**3GPP TSG-RAN WG2 Meeting #113-e R2-21xxxxx**

**Online, 25 January-5 February 2021**

**Agenda Item: 6.1.1**

**Source: MediaTek Inc.**

**Title: Summary of email discussion [Post112-e][060][NR16] Extension of ToAddMod lists (MediaTek)**

**Document for: Discussion, decision**

1 Introduction

This document summarises the following email discussion:

* [Post112-e][060][NR16] Extension of ToAddMod lists (Mediatek)

 Scope: Finalize the remaining open points

 Intended outcome: Report (discussion summary), Agreeable CR 38331

 Deadline: long

This discussion is a sequel to discussion [AT112-e][045], which received a few comments towards an updated draft CR using document R2-2009982 as a baseline.

The rapporteur requests comments on this document by 2021-01-05, allowing some time for finalisation of the CR before the meeting submission deadline.

2 Discussion

## 2.1 ToRelease list size

During the various discussions of this issue, two approaches to dimensioning the listElementToReleaseSizeExt-rN were considered, summarised as “short” and “long”. In the “short” approach, the SizeExt list relates only to the “new” entries, i.e. its size is maxNrofListElementsDiff-rN, and (using the nomenclature from example 3 of R2-2009982) its entry type is ListElementId-vNxy (so that it can release only the entries that were added to the list by the listElementToAddModListSizeExt-rN). In the “long” approach, the SizeExt list has size maxNrofListElements-rN, its entry type is ListElementId-rN, and it can release any of the list entries.

Both approaches have been suggested in email discussion of this topic; at the conclusion of [AT112-e][045], the rapporteur’s impression was that companies may have some preference for the “short” version. This avoids having two different lists that can release the same entry, but means that it may be necessary for the UE to process two fields to release a set of list entries (e.g. some entries with IDs in the original range and others with IDs in the extended range).

**Q1: Is it agreeable to use the “short” approach to dimensioning the listElementToReleaseSizeExt-rN?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comment** |
| Huawei | Yes |  |
| Samsung | Yes | This option avoids the need to clarify when network can use the new/ extended field (but not a very strong view) |

**Rapporteur’s summary:** Both responding companies agree to the “short” approach (i.e. no change to the draft CR in this respect).

## 2.2 Omission of a ToRelease list in example 3

In case the list size is extended and fields are added to the list element structure (the “example 3” case from R2-2009982), the extension process results in three ToAddMod lists: the original listElementToAddModList, the size-extended listElementToAddModListSizeExt-rN, and the parallel listElementToAddModListExt-rN. In the previously drafted examples, these lists have corresponded to three matching ToRelease lists:

* listElementToReleaseList, the original non-extended list that can release only entries from the original portion of the list;
* listElementToReleaseListSizeExt-rN, the size-extended list that can release any list entry (in case of the “long” structure) or only the extended entries (in case of the “short” structure);
* listElementToReleaseListExt-rN, the full-length list that can release any list entry.

Functionally, either of the second and third lists could be omitted, while still making it possible to release any entry of the list. It could be desirable to eliminate one of the lists to avoid redundancy. There does not seem to be a functional reason to choose one list over the other—it is a question of preferred style. In discussion [AT112-e][045], the only comment received favoured omitting the third list (listElementToReleaseListExt-rN).

**Q2: In the “example 3” scenario with three ToAddMod lists, is it agreeable to omit the third ToRelease list?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comment** |
| Huawei | Yes |  |
| Samsung | Yes | This option seems consistent with the approach suggested in Q1. We anyhow assume that clarification is provide that field is used to release the additional entries covered by listElementToAddModListSizeExt |

**Rapporteur’s summary:** Both responding companies agree to omit the third ToRelease list (i.e. no change to the draft CR in this respect).

## 2.3 Additional comments on the draft CR

Discussion [AT112-e][045] invited further comments on R2-2009982. The following points were raised:

1. Make the existing field descriptions consistent in terms of the relations between the different lists
2. Align the field description of pathlossReferenceRSToAddModList with the recommended practices
3. Correct the field naming in the field description of spatialRelationInfoToAddModList
4. In example 1, change the ToRelease list entries to ListElementId-vNxy to reflect the increase in index space
5. Note the error in the list entry type for spatialRelationInfoToReleaseListSizeExt-r16

Rapporteur agrees with points 1-3. On point 4, it seems that updating the ID type would normally mean that the ListElement structure also needed to be extended with the new ID type, changing the scenario into example 3; this means that example 1 is really only applicable to cases where the ID type is not a simple index into the list (e.g. if there were an extension to the cellsToAddModList in MeasObjectNR, where the ID type is a PCI). On point 5, it seems clear that this was a mistake in Rel-16, but no backward-compatible fix is evident; fortunately the problem does not break the function of the IE, since the spatialRelationInfoToReleaseListExt-r16 can release any entry of the combined lists.

Accordingly, the updated draft CR provided as an annex to this discussion document implements points 1-3 from the list above. Any additional comments on the draft CR are solicited.

**Q3: Any further comments on the provided draft CR?**

|  |  |
| --- | --- |
| **Company** | **Comment** |
| Huawei | schedulingRequestResourceToAddModList-v1610 and SchedulingRequestResourceConfig-v1610 (defined in the SchedulingRequestResourceConfig IE) should also have the "Ext" suffix added.A.4.3x: Having an existing element ID with a value range beyond the existing list size, as assumed in example 1, is less frequent than having to extend the element ID without the need for one more list as in example 3, because there are extension markers in the list element type and the additional ID values can be placed there directly. We suggest revising the example 1 in order to cover that more frequent case, which is not covered now.- if the element type includes extension markers, there is no need to do an extension like in extension 3, unlike what the text now says- the example reflects the rare where |
| Samsung | We like to raise one aspect not clear in the CR and probably not discussed so far* For the case parameters are added after list size was extended, it seems the situation is clear from the discussion. We are not sure if situation is clear in case the extensions are in the reverse order i.e. first parameters are added by List-Rn and then the list size is extended. Regardless, an example to clarify this case seems useful
	+ Example: list1 is defined in R15.
	+ In R16, additional fields are added i.e. by listExt-Rn
	+ In R17, size of the list is extended. In this case, will we extend size of both lists? If so, what suffices will we use?
		- Field for additional entries of (original) list  listSizeExt-R17?
		- Field for additional entries of listExt-Rn  listExtSizeExt-R17?
	+ Or will we create one list and use a new IE in which contents of original IE and one used for listExt-R17 is merged. Latter approach is probably better..?

