3GPP TSG-RAN WG2 #115-e R2-21xxxxx

Online, 16-27 August 2021

Agenda Item: xx

Source: MediaTek Inc.

**Title: Report of email discussion [Post114-e][071][NR16] CandidateBeamRSList set to release (MediaTek)**

Document for: Discussion, decision

# 1 Introduction

This document is a report on the following email discussion, initiated after RAN2#114-e:

* [Post114-e][071][NR16] CandidateBeamRSList set to release (MediaTek)

Scope: how UE shall handle the extension field of candidateBeamRSList. The intention is to agree a 38.331 clarification CR in next meeting. Could consider option 2 and option 3 proposed in R2-2106115 as a starting point. This was also discussed in [AT114-e][022].

Intended outcome: Report, agreeable CR.

Deadline: Long

The discussion will proceed in two phases, first to determine an agreeable mechanism for handling the extension field and second to converge on an agreeable CR. The deadlines are as follows:

Phase 1: Friday 2 July 1700 UTC

Phase 2: Friday 6 August 0900 UTC

# 2 Contact Points

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

|  |  |  |
| --- | --- | --- |
| Company | Name | Email Address |
| MediaTek (rapporteur) | Nathan Tenny | nathan.tenny@mediatek.com |
| ZTE | Liu yu | liu.yu3@zte.com.cn |
| Intel | Sudeep Palat | Sudeep.k.palat@intel.com |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 3 Background

## 3.1 Original options

The discussion from [1] considered three options drawn from the discussion in [2]:

* **Option 1:** The UE releases the entire concatenated list, both the entries configured with *candidateBeamRSList* and the entries configured with *candidateBeamRSListExt-v1610*.
* **Option 2:** The UE releases only the extended entries that were configured with *candidateBeamRSListExt-v1610*.
* **Option 3:** The *release* branch is not used, and the UE treats *candidateBeamRSList* and *candidateBeamRSListExt-v1610* as a single concatenated field with Need M. The extended list *candidateBeamRSListExt-v1610* is only included when *candidateBeamRSList* is included and fully populated.

## 3.2 Updated options for this discussion

In the discussion, option 1 had less support compared to options 2 and 3. Rapporteur also understands that in continued offline discussion (separate from the official email discussion), a network-based restriction was proposed, in which the network is required to signal the extension (*candidateBeamRSListExt-v1610*) whenever it wants the extension entries to remain unchanged in the UE, and the “release” option on the extension list is used only when the network intends to reconfigure the UE to a number of entries fitting within the original list. This option (option C below) disambiguates the UE behaviour by having the network always indicate explicitly the fate of the extension entries.

Accordingly, this discussion considers three options:

* **Option A:** When *candidateBeamRSListExt-v1610* is set to *release*, the UE releases only the extended entries that were configured with *candidateBeamRSListExt-v1610*.
* **Option B:** The *release* branch is not used, and the UE treats *candidateBeamRSList* and *candidateBeamRSListExt-v1610* as a single concatenated field with Need M. The extended list *candidateBeamRSListExt-v1610* is only included when *candidateBeamRSList* is included and fully populated.
* **Option C:** The network is required to signal the extension (*candidateBeamRSListExt-v1610*) whenever it wants the extension entries to remain unchanged in the UE, and the *release* option on the extension list is used only when the network intends to reconfigure the UE to a number of entries fitting within the original list.

The details of the options may require some clarification (e.g. how to define the “extended entries” in option A), so the following discussion subsections include space for discussion of the details of each option.

## 3.3 Examples

For clarity, this section illustrates how options A/B/C would operate in two example scenarios.

**Example 1:** The network reduces the list size while extension entries are configured, and the resulting list is still larger than the legacy list size:

1. Network sends a *BeamFailureRecoveryConfig* containing a fully populated *candidateBeamRSList* (16 entries) and a partly populated *candidateBeamRSListExt-v1610* (2 entries).



