3GPP TSG-RAN WG2 #116 electronic R2-210xxxx

Electronic meeting, November 01 – 12, 2021

Agenda Item: 8.0

Source: Ericsson

Title: Summary of offline 048 Rel-17 RRC SetModifyRelease

Document for: Discussion, Decision

# 1 Introduction

This contribution summarizes the following discussion:

* [AT116-e][048][NR17] RRC SetModifyRelease (Ericsson)

Scope: Review R2-2110778, R2-2110779, collect comments.

Intended outcome: Report

Deadline: EOM

Companies are invited to fill in contact details.

|  |  |
| --- | --- |
| **Company** | **Contact details** |
| MediaTek | Nathan Tenny ([nathan.tenny@mediatek.com](mailto:nathan.tenny@mediatek.com)) |
| Intel | Sudeep Palat (sudeep.k.palat@intel.com) |
| Huawei, HiSilicon | David Lecompte (david.lecompte@huawei.com) |
| OPPO | duzhongda@oppo.com |
| Nokia | amaanat.ali@nokia.com |
|  |  |
|  |  |
|  |  |

# 2 Discussion

The paper in [1] discusses the use of SetModifyRelease structure for Rel-17, to overcome the current limitations on SetupRelease structure, especially when it comes to its use for delta configuration, as explained in the paper. To further illustrate this issue some general aspects are outlined below.

Considering a source RRCConfiguration and a target RRCConfiguration, one needs to know whether the latter can be signaled as a delta from the former. Each configuration is a tree of fields. For each of those fields RRC specifies whether it is mandatory or optional; and if optional whether it is M, N, R or S.

To determine the delta between source and target RRCRecofiguration, one has to traverse both trees of configurations in parallel:

1. Fields which are mandatory must always be kept in the target tree.
2. Fields which are optional Need R and present in the target tree must be kept.
3. Fields which are optional Need M, and
   1. present in the target tree can be removed from the target tree if the source tree contains the same element with the same value.
   2. present in the target tree must be kept if the value is different from the value in the source tree
   3. absent in the target tree but present in the source tree
      1. must be set to Release (if they are a SetupRelease Structure)
      2. else, require releasing the parent field (e.g. by absence if the parent field is Need R)

The problem occurs if case 3)a and case 3)c-ii occur in the same IE. The IE and the affected value must be present due to 3) but must be absent due to 3)c-ii. This may imply on the need to e.g. send a first message where the field is set to "release" and a subsequent message where it is set to "setup" again, which may anyway not be possible for mutiple IEs (e.g. the UE would not be able to have pdcch-Config simply released while no configuration is added in the same message).

**Q1 Do companies have any comments on the aspects above?**

|  |  |
| --- | --- |
| **Company** | **Comments** |
| MediaTek | We agree with the general principle that there are cases where a release-and-setup is necessary to make specific changes in the UE’s configuration. The RAN2 intention should be to avoid requiring this, but sometimes cases slip through.  However, this is not a new problem, and in the past we haven’t addressed it by introducing a „release/delta/fullConfig“ structure in the ASN.1; we could always do so, but it would be discussed when needed for a specific field. Do we have Rel-17 cases where it seems necessary? |
| Intel | During previous discussions, RAN2 had encountered this problem where a release and addition of a configuration was needed together and it was not possible to do it in one message. So we have some sympathy on the issue.  However, from our recollection, the issue we had in the past was related to absence of a “release” mechanism for some sub-fields that we had in Rel-15 such that a release of the parent field at top level was needed.  Since then, in later releases, we have ensured that all fields can be released. Then, the benefit of this release and add of a field is unclear as all the subfields can be released and delta configuration is possible – so there shouldn’t be a need to release the top level field and add it again. It would be good to have some concrete use case where one has to/useful to do “local full configuration”?  Our other concern is that with this structure defined, there is a risk of wide spread “abuse” of local full configuration instead of using the delta configuraiton.  Our feeling is that the current structure should meet most requirements and this usage should be an exception (there is already a lot of concern about the RRCReconfiguration message size).  Another option for any exceptional cases is to have a release field in the ElementTypeParam IE itself – that would first release the current configuration and then apply everything that follows (one example were we used this release and setup was, I think, for SCG configuration).  This kind of approach could be used on an as needed basis with due consideration and minimise the risk of extensive use of “local full configuration”. |
| Huawei, HiSilicon | This problem may exist but if the parent or any of its ancestors is an element of a ToAddModList and release and add in the same message is supported, the problem can be solved that way.  Due to the large presence of ToAddModList structures, the highlighted problem may not be common. |
| OPPO | Logically a local full configuration is still possible assuming network just want to updating configured parameter or configure new parameters instead of releasing a parameter which is optional need M and unfortunatly located in the same IE with parameter(s) suppose to continue. But the question is why such parameter is designed as optional need M from 1st place? It looks like proponent try to say 3Cii could exists in the spec during standardizatino, but later on in field network vendors find a local full configuration is useful? Maybe proponents can clarify the issue at first.  In general we don’t see strong motivation to introduce the new structure. In addition it is very difficult to define how “local“ the full configuration could be i.e. to which level of the paramter tree we can call such behaviour is local full configuration to avoid any impact on user plane? |
| Nokia | We are on the same page as MediaTek but are definitely not averse to discuss with with a clear example where all of us can see the existing structure does not work. Probably rapporteur can suggest a long email discussion until next meeting and we can discuss some examples in that discussion and how the new structure helps. |
|  |  |
|  |  |
|  |  |