Some remarks form our side (did not comment parts related to release as still under discussion):* Suggestion about 1st change in A.4.2

In the case of list fields (SEQUENCE OF types in ASN.1) using the ToAddMod/ToRelease construction, the use of critical extensions to increase the size of a list should be avoided; that is, replacing the original list field by a new field also used to signal entries previously covered by the original field. I.e. extensions done according to the following example should be avoided:* When doing size extensions in releases Rn and Rm, we understand that we will end up with fields named ListSizeExt-Rn and ListSizeExt-Rm
	+ This seems fine i.e. we assume this kind of extension is done only at start of new release/ once per release
* When adding parameters in Rn and Rm, we understand that we will end up with fields named ListExt-Rn and ListExt-Rm
	+ If we have to add up a further parameter while correcting a release, will we end up using ListExt2, ListExt3..? If so, this would be good to clarify as it seems a quite common case

(We note that for these extensions we will not have a suffix indicating in which version parameters are added/ extended, as we have for other non-critical fields)* We understand that the proposed naming for non-critical extensions of IEs defining the information structure of a list entry deviates from what we do for other IEs. I.e. for any regular IE we add –vNxy but for IEs defining the entry of list we add Ext-rN.
	+ We are not aware of reasons to deviate, and in general prefer to avoid deviations as much as possible so conventions/ guidelines can remain as a simple and straightforward as possible (to reduce likelihood of incorrect usages), but see no real strong need to re-open if this has been debated for a long time already
 |
| Intel | * Example 1: “A new ToRelease list is needed, and its range should generally include the total number of entries to allow the new ToRelease list also to release the original entries” seems not aligned with the previous conclusions.
* Example 2: With the addition of the new sentence, the next sentence is a bit misleading as the example is for only one extension: “If multiple extensions of the same list are needed, numeric suffixes are used (e.g. *listElementToAddModListExt2-rN*). The result is as shown in the following example:”. It could actually be useful to have the example of multiple extensions.
* We support comment 6 below to use the -vNxy suffix.

  |

**Rapporteur’s summary:** Several additional comments were received:

1. The fields schedulingRequestResourceToAddModList-v1610 and SchedulingRequestResourceConfig-v1610 should have the “Ext” suffix for alignment with the other extension lists.
2. Example 1 in the draft CR could be extended to cover the case where at least the list item ID field needs to be extended (with an extension marker available).
3. A fourth example, to show the multi-release case where at first fields are added (listExt-rN) and later the list size is extended (two additional lists—e.g. listSizeExt-rM, listExtSizeExt-rM—or one combined list?), would be beneficial.
4. Additional text is proposed for the first change in section A.4.2 (“replacing the original list field by a new field also used to signal entries previously covered by the original field”).
5. If fields are added multiple times during a release, the extension lists may need to be named ListExt-rN, ListExt2-rN, ListExt3-rN, and so on, and this should be clarified in the text. It is also noted that these lists will not have a -vNxy suffix indicating which version of the spec they were added in.
6. In general, the proposed naming does not match what we do with non-critical extensions of IEs (-rN suffix instead of -vNxy).

Rapporteur understanding on these points is as follows:

1. This seems an oversight and will be fixed in the revision of the CR.
2. This seems a fair point. Currently we have only a brief note on the case where an extension marker is available, and it is not explicitly identified that this case could change the list entry type for the ToRemove list. Rapporteur will propose draft text in the revision of the CR.
3. The fourth example will be added in the revision of the CR. However, we will need to confirm if companies can agree to the “one combined list” extension mechanism. Rapporteur would like to call the attention of interested companies to the proposed new example in the second phase of the discussion.
4. No strong view from the rapporteur. The revision of the CR will reflect the proposed text.
5. Some text to describe this case should be captured, but the details interact with issue 6 below.
6. The use of the -vNxy suffix was also suggested by one company in email discussion [Post111-e][901].
	1. Rapporteur understands that there is a semantic question here: When we create the Rel-N field listElementToAddModListExt (or SizeExt), is this a NCE of the existing list (which would conventionally be named listElementToAddModList-vNxy) or a new list field that holds the new information (which would normally receive the -rN suffix as a new field in Rel-N)?
	2. Since we agreed previously to change the name from the “base” field by adding Ext/SizeExt, it looks formally like a new field, but the proposed extension mechanism is supposed to be in a NCE style, which may militate for using the -vNxy suffix anyway (also capturing the information about which revision of the spec a field was added in).
	3. It is not possible to apply the NCE style and nomenclature exactly for example 3, because it would result in two different extension lists named listElementToAddModList-vNxy (the current Ext-rN and SizeExt-rN).
	4. Rapporteur’s suggestion is to keep the current nomenclature for the draft CR, but consider in the limited time available whether a change should be made.

Additional comments were received after the initial summary on the following points:

1. The text in example 1 still describes the “long” rather than “short” size extension list, and this should be corrected.
2. The wording in example 2 suggests that the numerical suffixes for additional extensions would be shown in the example, which may be useful to include.
3. The responding company supports point 6 above (-vNxy suffix).

Rapporteur comments:

1. This is a clear oversight and will be fixed.
2. The comment seems correct, but if we take the suggestion in point 3, the style of naming the extension lists changes and it seems no longer necessary to have the numerical suffixes, since the -vNxy suffixes will already distinguish the extensions made in different versions of the same release.
3. The only companies expressing a preference prefer the -vNxy suffix as in other NCEs, so a revision of the CR will migrate to this naming convention.

3 Conclusion

As discussed above, the draft CR seems generally agreeable, with the additional comments listed in section 2.3. Rapporteur proposes to update the draft CR in accordance with the changes proposed in section 2.3 and continue discussion on that basis. The updated draft CR will be provided in a separate document (the original draft is preserved as an Annex in this document for reference).

Annex Draft CR

– *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. If the IE is used for a dormant BWP, the fields other than *controlResourceSetToAddModList* and *controlResourceSetToReleaseList* 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

 ...,

 [[

 controlResourceSetToAddModListSizeExt-r16 SEQUENCE (SIZE (1..2)) OF ControlResourceSet OPTIONAL, -- Need N

 controlResourceSetToReleaseListSizeExt-r16 SEQUENCE (SIZE (1..5)) OF ControlResourceSetId-r16 OPTIONAL, -- Need N

 searchSpacesToAddModListExt-r16 SEQUENCE(SIZE (1..10)) OF SearchSpaceExt-r16 OPTIONAL, -- Need N

 uplinkCancellation-r16 SetupRelease { UplinkCancellation-r16 } OPTIONAL, -- Need M

 monitoringCapabilityConfig-r16 ENUMERATED { r15monitoringcapability,r16monitoringcapability } OPTIONAL, -- Need M

 searchSpaceSwitchConfig-r16 SearchSpaceSwitchConfig-r16 OPTIONAL -- Need R

 ]]

}

SearchSpaceSwitchConfig-r16 ::= SEQUENCE {

 cellGroupsForSwitchList-r16 SEQUENCE(SIZE (1..4)) OF CellGroupForSwitch-r16 OPTIONAL, -- Need R

 searchSpaceSwitchDelay-r16 INTEGER (10..52) OPTIONAL -- Need R

}

CellGroupForSwitch-r16 ::= SEQUENCE(SIZE (1..16)) OF ServCellIndex

-- TAG-PDCCH-CONFIG-STOP

-- ASN1STOP

|  |
| --- |
| ***PDCCH-Config* field descriptions** |
| ***controlResourceSetToAddModList, controlResourceSetToAddModListSizeExt***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). The UE shall consider entries in *controlResourceSetToAddModList* and in *controlResourceSetToAddModListSizeExt* as a single list, i.e. an entry created using *controlResourceSetToAddModList* can be modifed using *controlResourceSetToAddModListSizeExt* (or deleted using *controlResourceSetToReleaseListSizeExt*) and vice-versa. 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, controlResourceSetToReleaseListSizeExt***List of UE specifically configured Control Resource Sets (CORESETs) to be released by the UE. This field only applies to CORESETs configured by *controlResourceSetToAddModList* or *controlResourceSetToAddModListSizeExt* 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). |
| ***monitoringCapabilityConfig***Configures either Rel-15 PDCCH monitoring capability or Rel-16 PDCCH monitoring capability for PDCCH monitoring on a serving cell. Value *r15monitoringcapablity* enables the Rel-15 monitoring capability, and value *r16monitoringcapablity* enables the Rel-16 PDCCH monitoring capability (see TS 38.213 [13], clause 10.1). |
| ***searchSpacesToAddModList, searchSpacesToAddModListExt***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). If the network includes searchSpaceToAddModListExt, it includes the same number of entries, and listed in the same order, as in searchSpacesToAddModList. |
| ***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. |
| ***uplinkCancellation***Configuration of uplink cancellation indications to be monitored in this cell (see TS 38.213 [13], clause 11.2A). |

