structure in general. Also, featureCombination-r17 parameter should be mandatory in FeatureCombinationPreambles IE. |

# Summary and proposals

TBD

# Annex (contact details for email discussions)

|  |  |  |
| --- | --- | --- |
| Company | Contact name | Contact email |
| Intel | Seau Sian Lim | seau.s.lim@intel.com |
| ZTE | He Huang | Huang.he4@zte.com.cn |
| Samsung | Anil Agiwal | anilag@samsung.com |
| LGE | Hanseul Hong | hanseul.hong@lge.com |
| Nokia, Nokia Shanghai Bell | Malgorzata Tomala | malgorzata.tomala@nokia.com |
| vivo | Yitao Mo (Stephen) | yitao.mo@vivo.com |