1. UE concatenates the fields into a single list of 18 entries.



1. Network sends a *BeamFailureRecoveryConfig* containing a partly populated *candidateBeamRSList* (15 entries) and omitting the *candidateBeamRSListExt-v1610*.
   1. With option A, the UE populates a list of 17 entries, and the handling of a future *release* indication depends on the interpretation of the option (see section 4.1 below).



* 1. With option B, the UE populates a list of 15 entries; the 2 entries from *candidateBeamRSListExt-v1610* are released.



* 1. With option C, this step is not allowed; if the network intends to reconfigure the UE to a list of 15 entries, it needs to include the 15 entries explicitly along with the *candidateBeamRSListExt-v1610* set to *release*, and if it intends to reconfigure the UE to a list of 17 entries, it needs to include the 17 entries explicitly.

**Example 2:** The network reduces the list size while extension entries are configured, and the resulting list fits inside the legacy list size:

1. Network sends a *BeamFailureRecoveryConfig* containing a fully populated *candidateBeamRSList* (16 entries) and a partly populated *candidateBeamRSListExt-v1610* (2 entries).



1. UE concatenates the fields into a single list of 18 entries.



1. Network sends a *BeamFailureRecoveryConfig* containing a partly populated *candidateBeamRSList* (10 entries) and omitting the *candidateBeamRSListExt-v1610*.
   1. With option A, the UE populates a list of 12 entries, and the handling of a future *release* indication depends on the interpretation of the option (see section 4.1 below).



* 1. With option B, the UE populates a list of 10 entries; the 2 entries from *candidateBeamRSListExt-v1610* are released.



* 1. With option C, this step is not allowed; if the network intends to reconfigure the UE to a list of either r10 or 12 entries, it needs to include the entries explicitly along with the *candidateBeamRSListExt-v1610* set to *release*.

# 4 Discussion (Phase 1)

## 4.1 Details of option A

During the previous discussion of option A, it became clear that there are two potential understandings of the definition of “only the extended entries”.

* Approach A.1: The UE remembers which list entries were *initially* configured by *candidateBeamRSListExt-v1610*, and subsequently treats these as being the extension entries. Thus, even if the list is later shortened to a length that fits within the original list size, some entries may be marked as “extended entries” and can be released with the extension field.
  + In Example 1 from section 3.3, sending a *release* would release the 2 entries that were initially configured by *candidateBeamRSListExt-v1610*, leaving the UE with a list of 15 entries.



* + In Example 2 from section 3.3, sending a *release* would release the 2 entries that were initially configured by *candidateBeamRSListExt-v1610*, leaving the UE with a list of 10 entries.



* Approach A.2: The UE treats the entries from the two list fields as a single undifferentiated list (as usual for lists without ToAddMod structure), and the extension field only addresses entries beyond the size of the original list. Thus, if the list is shortened to a length that fits within the original list size, the UE considers that it has no more extended entries, and setting *candidateBeamRSListExt-v1610* to *release* becomes vacuous.
  + In Example 1 from section 3.3, sending a *release* would release the 1 entry that exceeds the legacy list size, leaving the UE with a list of 16 entries.



* + In Example 2 from section 3.3, sending a *release* would have no effect.



**Question 1.1:** Which of the two approaches do companies prefer, within the scope of option A?

|  |  |  |
| --- | --- | --- |
| **Company** | **Preferred Approach** | **Comments** |
| MediaTek | A.1 | The extension field is Need M, which means that if it is omitted rather than set to *release*, it should be considered to maintain its previous contents. This suggests that we should view *candidateBeamRSListExt-v1610* as a field separate from *candidateBeamRSList*, and the UE should be aware of which entries were configured with the extended list (so that it knows what to maintain in case the field is omitted). We don’t find this to be a problem for the implementation to remember, |
| ZTE | A.1 | If UE and NW store the R15/R16 list separately, NW can reconfigure the R15 list and R16 list separately, and this can save the signaling length effectively. |
| Intel | A2 | This is based on our understanding of A1 and A2 as given in response to Q1.2. |
| CATT | A.1 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Question 1.2:** Any other comment on the details of option A?