|  |
| --- |
| ***SearchSpaceSwitchConfig* field descriptions** |
| ***cellGroupsForSwitchList***The list of serving cells which are bundled for the search space group switching purpose (see TS 38.213 [13], clause 10.4). A serving cell can belong to only one *CellGroupForSwitch*. The network configures the same list for all BWPs of serving cells in the same *CellGroupForSwitch.* |
| ***searchSpaceSwitchDelay***Indicates the value to be applied by a UE for Search Space Set Group switching; corresponds to the P value in TS 38.213 [13], clause 10.4. The network configures the same value for all BWPs of serving cells in the same *CellGroupForSwitch.* |

[…]

– *PUCCH-Config*

The IE *PUCCH-Config* is used to configure UE specific PUCCH parameters (per BWP).

***PUCCH-Config* information element**

-- ASN1START

-- TAG-PUCCH-CONFIG-START

PUCCH-Config ::= SEQUENCE {

 resourceSetToAddModList SEQUENCE (SIZE (1..maxNrofPUCCH-ResourceSets)) OF PUCCH-ResourceSet OPTIONAL, -- Need N

 resourceSetToReleaseList SEQUENCE (SIZE (1..maxNrofPUCCH-ResourceSets)) OF PUCCH-ResourceSetId OPTIONAL, -- Need N

 resourceToAddModList SEQUENCE (SIZE (1..maxNrofPUCCH-Resources)) OF PUCCH-Resource OPTIONAL, -- Need N

 resourceToReleaseList SEQUENCE (SIZE (1..maxNrofPUCCH-Resources)) OF PUCCH-ResourceId OPTIONAL, -- Need N

 format1 SetupRelease { PUCCH-FormatConfig } OPTIONAL, -- Need M

 format2 SetupRelease { PUCCH-FormatConfig } OPTIONAL, -- Need M

 format3 SetupRelease { PUCCH-FormatConfig } OPTIONAL, -- Need M

 format4 SetupRelease { PUCCH-FormatConfig } OPTIONAL, -- Need M

 schedulingRequestResourceToAddModList SEQUENCE (SIZE (1..maxNrofSR-Resources)) OF SchedulingRequestResourceConfig

 OPTIONAL, -- Need N

 schedulingRequestResourceToReleaseList SEQUENCE (SIZE (1..maxNrofSR-Resources)) OF SchedulingRequestResourceId

 OPTIONAL, -- Need N

 multi-CSI-PUCCH-ResourceList SEQUENCE (SIZE (1..2)) OF PUCCH-ResourceId OPTIONAL, -- Need M

 dl-DataToUL-ACK SEQUENCE (SIZE (1..8)) OF INTEGER (0..15) OPTIONAL, -- Need M

 spatialRelationInfoToAddModList SEQUENCE (SIZE (1..maxNrofSpatialRelationInfos)) OF PUCCH-SpatialRelationInfo

 OPTIONAL, -- Need N

 spatialRelationInfoToReleaseList SEQUENCE (SIZE (1..maxNrofSpatialRelationInfos)) OF PUCCH-SpatialRelationInfoId

 OPTIONAL, -- Need N

 pucch-PowerControl PUCCH-PowerControl OPTIONAL, -- Need M

 ...,

 [[

 resourceToAddModListExt-r16 SEQUENCE (SIZE (1..maxNrofPUCCH-Resources)) OF PUCCH-ResourceExt-r16 OPTIONAL, -- Need N

 dl-DataToUL-ACK-r16 SetupRelease { DL-DataToUL-ACK-r16 } OPTIONAL, -- Need M

 ul-AccessConfigListDCI-1-1-r16 SetupRelease { UL-AccessConfigListDCI-1-1-r16 } OPTIONAL, -- Need M

 subslotLengthForPUCCH-r16 CHOICE {

 normalCP-r16 ENUMERATED {n2,n7},

 extendedCP-r16 ENUMERATED {n2,n6}

 } OPTIONAL, -- Need R

 dl-DataToUL-ACK-DCI-1-2-r16 SetupRelease { DL-DataToUL-ACK-DCI-1-2-r16} OPTIONAL, -- Need M

 numberOfBitsForPUCCH-ResourceIndicatorDCI-1-2-r16 INTEGER (0..3) OPTIONAL, -- Need R

 dmrs-UplinkTransformPrecodingPUCCH-r16 ENUMERATED {enabled} OPTIONAL, -- Cond PI2-BPSK

 spatialRelationInfoToAddModListSizeExt-r16 SEQUENCE (SIZE (1..maxNrofSpatialRelationInfosDiff-r16)) OF PUCCH-SpatialRelationInfo

 OPTIONAL, -- Need N

 spatialRelationInfoToReleaseListSizeExt-r16 SEQUENCE (SIZE (1..maxNrofSpatialRelationInfosDiff-r16)) OF PUCCH-SpatialRelationInfoId

 OPTIONAL, -- Need N

 spatialRelationInfoToAddModListExt-r16 SEQUENCE (SIZE (1..maxNrofSpatialRelationInfos-r16)) OF PUCCH-SpatialRelationInfoExt-r16

 OPTIONAL, -- Need N

 spatialRelationInfoToReleaseListExt-r16 SEQUENCE (SIZE (1..maxNrofSpatialRelationInfos-r16)) OF

 PUCCH-SpatialRelationInfoId-r16 OPTIONAL, -- Need N

 resourceGroupToAddModList-r16 SEQUENCE (SIZE (1..maxNrofPUCCH-ResourceGroups-r16)) OF PUCCH-ResourceGroup-r16

 OPTIONAL, -- Need N

 resourceGroupToReleaseList-r16 SEQUENCE (SIZE (1..maxNrofPUCCH-ResourceGroups-r16)) OF PUCCH-ResourceGroupId-r16

 OPTIONAL, -- Need N

 sps-PUCCH-AN-List-r16 SetupRelease { SPS-PUCCH-AN-List-r16 } OPTIONAL, -- Need M

 schedulingRequestResourceToAddModList-v1610 SEQUENCE (SIZE (1..maxNrofSR-Resources)) OF SchedulingRequestResourceConfig-v1610

 OPTIONAL -- Need N

 ]]

}

PUCCH-FormatConfig ::= SEQUENCE {

 interslotFrequencyHopping ENUMERATED {enabled} OPTIONAL, -- Need R

 additionalDMRS ENUMERATED {true} OPTIONAL, -- Need R

 maxCodeRate PUCCH-MaxCodeRate OPTIONAL, -- Need R

 nrofSlots ENUMERATED {n2,n4,n8} OPTIONAL, -- Need S

 pi2BPSK ENUMERATED {enabled} OPTIONAL, -- Need R

 simultaneousHARQ-ACK-CSI ENUMERATED {true} OPTIONAL -- Need R

}

PUCCH-MaxCodeRate ::= ENUMERATED {zeroDot08, zeroDot15, zeroDot25, zeroDot35, zeroDot45, zeroDot60, zeroDot80}