To address this issue above, [1] discusses a SetModifyRelease structure for Rel-17, defined as:

SetModifyRelease { ElementTypeParam } ::= CHOICE {

release NULL,

set ElementTypeParam // sets or replaces the previous values

delta ElementTypeParam // a delta to a previous configuration

}

The possible options and comparison with the current SetupRelease structure can also be summarized as below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SetupRelease | | |  | SetModifyRelease | | | |
| Contained field is… | Config branch | |  | Contained field is… | Config branch | | |
| Setup | Release |  | Set | Modify | Release |
| Not configured | Full config | No action |  | Not configured | Full config | N/A | No action |
| Configured | Delta config | Release |  | Configured | Full config | Delta config | Release |

Note that essentially the behavior for the SetModifyRelease structure is alreadu present for SCG configuration, where we have a SetupRelease structure and *mrdc-ReleaseAndAdd* flag one level up. Essentially, an IE defined with SetModifyRelease allows to further perform a full configuration of this IE even when this IE is already configured.

**Q2 Do companies have any comments on the structure?**

|  |  |
| --- | --- |
| **Company** | **Comments** |
| MediaTek | If this tripartite structure is an idiom that we start using in ASN.1, it makes sense to capture as a parametrised type. It seems backward to introduce the type first for a pattern that we don’t actually use; the SetupRelease idiom was common throughout LTE already, and the point of introducing the parametrised type was to make the code more readable.  We are open to discussing whether there are cases where a structure like this would be needed, but let’s be driven by concrete examples. |
| Intel | While the proposed new structure seems logical, given that we have been using the current structure, our slight preference is to re-use the current structure as much as possible – that would be to have “set” and “delta” in the new structure continue to use the “setup” of the current structure (which is already used for setup and delta) – that is, there is no change in the setup branch.  And the new branch to be a “release and setup”. |
| Huawei, HiSilicon | Agree with MediaTek that a new paremetrised type makes more sense for something that is already used.  We could try to build on an existing case, e.g. *mrdc-ReleaseAndAdd*. |
| OPPO | Intend to agree with Intel regarding the parameter name in the strucutre. |
| Nokia | Yes, generally we are aligned to what companies have been saying above. |
|  |  |
|  |  |
|  |  |

A draft CR was also submitted to capture this in 38.331 [2].

**Q3 Do companies have any comments on the draft CR?**

|  |  |
| --- | --- |
| **Company** | **Comments** |
| MediaTek | 1. Some text is needed also in section 5.1.2, to clarify the general handling of the new *delta* branch, as well as the *set* branch, since the procedural text today only describes *setup* and *release*. 2. We are kind of inconsistent today about using explicit procedural code for fields that use SetupRelease (for instance, gapFR1 and gapFR2 have procedural text saying what to do if the field is set to *setup*, but pdsch-Config and pdcch-Config don’t). For SetModifyRelease this could be more important, because of the different handling of absent fields in the *set* and *delta* branches (basically the *set* branch treats all need codes as Need R), and it might be good to have some clarification of when procedural text is considered necessary. 3. The actual definition of the parametrised type in 6.3.0 is missing from the CR. 4. There is a bug in the definitions of RRCMessage-rX-IEs and Element-rX in the example: The whole IEs are flagged as OPTIONAL, which is incorrect syntax. This seems to be inherited from the SetupRelease guidelines which have the same mistake, so it would be good to fix it there as well. 5. As a small detail, it seems like the name of the type should match the names of the branches: either SetDeltaRelease, or change the name of the *delta* branch to *modify*. |
| Huawei, HiSilicon | We agree with all the comments from MediaTek.  Text for 5.1.2 would need to be verified as accurate against several examples. |
| Nokia | Yes, generally we are aligned to what companies have been saying above. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 3 Conclusion

- To be updated after discussion -

# 4 References

1. R2-2110778 Set Modify Release structure, Ericsson, RAN2 #116-e, November 01 – 12, 2021
2. R2-2110779 Draft CR for Setup Modify Release structure (38.331), Ericsson, RAN2 #116-e, November 01 – 12, 2021