|  |  |
| --- | --- |
| **Company** | **Comments** |
| Intel | I tried to summarise my understanding of A1 and A2 but some of it is still not clear to me (shown with ?).  Initial condition: # of entries >16   |  |  |  | | --- | --- | --- | |  | Option A1 | Option A2 | | Legacy list without extension list | Replaces only the elements that was previously signalled by original list? | Replaces only the original list | | Extension list without legacy list | Replaces entries signalled previously by ext list? | Replaces entries >16? | | Ext list with release | Release entries that were previously signalled by extension list | Release entries >16 | | Original+ext list (ext list configures new elements) | Replaces entries previously signalled by the original list and entries signalled by the ext list | Replaces both lists  (conf of ext list allowed if signalled original list is more than 16?) | | Original+ext list (ext list set to release) | Replaces entries previously signalled by the original list and releases entries signalled by the ext list? | Replaces entries <16 and releases entries >16 ? |   Initial condition: # of entries <=16 (may have been signalled as original or ext list)   |  |  |  | | --- | --- | --- | |  | Option A1 | Option A2 | | Legacy list without extension list | Replaces the entire list? Including entries previously signalled by ext list? | Replaces the entries signalled by original list? | | Extension list without legacy list | Replaces entries previously signalled by ext list? | N/A | | Ext list with release | Release entries that were previously signalled by extension list | N/A | | Original+ext list (ext list configures new elements) | Replaces both lists (is it allowed if original list is less than 16?) | Replaces both lists (allowed if original list is more than 16?) | | Original+ext list (ext list set to release) | Replaces entries previously signalled by the original list and releases entries signalled by the ext list? | Replaces entries <16 and releases entries >16 ? | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 4.2 Details of option B

Rapporteur understands that option B is fairly straightforward and there may not be many details that need clarification. This section is provided for any comments on the details of option B.

**Question 2.1:** Any comment on the details of option B?

|  |  |
| --- | --- |
| **Company** | **Comments** |
| ZTE | We think the release branch(i.e. release in SetupRelease) can be used in option B, and the need code of *candidateBeamRSListExt-v1610* can be ‘Need M’.  For option B ,we think the key concerns are:   1. The UE treats *candidateBeamRSList* and *candidateBeamRSListExt-v1610* as a single concatenated list;   2) The first 16 entries are configured by *candidateBeamRSList*, and the entries over 16 are configured by candidateBeamRSListExt-v1610;  3) The need codes of *candidateBeamRSList* and *candidateBeamRSListExt-v1610* all are ‘Need M’;  4) If NW wants to release the entries over 16, NW uses release branch of *candidateBeamRSListExt-v1610*.  **For** **example,**  Step1 (NW wants to add 16 entries):  NW only includes 16 entries by *candidateBeamRSList*;  Step2(NW wants to add 6 more entries, and maintains the first 16 entries unchanged):  NW only includes 6 entries by *candidateBeamRSListExt-v1610*(setup), does not include *candidateBeamRSList*( because the field is ‘Need M’, when the UE receives the second message, the UE stores the new 6 entries concatenated to the first 16 entries. After processing the second message, the UE has 22 entries. )  Step3(NW only wants to modify any entry from the first 16 entries):  Because the later 6 entries are unchanged, NW only includes the first 16 entries by *candidateBeamRSList*;  Step4(NW only wants to modify any entry from the later 6 entries):  Because the first 16 entries are unchanged, NW only includes the later 6 entries by *candidateBeamRSListExt-v1610*(setup);  Step5(NW wants to delete any 10 entries):  Because there are 12 remaining entries after 10 entries are deleted, NW should include the 12 remaining entries by *candidateBeamRSList*, and simultaneously include release command by *candidateBeamRSListExt-v1610*(release). After processing this message, the UE stores the 12 remaining entries. |
| Intel | The main “motivation” of option B is that it follows the principle we had previously agreed that non-AddMod lists are always replaced and there is no delta configuration of partial replacement or release of the elements. It is also simple as there is only one list in the UE and network and behaviour is common for the whole list. There is no release mechanism but then there was no release of the original list anyway.  I tried to summarise my understanding of option B:  Initial condition: # of entries >16   |  |  | | --- | --- | |  | Option B | | Legacy list without extension list | Replaces the entire list | | Extension list without legacy list | Replaces entries >16 | | Ext list with release | N/A | | Original+ext list (ext list configures new elements) | Replaces both lists  (conf of ext list allowed if signalled original list is more than 16) | | Original+ext list (ext list set to release) | N/A |   Initial condition: # of entries <=16 (may have been signalled as original or ext list)   |  |  | | --- | --- | |  | Option B | | Legacy list without extension list | Replaces the entire list | | Extension list without legacy list | N/A | | Ext list with release | N/A | | Original+ext list (ext list configures new elements) | Replaces both lists  (conf of ext list allowed if signalled original list is more than 16) | | Original+ext list (ext list set to release) | N/A | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 4.3 Details of option C