-- A set with one or more PUCCH resources

PUCCH-ResourceSet ::= SEQUENCE {

 pucch-ResourceSetId PUCCH-ResourceSetId,

 resourceList SEQUENCE (SIZE (1..maxNrofPUCCH-ResourcesPerSet)) OF PUCCH-ResourceId,

 maxPayloadSize INTEGER (4..256) OPTIONAL -- Need R

}

PUCCH-ResourceSetId ::= INTEGER (0..maxNrofPUCCH-ResourceSets-1)

PUCCH-Resource ::= SEQUENCE {

 pucch-ResourceId PUCCH-ResourceId,

 startingPRB PRB-Id,

 intraSlotFrequencyHopping ENUMERATED { enabled } OPTIONAL, -- Need R

 secondHopPRB PRB-Id OPTIONAL, -- Need R

 format CHOICE {

 format0 PUCCH-format0,

 format1 PUCCH-format1,

 format2 PUCCH-format2,

 format3 PUCCH-format3,

 format4 PUCCH-format4

 }

}

PUCCH-ResourceExt-r16 ::= SEQUENCE {

 interlaceAllocation-r16 SEQUENCE {

 rb-SetIndex INTEGER (0..4),

 interlace0 CHOICE {

 scs15 INTEGER (0..9),

 scs30 INTEGER (0..4)

 }

 } OPTIONAL, --Need R

 formatExt-v1610 CHOICE {

 interlace1-v1610 INTEGER (0..9),

 occ-v1610 SEQUENCE {

 occ-Length-v1610 ENUMERATED {n2,n4} OPTIONAL, -- Need M

 occ-Index-v1610 ENUMERATED {n0,n1,n2,n3} OPTIONAL -- Need M

 }

 } OPTIONAL, -- Need R

 ...

}

PUCCH-ResourceId ::= INTEGER (0..maxNrofPUCCH-Resources-1)

PUCCH-format0 ::= SEQUENCE {

 initialCyclicShift INTEGER(0..11),

 nrofSymbols INTEGER (1..2),

 startingSymbolIndex INTEGER(0..13)

}

PUCCH-format1 ::= SEQUENCE {

 initialCyclicShift INTEGER(0..11),

 nrofSymbols INTEGER (4..14),

 startingSymbolIndex INTEGER(0..10),

 timeDomainOCC INTEGER(0..6)

}

PUCCH-format2 ::= SEQUENCE {

 nrofPRBs INTEGER (1..16),

 nrofSymbols INTEGER (1..2),

 startingSymbolIndex INTEGER(0..13)

}

PUCCH-format3 ::= SEQUENCE {

 nrofPRBs INTEGER (1..16),

 nrofSymbols INTEGER (4..14),

 startingSymbolIndex INTEGER(0..10)

}

PUCCH-format4 ::= SEQUENCE {

 nrofSymbols INTEGER (4..14),

 occ-Length ENUMERATED {n2,n4},

 occ-Index ENUMERATED {n0,n1,n2,n3},

 startingSymbolIndex INTEGER(0..10)

}

PUCCH-ResourceGroup-r16 ::= SEQUENCE {

 pucch-ResourceGroupId-r16 PUCCH-ResourceGroupId-r16,

 resourcePerGroupList-r16 SEQUENCE (SIZE (1..maxNrofPUCCH-ResourcesPerGroup-r16)) OF PUCCH-ResourceId

}

PUCCH-ResourceGroupId-r16 ::= INTEGER (0..maxNrofPUCCH-ResourceGroups-1-r16)

DL-DataToUL-ACK-r16 ::= SEQUENCE (SIZE (1..8)) OF INTEGER (-1..15)

DL-DataToUL-ACK-DCI-1-2-r16 ::= SEQUENCE (SIZE (1..8)) OF INTEGER (0..15)

UL-AccessConfigListDCI-1-1-r16 ::= SEQUENCE (SIZE (1..16)) OF INTEGER (0..15)

-- TAG-PUCCH-CONFIG-STOP

-- ASN1STOP

|  |
| --- |
| ***PUCCH-Config* field descriptions** |
| ***dl-DataToUL-ACK, dl-DataToUL-ACK-DCI-1-2***List of timing for given PDSCH to the DL ACK (see TS 38.213 [13], clause 9.1.2). The field *dl-DataToUL-ACK* applies to DCI format 1\_1 and the field *dl-DataToUL-ACK-DCI-1-2* applies to DCI format 1\_2 (see TS 38.212 [17], clause 7.3.1 and TS 38.213 [13], clause 9.2.3). If *dl-DataToUL-ACK-r16* is signalled, UE shall ignore the *dl-DataToUL-ACK* (without suffix). The value -1 corresponds to "non-numerical value" for the case where the A/N feedback timing is not explicitly included at the time of scheduling PDSCH. |
| ***dmrs-UplinkTransformPrecodingPUCCH***This field is used for PUCCH formats 3 and 4 according to TS 38.211, Clause 6.4.1.3.3.1. |
| ***format1***Parameters that are common for all PUCCH resources of format 1. |
| ***format2***Parameters that are common for all PUCCH resources of format 2. |
| ***format3***Parameters that are common for all PUCCH resources of format 3. |
| ***format4.***Parameters that are common for all PUCCH resources of format 4 |
| ***numberOfBitsForPUCCH- ResourceIndicatorDCI-1-2***Configuration of the number of bits for "PUCCH resource indicator" in DCI format 1\_2 (see TS 38.212 [17], clause 7.3.1 and TS 38.213 [13], clause 9.2.3). |
| ***resourceGroupToAddModList, resourceGroupToReleaseList***Lists for adding and releasing groups of PUCCH resources that can be updated simultaneously for spatial relations with a MAC CE |
| ***resourceSetToAddModList, resourceSetToReleaseList***Lists for adding and releasing PUCCH resource sets (see TS 38.213 [13], clause 9.2). |
| ***resourceToAddModList, resourceToAddModListExt, resourceToReleaseList***Lists for adding and releasing PUCCH resources applicable for the UL BWP and serving cell in which the *PUCCH-Config* is defined. The resources defined herein are referred to from other parts of the configuration to determine which resource the UE shall use for which report. If the network includes of *resourceToAddModListExt*, it includes the same number of entries, and listed in the same order, as in *resourceToAddModList*. |
| ***spatialRelationInfoToAddModList, spatialRelationInfoToAddModListSizeExt , spatialRelationInfoToAddModListExt***Configuration of the spatial relation between a reference RS and PUCCH. Reference RS can be SSB/CSI-RS/SRS. If the list has more than one element, MAC-CE selects a single element (see TS 38.321 [3], clause 5.18.8 and TS 38.213 [13], clause 9.2.2). The UE shall consider entries in *spatialRelationInfoToAddModList* and in *spatialRelationInfoToAddModListSizeExt* as a single list, i.e. an entry created using *spatialRelationInfoToAddModList* can be modifed using *spatialRelationInfoToAddModListSizeExt* (or deleted using *spatialRelationInfoToReleaseListSizeExt*) and vice-versa. If the network includes *spatialRelationInfoToAddModListExt*, it includes the same number of entries, and listed in the same order, as in the concatenation of *spatialRelationInfoToAddModList* and of *spatialRelationInfoToAddModListSizeExt*. |
| ***spatialRelationInfoToReleaseList, spatialRelationInfoToReleaseListSizeExt, spatialRelationInfoToReleaseListExt***Lists of spatial relation configurations between a reference RS and PUCCH to be released by the UE. |
| ***sps-PUCCH-AN-List***Indicates a list of PUCCH resources for DL SPS HARQ ACK. The field *maxPayloadSize* is absent for the first and the last *SPS-PUCCH-AN* in the list. If configured, this overrides *n1PUCCH-AN* in *SPS-config.* |
| ***subslotLengthForPUCCH***Indicate the sub-slot length for sub-slot based PUCCH feedback in number of symbols (see TS 38.213 [13], clause 9). Value *n2* corresponds to 2 symbols, value *n6* corresponding to 6 symbols, value *n7* corresponds to 7 symbols. For normal CP, the value is either *n2* or *n7*. For extended CP, the value is either *n2* or *n6*. |
| ***ul-AccessConfigListDCI-1-1***List of the combinations of cyclic prefix extension and UL channel access type (See TS 38.212 [17], Clause 7.3.1). |

|  |
| --- |
| ***PUCCH-format3* field descriptions** |
| ***nrofPRBs***The supported values are 1,2,3,4,5,6,8,9,10,12,15 and 16. The UE shall ignore this field when *formatExt* is configured. |

|  |
| --- |
| ***PUCCH-FormatConfig* field descriptions** |
| ***additionalDMRS***If the field is present, the UE enables 2 DMRS symbols per hop of a PUCCH Format 3 or 4 if both hops are more than X symbols when FH is enabled (X=4). And it enables 4 DMRS symbols for a PUCCH Format 3 or 4 with more than 2X+1 symbols when FH is disabled (X=4). The field is not applicable for format 1 and 2. See TS 38.213 [13], clause 9.2.2. |
| ***interslotFrequencyHopping***If the field is present, the UE enables inter-slot frequency hopping when PUCCH Format 1, 3 or 4 is repeated over multiple slots. For long PUCCH over multiple slots, the intra and inter slot frequency hopping cannot be enabled at the same time for a UE. The field is not applicable for format 2. See TS 38.213 [13], clause 9.2.6. |
| ***maxCodeRate***Max coding rate to determine how to feedback UCI on PUCCH for format 2, 3 or 4. The field is not applicable for format 1. See TS 38.213 [13], clause 9.2.5. |
| ***nrofSlots***Number of slots with the same PUCCH F1, F3 or F4. When the field is absent the UE applies the value *n1*. The field is not applicable for format 2. See TS 38.213 [13], clause 9.2.6. |
| ***pi2BPSK***If the field is present, the UE uses pi/2 BPSK for UCI symbols instead of QPSK for PUCCH. The field is not applicable for format 1 and 2. See TS 38.213 [13], clause 9.2.5. |
| ***rb-SetIndex***Indicates the RB set where PUCCH resource is allocated. |
| ***simultaneousHARQ-ACK-CSI***If the field is present, the UE uses simultaneous transmission of CSI and HARQ-ACK feedback with or without SR with PUCCH Format 2, 3 or 4. See TS 38.213 [13], clause 9.2.5. When the field is absent the UE applies the value *off.* The field is not applicable for format 1. |

|  |
| --- |
| ***PUCCH-Resource, PUCCH-ResourceExt* field descriptions** |
| ***format,*** ***formatExt***Selection of the PUCCH format (format 0 – 4) and format-specific parameters, see TS 38.213 [13], clause 9.2. *format0* and *format1* are only allowed for a resource in a first PUCCH resource set. *format2*, *format3* and *format4* are only allowed for a resource in non-first PUCCH resource set. The network can only configure *formatExt* when format is set to *format2* or *format3*. |
| ***interlace0***This is the only interlace of interlaced PUCCH Format 0 and 1 and the first interlace for interlaced PUCCH Format 2 and 3. |
| ***interlace1***A second interlace, in addition to interlace 0, as specified in TS 38.213 [13], clause 9.2.1. For 15KHz SCS, values {0..9} are applicable; for 30Khz SCS, values {0..4} are applicable. For 15kHz SCS, the values of *interlace1* shall satisfy *interlace1*=mod(*interlace0*+X,10) where X=1, -1, or 5. |
| ***intraSlotFrequencyHopping***Enabling intra-slot frequency hopping, applicable for all types of PUCCH formats. For long PUCCH over multiple slots, the intra and inter slot frequency hopping cannot be enabled at the same time for a UE. See TS 38.213 [13], clause 9.2.1. |
| ***occ-Index***Indicates the orthogonal cover code index (see TS 38.213 [13], clause 9.2.1). This field is Applicable when *useInterlacePUCCH-Dedicated-r16* is configured. |
| ***occ-Length***Indicates the orthogonal cover code length (see TS 38.213 [13], clause 9.2.1). Applicable when *useInterlacePUCCH-Dedicated-r16* is configured. |
| ***pucch-ResourceId***Identifier of the PUCCH resource. |
| ***secondHopPRB***Index of first PRB after frequency hopping of PUCCH. This value is applicable for intra-slot frequency hopping (see TS 38.213 [13], clause 9.2.1) or inter-slot frequency hopping (see TS 38.213 [13], clause 9.2.6). |

|  |
| --- |
| ***PUCCH-ResourceSet* field descriptions** |
| ***maxPayloadSize***Maximum number of UCI information bits that the UE may transmit using this PUCCH resource set (see TS 38.213 [13], clause 9.2.1). In a PUCCH occurrence, the UE chooses the first of its *PUCCH-ResourceSet* which supports the number of bits that the UE wants to transmit. The field is absent in the first set (Set0) and in the last configured set since the UE derives the maximum number of UCI information bits as specified in TS 38.213 [13], clause 9.2.1. This field can take integer values that are multiples of 4. |
| ***resourceList***PUCCH resources of *format0* and *format1* are only allowed in the first PUCCH resource set, i.e., in a PUCCH-ResourceSet with *pucch-ResourceSetId* = 0. This set may contain between 1 and 32 resources. PUCCH resources of *format2*, *format3* and *format4* are only allowed in a *PUCCH-ResourceSet* with *pucch-ResourceSetId* > 0. If present, these sets contain between 1 and 8 resources each. The UE chooses a *PUCCH-Resource* from this list as specified in TS 38.213 [13], clause 9.2.3. Note that this list contains only a list of resource IDs. The actual resources are configured in *PUCCH-Config*. |

|  |  |
| --- | --- |
| **Conditional Presence** | **Explanation** |
| *PI2-BPSK* | The field is optionally present, Need R, if *format3* and/or *format4* are configured and *pi2BPSK* is configured in each of them. It is absent, Need R otherwise. |

[…]

– *PUSCH-PowerControl*

The IE *PUSCH-PowerControl* is used to configure UE specific power control parameter for PUSCH.