Option C was introduced in informal discussion and may need some analysis to make sure that all the implications are understood. Rapporteur understanding is that this option is intended to have no UE impact (i.e., to be compatible with any UE handling of the *release* branch), since the *release* branch is only used when the original field is populated and the list fits within the original entries.

**Question 3.1:** Companies are invited to provide details of their understanding of option C**.**

|  |  |
| --- | --- |
| **Company** | **Comments** |
| MediaTek | Our understanding is described in the paragraph above; this option should be possible to specify without UE impact. As in the examples from section 3.3, we think it successfully disambiguates the handling of the extended entries. |
| ZTE | For option C, when NW only wants to modify R16 list, whether NW needs to include R15 list or not? (i.e. Do the UE and NW need to store the R15 list and R16 list separately? If yes, we think option A.1 is better than option C)  In addition, the behavior that ‘the network is required to signal the extension (*candidateBeamRSListExt-v1610*) whenever it wants the extension entries to remain unchanged in the UE’ conflicts with the need code ‘Need M’ of the field *candidateBeamRSListExt-v1610*. |
| Intel | The description of C in the introduction is not fully self contained in our view. We think option C is similar to option B in that a common list is maintained in the network and UE except that the release is still applicable for option C and it avoids the signalling scenarios which could avoid some ambiguous scenarios based on network restrictions.  I tried to summarise my understanding of option C:  Initial condition: # of entries >16   |  |  | | --- | --- | |  | Option C | | Legacy list without extension list | N/A | | Extension list without legacy list | N/A | | Ext list with release | Release entries >16 | | Original+ext list (ext list configures new elements) | Replaces both lists (conf of ext list allowed if signalled original list is more than 16) | | Original+ext list (ext list set to release) | Replaces entries <16 and releases entries >16 |   Initial condition: # of entries <=16 (may have been signalled as original or ext list)   |  |  | | --- | --- | |  | Option C | | Legacy list without extension list | Replaces the entire list | | Extension list without legacy list | N/A | | Ext list with release | N/A | | Original+ext list (ext list configures new elements) | Replaces both lists (conf of ext list allowed if signalled original list is more than 16) | | Original+ext list (ext list set to release) | Replaces entries <16 and releases entries >16 | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Question 3.2:** Do companies understand that option C can be specified/implemented without UE impact?

|  |  |  |
| --- | --- | --- |
| **Company** | **Y/N** | **Comments** |
| MediaTek | Y |  |
| Intel | Most likely | It depends on exact definition of C. Based on our understanding of C as captured above, it is quite likely that it is aligned with any possible UE implementation. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Question 3.3:** Any other comment on the details of option C?

|  |  |
| --- | --- |
| **Company** | **Comments** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 4.4 Preferred option

Companies are invited to indicate their preferred option (A/B/C).