***PUSCH-PowerControl* information element**

-- ASN1START

-- TAG-PUSCH-POWERCONTROL-START

PUSCH-PowerControl ::= SEQUENCE {

 tpc-Accumulation ENUMERATED { disabled } OPTIONAL, -- Need S

 msg3-Alpha Alpha OPTIONAL, -- Need S

 p0-NominalWithoutGrant INTEGER (-202..24) OPTIONAL, -- Need M

 p0-AlphaSets SEQUENCE (SIZE (1..maxNrofP0-PUSCH-AlphaSets)) OF P0-PUSCH-AlphaSet OPTIONAL, -- Need M

 pathlossReferenceRSToAddModList SEQUENCE (SIZE (1..maxNrofPUSCH-PathlossReferenceRSs)) OF PUSCH-PathlossReferenceRS

 OPTIONAL, -- Need N

 pathlossReferenceRSToReleaseList SEQUENCE (SIZE (1..maxNrofPUSCH-PathlossReferenceRSs)) OF PUSCH-PathlossReferenceRS-Id

 OPTIONAL, -- Need N

 twoPUSCH-PC-AdjustmentStates ENUMERATED {twoStates} OPTIONAL, -- Need S

 deltaMCS ENUMERATED {enabled} OPTIONAL, -- Need S

 sri-PUSCH-MappingToAddModList SEQUENCE (SIZE (1..maxNrofSRI-PUSCH-Mappings)) OF SRI-PUSCH-PowerControl

 OPTIONAL, -- Need N

 sri-PUSCH-MappingToReleaseList SEQUENCE (SIZE (1..maxNrofSRI-PUSCH-Mappings)) OF SRI-PUSCH-PowerControlId

 OPTIONAL -- Need N

}

P0-PUSCH-AlphaSet ::= SEQUENCE {

 p0-PUSCH-AlphaSetId P0-PUSCH-AlphaSetId,

 p0 INTEGER (-16..15) OPTIONAL, -- Need S

 alpha Alpha OPTIONAL -- Need S

}

P0-PUSCH-AlphaSetId ::= INTEGER (0..maxNrofP0-PUSCH-AlphaSets-1)

PUSCH-PathlossReferenceRS ::= SEQUENCE {

 pusch-PathlossReferenceRS-Id PUSCH-PathlossReferenceRS-Id,

 referenceSignal CHOICE {

 ssb-Index SSB-Index,

 csi-RS-Index NZP-CSI-RS-ResourceId

 }

}

PUSCH-PathlossReferenceRS-r16 ::= SEQUENCE {

 pusch-PathlossReferenceRS-Id-r16 PUSCH-PathlossReferenceRS-Id-v1610,

 referenceSignal-r16 CHOICE {

 ssb-Index-r16 SSB-Index,

 csi-RS-Index-r16 NZP-CSI-RS-ResourceId

 }

}

PUSCH-PathlossReferenceRS-Id ::= INTEGER (0..maxNrofPUSCH-PathlossReferenceRSs-1)

PUSCH-PathlossReferenceRS-Id-v1610 ::= INTEGER (maxNrofPUSCH-PathlossReferenceRSs..maxNrofPUSCH-PathlossReferenceRSs-1-r16)

SRI-PUSCH-PowerControl ::= SEQUENCE {

 sri-PUSCH-PowerControlId SRI-PUSCH-PowerControlId,

 sri-PUSCH-PathlossReferenceRS-Id PUSCH-PathlossReferenceRS-Id,

 sri-P0-PUSCH-AlphaSetId P0-PUSCH-AlphaSetId,

 sri-PUSCH-ClosedLoopIndex ENUMERATED { i0, i1 }

}

SRI-PUSCH-PowerControlId ::= INTEGER (0..maxNrofSRI-PUSCH-Mappings-1)

PUSCH-PowerControl-v1610 ::= SEQUENCE {

 pathlossReferenceRSToAddModListSizeExt-r16 SEQUENCE (SIZE (1..maxNrofPUSCH-PathlossReferenceRSsDiff-r16)) OF PUSCH-PathlossReferenceRS-r16

 OPTIONAL, -- Need N

 pathlossReferenceRSToReleaseListSizeExt-r16 SEQUENCE (SIZE (1..maxNrofPUSCH-PathlossReferenceRSsDiff-r16)) OF PUSCH-PathlossReferenceRS-Id-v1610

 OPTIONAL, -- Need N

 p0-PUSCH-SetList-r16 SEQUENCE (SIZE (1..maxNrofSRI-PUSCH-Mappings)) OF P0-PUSCH-Set-r16 OPTIONAL, -- Need R

 olpc-ParameterSet SEQUENCE {

 olpc-ParameterSetDCI-0-1-r16 INTEGER (1..2) OPTIONAL, -- Need R

 olpc-ParameterSetDCI-0-2-r16 INTEGER (1..2) OPTIONAL -- Need R

 } OPTIONAL, -- Need M

 ...

}

P0-PUSCH-Set-r16 ::= SEQUENCE {

 p0-PUSCH-SetId-r16 P0-PUSCH-SetId-r16,

 p0-List-r16 SEQUENCE (SIZE (1..maxNrofP0-PUSCH-Set-r16)) OF P0-PUSCH-r16 OPTIONAL, -- Need R

 ...

}

P0-PUSCH-SetId-r16 ::= INTEGER (0..maxNrofSRI-PUSCH-Mappings-1)

P0-PUSCH-r16 ::= INTEGER (-16..15)

-- TAG-PUSCH-POWERCONTROL-STOP

-- ASN1STOP

|  |
| --- |
| ***P0-PUSCH-AlphaSet* field descriptions** |
| ***alpha***alpha value for PUSCH with grant (except msg3) (see TS 38.213 [13], clause 7.1). When the field is absent the UE applies the value 1. |
| ***p0***P0 value for PUSCH with grant (except msg3) in steps of 1dB (see TS 38.213 [13], clause 7.1). When the field is absent the UE applies the value 0. |

|  |
| --- |
| ***P0-PUSCH-Set* field descriptions** |
| ***p0-List***Configuration of {p0-PUSCH, p0-PUSCH} sets for PUSCH. If SRI is present in the DCI, then one p0-PUSCH can be configured in P0-PUSCH-Set. If SRI is not present in the DCI, and both *olpc-ParameterSetDCI-0-1* and *olpc-ParameterSetDCI-0-2* are configured to be 1 bit, then one p0-PUSCH can be configured in P0-PUSCH-Set. If SRI is not present in the DCI, and if any of *olpc-ParameterSetDCI-0-1* and *olpc-ParameterSetDCI-0-2* is configured to be 2 bits, then two p0-PUSCH values can be configured in P0-PUSCH-Set (see TS 38.213 [13] clause 7 and TS 38.212 [17] clause 7.3.1). |
| ***p0-PUSCH-SetId***Configure the index of a p0-PUSCH-Set (see TS 38.213 [13] clause 7 and TS 38.212 [17] clause 7.3.1). |

|  |
| --- |
| ***PUSCH-PowerControl* field descriptions** |
| ***deltaMCS***Indicates whether to apply delta MCS. When the field is absent, the UE applies Ks = 0 in delta\_TFC formula for PUSCH (see TS 38.213 [13], clause 7.1). |
| ***msg3-Alpha***Dedicated alpha value for msg3 PUSCH (see TS 38.213 [13], clause 7.1). When the field is absent the UE applies the value 1. |
| ***olpc-ParameterSetDCI-0-1, olpc-ParameterSetDCI-0-2***Configures the number of bits for Open-loop power control parameter set indication for DCI format 0\_1/0\_2 in case SRI is not configured in the DCI. 2 bits is applicable only if SRI is not present in the DCI format 0\_1. The field *olpc-ParameterSetDCI-0-1* applies to DCI format 0\_1 and the field *olpc-ParameterSetDCI-0-2* applies to DCI format 0\_2 (see TS 38.212 [17], clause 7.3.1 and TS 38.213 [13], clause 11). |
| ***p0-AlphaSets***configuration {p0-pusch, alpha} sets for PUSCH (except msg3), i.e., { {p0,alpha,index1}, {p0,alpha,index2},...} (see TS 38.213 [13], clause 7.1). When no set is configured, the UE uses the P0-nominal for msg3 PUSCH, P0-UE is set to 0 and alpha is set according to msg3-Alpha configured for msg3 PUSCH. |
| ***p0-NominalWithoutGrant***P0 value for UL grant-free/SPS based PUSCH. Value in dBm. Only even values (step size 2) allowed (see TS 38.213 [13], clause 7.1). |
| ***p0-PUSCH-SetList***Configure one additional *P0-PUSCH-Set* per SRI. If present, the one bit or 2 bits in the DCI is used to dynamically indicate among the P0 value from the existing *P0-PUSCH-AlphaSet* and the P0 value(s) from the *P0-PUSCH-Set* (See TS 38.212 [17], clause 7.3.1 and TS 38.213 [13], clause 17). |
| ***pathlossReferenceRSToAddModList, pathlossReferenceRSToAddModListSizeExt***A set of Reference Signals (e.g. a CSI-RS config or a SS block) to be used for PUSCH path loss estimation. The set consists of Reference Signals configured using *pathLossReferenceRSToAddModList* and *Reference* Signals configured using *pathlossReferenceRSToAddModListSizeExt*.Up to *maxNrofPUSCH-PathlossReferenceRSs* may be configured (see TS 38.213 [13], clause 7.1). The UE shall consider entries in *pathlossReferenceRSToAddModList* and in *pathlossReferenceRSToAddModListSizeExt* as a single list, i.e. an entry created using *pathlossReferenceRSToAddModList* can be modifed using *pathlossReferenceRSToAddModListSizeExt* (or deleted using *pathlossReferenceRSToReleaseListSizeExt*) and vice-versa. |
| ***pathlossReferenceRSToReleaseList, pathlossReferenceRSToReleaseListSizeExt***Lists of reference symbols for PUSCH path loss estimation to be released by the UE. |
| ***sri-PUSCH-MappingToAddModList***A list of *SRI-PUSCH-PowerControl* elements among which one is selected by the SRI field in DCI (see TS 38.213 [13], clause 7.1). |
| ***tpc-Accumulation***If enabled, UE applies TPC commands via accumulation. If not enabled, UE applies the TPC command without accumulation. If the field is absent, TPC accumulation is enabled (see TS 38.213 [13], clause 7.1). |
| ***twoPUSCH-PC-AdjustmentStates***Number of PUSCH power control adjustment states maintained by the UE (i.e., fc(i)). If the field is present (*n2*) the UE maintains two power control states (i.e., fc(i,0) and fc(i,1)). If the field is absent, it maintains one power control state (i.e., fc(i,0)) (see TS 38.213 [13], clause 7.1). |

|  |
| --- |
| ***SRI-PUSCH-PowerControl* field descriptions** |
| ***sri-P0-PUSCH-AlphaSetId***The ID of a *P0-PUSCH-AlphaSet* as configured in *p0-AlphaSets* *in PUSCH-PowerControl*. |
| ***sri-PUSCH-ClosedLoopIndex***The index of the closed power control loop associated with this *SRI-PUSCH-PowerControl.* |
| ***sri-PUSCH-PathlossReferenceRS-Id***The ID of *PUSCH-PathlossReferenceRS* as configured in the *pathlossReferenceRSToAddModList* in *PUSCH-PowerControl*. |
| ***sri-PUSCH-PowerControlId***The ID of this *SRI-PUSCH-PowerControl* configuration. It is used as the codepoint (payload) in the SRI DCI field. |

 […]

A.4.2 Critical extension of messages and fields

The mechanisms to critically extend a message are defined in A.3.3. There are both "outer branch" and "inner branch" mechanisms available. The "outer branch" consists of a CHOICE having the name *criticalExtensions*, with two values, *c1* and *criticalExtensionsFuture*. The *criticalExtensionsFuture* branch consists of an empty SEQUENCE, while the c1 branch contains the "inner branch" mechanism.

The "inner branch" structure is a CHOICE with values of the form "*MessageName-rX-IEs*" (e.g., "*RRCConnectionReconfiguration-r8-IEs*") or "*spareX*", with the spare values having type NULL. The "-rX-IEs" structures contain the *complete* structure of the message IEs for the appropriate release; i.e., the critical extension branch for the Rel-10 version of a message includes all Rel-8 and Rel-9 fields (that are not obviated in the later version), rather than containing only the additional Rel-10 fields.

The following guidelines may be used when deciding which mechanism to introduce for a particular message, i.e. only an 'outer branch', or an 'outer branch' in combination with an 'inner branch' including a certain number of spares:

- For certain messages, e.g. initial uplink messages, messages transmitted on a broadcast channel, critical extension may not be applicable.

- An outer branch may be sufficient for messages not including any fields.

- The number of spares within inner branch should reflect the likelihood that the message will be critically extended in future releases (since each release with a critical extension for the message consumes one of the spare values). The estimation of the critical extension likelihood may be based on the number, size and changeability of the fields included in the message.

- In messages where an inner branch extension mechanism is available, all spare values of the inner branch should be used before any critical extensions are added using the outer branch.

The following example illustrates the use of the critical extension mechanism by showing the ASN.1 of the original and of a later release

-- /example/ ASN1START -- Original release

RRCMessage ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 c1 CHOICE{

 rrcMessage-r8 RRCMessage-r8-IEs,

 spare3 NULL, spare2 NULL, spare1 NULL

 },

 criticalExtensionsFuture SEQUENCE {}

 }

}

-- ASN1STOP

-- /example/ ASN1START -- Later release

RRCMessage ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 c1 CHOICE{

 rrcMessage-r8 RRCMessage-r8-IEs,

 rrcMessage-r10 RRCMessage-r10-IEs,

 rrcMessage-r11 RRCMessage-r11-IEs,

 rrcMessage-r14 RRCMessage-r14-IEs

 },

 later CHOICE {

 c2 CHOICE{

 rrcMessage-r16 RRCMessage-r16-IEs,

 spare7 NULL, spare6 NULL, spare5 NULL, spare4 NULL,

 spare3 NULL, spare2 NULL, spare1 NULL

 },

 criticalExtensionsFuture SEQUENCE {}

 }

 }

}

-- ASN1STOP

It is important to note that critical extensions may also be used at the level of individual fields i.e. a field may be replaced by a critically extended version. When sending the extended version, the original version may also be included (e.g. original field is mandatory, E-UTRAN is unaware if UE supports the extended version). In such cases, a UE supporting both versions may be required to ignore the original field. The following example illustrates the use of the critical extension mechanism by showing the ASN.1 of the original and of a later release.

-- /example/ ASN1START -- Original release

RRCMessage ::= SEQUENCE {

 rrc-TransactionIdentifier RRC-TransactionIdentifier,

 criticalExtensions CHOICE {

 c1 CHOICE{

 rrcMessage-r8 RRCMessage-r8-IEs,

 spare3 NULL, spare2 NULL, spare1 NULL

 },

 criticalExtensionsFuture SEQUENCE {}

 }

}

RRCMessage-rN-IEs ::= SEQUENCE {

 field1-rN ENUMERATED {

 value1, value2, value3, value4} OPTIONAL, -- Need N

 field2-rN InformationElement2-rN OPTIONAL, -- Need N

 nonCriticalExtension RRCConnectionReconfiguration-vMxy-IEs OPTIONAL

}

RRCConnectionReconfiguration-vMxy-IEs ::= SEQUENCE {

 field2-rM InformationElement2-rM OPTIONAL, -- Cond NoField2rN

 nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- ASN1STOP

| **Conditional presence** | **Explanation** |
| --- | --- |
| *NoField2rN* | The field is optionally present, need N, if field2-rN is absent. Otherwise the field is absent |

Finally, it is noted that a critical extension may be introduced in the same release as the one in which the original field was introduced e.g. to correct an essential ASN.1 error. In such cases a UE capability may be introduced, to assist the network in deciding whether or not to use the critical extension.