**Question 4.1:** Which option do companies prefer among options A/B/C?

|  |  |  |
| --- | --- | --- |
| **Company** | **Preferred Option** | **Comments** |
| MediaTek | A or C | We have some concern about option B, as it effectively changes the need code of *candidateBeamRSListExt-v1610* from Need M to Need R. We are fine with the other two options. |
| ZTE | prefer A.1, then B | We prefer option A.1. But if companies think option A.1 has NBC issue, option B is ok to us.  @ MediaTek: For option B, we think it is unnecessary to change the need code of *candidateBeamRSListExt-v1610* from Need M to Need R. We can use release branch also. Please see our comments in 4.2. |
| Intel | B, C or combination of B and C (see comments) | We particularly don’t like option A1 as it seems quite complex for the UE to remember which list an entry was signalled in. We also have several open points with regard to option A1 which is not clear to us as identified in the table in Q1.2.  We support UE has a single list containing entries signalled by both lists without any additional differentiation.  Option B and C are similar in our understanding in terms of how the list is maintained by the UE. Option B does not use Release branch, while option C restricts signalling just the ext. A combination of B and C could also be considered.  A2 is not entirely clear to us (as indicated in the table on A2 behaviour) – if it is clarified, it may also be acceptable. |
| CATT | A.1 | From our point of view A.1 is reasonable and simpler. The ext list introduced in R16 was so that if it released the UE only keeps the R15 list. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 4.5 Text for the general case

The difficulty of this example seems to suggest that we should have some general guidance in the spec for extending a list without ToAddMod, e.g. in a new section A.4.3.7 or by expanding on the existing example in section A.3.10. Any general text to be captured will depend on what approach we take to solving the specific example, but companies are invited to provide candidate text or general guidance for discussion.

**Question 5.1:** What guidance should we provide for the general case of extending lists without ToAddMod?

|  |  |
| --- | --- |
| **Company** | **Comments** |
| MediaTek | This ASN.1 idiom is definitely unfortunate, but it may not always be avoidable. Basically, we think the agreed-upon option should be documented as a general practice for the case that such an extension is necessary. A couple of principles that should be captured:   * If option A is selected, we should document that the UE needs to remember which entries were configured by the extension field, as this is a departure from the usual extension practice of considering the fields as a single combined list. * If option B is selected, we should make the extension field Need R instead of Need M. Some clarification is still needed about which entries are released, but the combination of Need M with “release when absent” behaviour is confusing. * If option C is selected, the field description should clarify the network behaviour—we shouldn’t rely only on the general guidance to specify what is expected, as it’s easy for general principles like this to be overlooked in specific cases. * In general, we don’t see that any of these solutions are really in conflict with the principle that lists without ToAddMod are always replaced when signalled. The base and extension lists are separate fields with separate need codes, and we understand the existing guideline (section A.3.10) as applying to a single field. It might be good to change section A.3.10 to state this unambiguously (in the first sentence, “the contents of the field are always replaced”).   We think clarification of how the extension works should go in a new subsection of A.4.3 (the non-critical extension section), with a reference from section A.3.10 (the “lists without ToAddMod” section). |
| Intel | I think we should differentiate the general case going forward from what is agreed for this particular case.  We have already captured the following:  Upon reception of a list not using ToAddModList and ToReleaseList structure, the UE shall delete all entries of the list currently in the UE configuration before applying the received list and shall consider each entry as newly created. This applies also to lists whose size is extended (i.e. with a second list structure in the ASN.1 comprising additional entries). This implies that Need M should not be used for fields in the entries of these lists; if used, UE will handle such fields equivalent to a Need R.  Non-AddMod lists will always be fully replaced and there is no mechanism for partial release or update of the entries. The issue for this specific field came about because we didn’t have a release for the original list and had a release for the ext. That won’t happen for the future releases.  In the future, we won’t have a scenario where we have an original list that cannot be released. And we won’t have the scenario where there is ambiguity on release of the extension list. The current text also implies the full list (org and ext) always has to be signalled. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 5 Discussion (Phase 2)

[To be populated]

# 6 Conclusions

Based on the discussion in sections 4 and 5 above, we propose the following outcomes:

Phase 1

Phase 2

# 7 References

[1] R2-2106736: “Report of e-mail discussion [AT114-e][022][NR16] RRC II (MediaTek), MediaTek Inc., RAN2#114-e

[2] R2-2106115: “Extension of candidateBeamRSList set to ‘release’”, MediaTek Inc./Intel Corporation, RAN2#114-e