In the case of list fields (SEQUENCE OF types in ASN.1) using the ToAddMod/ToRelease construction, the use of critical extensions to increase the size of a list should be avoided; that is, extensions done according to the following example should be avoided:

-- /example/ ASN1START -- Discouraged example

ContainingStructure ::= SEQUENCE {

 listElementToAddModList SEQUENCE (SIZE (1..maxNrofListElements)) OF ListElement OPTIONAL, -- Need N

 ...,

 [[

 listElementToAddModList-rN SEQUENCE (SIZE (1..maxNrofListElements-rN)) OF ListElement OPTIONAL -- Need N

 ]]

}

-- ASN1STOP

Instead, a non-critical list extension mechanism should typically be used, such that the extension field only adds the new entries of the list. This approach is further described in subclause A.4.3.x.

If the critical extension mechanism for a list is used, it should be clarified in the field description that the two versions of the list are not configured together, and that the network should release the contents of the original version when configuring the replacement version.

A.4.3 Non-critical extension of messages

[…]

A.4.3.x Non-critical extensions of lists with ToAddMod/ToRelease

When the size of a list using the ToAddMod/ToRelease construction is extended and/or fields are added to the list element structure, the list should be non-critically extended in accordance with the following general principles:

– When only the size of the list is extended, this is reflected in a non-critical extension of the list, with a "SizeExt" suffix added to the end of the field name (before any -rN suffix). The differential size of the extended list uses the suffix "Diff". A new ToRelease list is needed, and its range should generally include the total number of entries to allow the new ToRelease list also to release the original entries. In many cases, extending the list size will also require an extended list element ID type to account for the increased size of the list; in these cases the element type will need to be extended to include the extended element ID, resulting in a more complex extension similar to example 3 below. The field description table should indicate that the UE considers the original list and the extension list as a single list; thus entries added with the original list can be modified by the extension list (or removed by the extension of the ToRelease list), or vice versa. The result is as shown in the following example:

-- /example 1/ ASN1START

ContainingStructure ::= SEQUENCE {

 listElementToAddModList SEQUENCE (SIZE (1..maxNrofListElements)) OF ListElement OPTIONAL, -- Need N

 listElementToReleaseList SEQUENCE (SIZE (1..maxNrofListElements)) OF ListElementId OPTIONAL, -- Need N

 ...,

 [[

 -- Non-critical extension lists

 listElementToAddModListSizeExt-rN SEQUENCE (SIZE (1..maxNrofListElementsDiff-rN)) OF ListElement OPTIONAL, -- Need N

 listElementToReleaseListSizeExt-rN SEQUENCE (SIZE (1..maxNrofListElementsDiff-rN)) OF ListElementId OPTIONAL -- Need N

 ]]

}

-- ASN1STOP

– When fields are added to the list element structure, an extension marker should normally be used if available. If no extension marker is available or if overhead or other considerations prevent using the extension marker, an extension structure should be created for the new fields, with the suffix "Ext" added to the end of the field name and the element structure type name (before the -rN suffix), and a parallel ToAddMod list introduced to hold the new structures, also with the "Ext" suffix. The field description table should indicate that the parallel list contains the same number of entries, and in the same order, as the original list. No new ToRelease list is typically needed (unless the list element ID type changes). It should typically be ensured that the contained fields in the "Ext" elements are releaseable without release and add of the entire list element; this can, for instance, be ensured by having the new fields be OPTIONAL Need R. The result is as shown in the following example:

-- /example 2/ ASN1START

ContainingStructure ::= SEQUENCE {

 listElementToAddModList SEQUENCE (SIZE (1..maxNrofListElements)) OF ListElement OPTIONAL, -- Need N

 listElementToReleaseList SEQUENCE (SIZE (1..maxNrofListElements)) OF ListElementId OPTIONAL, -- Need N

 ...,

 [[

 -- Parallel list

 listElementToAddModListExt-rN SEQUENCE (SIZE (1..maxNrofListElements)) OF ListElementExt-rN OPTIONAL -- Need N

 ]]

}

ListElement ::= SEQUENCE {

 elementId ListElementId,

 field1 INTEGER (0..3),

 field2 ENUMERATED { value1, value2, value3 }

}

ListElementExt-rN ::= SEQUENCE {

 field3-rN BIT STRING (SIZE (8)) OPTIONAL -- Need R

}

-- ASN1STOP

– When the size of a list is extended and fields are added to the list element structure, an extension marker should normally be used for the added fields if available, and the list extended with the non-critical mechanism as described in the first example above*.* If no extension marker is available or if overhead or other considerations prevent using the extension marker, an extension structure should be created for the new fields and a parallel list with ToAddMod introduced to hold the extension structures, as in the second example above, for entries of the original list and for entries of the extension list holding new entries. The field description table should indicate that the parallel list contains the same number of entries, and in the same order, as the concatenation of the original list and the extension list. An extended ToRelease list is needed, but no additional parallel ToRelease list is needed (i.e. there is no *listElementToReleaseListExt-rN* in the example below), as the original and extended ToRelease lists suffice to release any element of the combined list. The extended element ID type should be captured as a non-critical extension of the original element ID type, with the field description indicating that if the extended ID is present, the original ID is ignored. The result is as shown in the following example:

-- /example 3/ ASN1START

ContainingStructure ::= SEQUENCE {

 listElementToAddModList SEQUENCE (SIZE (1..maxNrofListElements)) OF ListElement OPTIONAL, -- Need N

 listElementToReleaseList SEQUENCE (SIZE (1..maxNrofListElements)) OF ListElementId OPTIONAL, -- Need N

 ...,

 [[

 -- Non-critical extension lists

 listElementToAddModListSizeExt-rN SEQUENCE (SIZE (1..maxNrofListElementsDiff-rN)) OF ListElement OPTIONAL, -- Need N

 listElementToReleaseListSizeExt-rN SEQUENCE (SIZE (1..maxNrofListElementsDiff-rN)) OF ListElementId-vNxy OPTIONAL, -- Need N

 -- Parallel list with maxNrofListElements-rN = maxNrofListElements + maxNrofListElementsDiff

 listElementToAddModListExt-rN SEQUENCE (SIZE (1..maxNrofListElements-rN)) OF ListElementExt-rN OPTIONAL, -- Need N

 ]]

}

ListElement ::= SEQUENCE {

 elementId ListElementId,

 field1 INTEGER (0..3),

 field2 ENUMERATED { value1, value2, value3 }

}

ListElementExt-rN ::= SEQUENCE {

 -- Field description should indicate that if the elementId-vNxy is present, the elementId (without suffix) is ignored

 elementId-vNxy ListElementId-vNxy OPTIONAL, -- Need S

 field3-rN BIT STRING (SIZE (8)) OPTIONAL -- Need R

}

ListElementId ::= INTEGER (0..maxNrofListElements-1)

ListElementId-rN ::= INTEGER (0..maxNrofListElements-rN-1)

ListElementId-vNxy ::= INTEGER (maxNrofListElements..maxNrofListElements-rN-1)

-- ASN1STOP