**3GPP TSG RAN WG1 #104-bis-e R1-210xxxx**

**e-Meeting, April 12th – 20th, 2021**

**Source: Moderator (Intel Corporation)**

**Title: Discussion on**

**Agenda item: 7.2.4**

**Document for:** **Discussion and Decision**

Introduction

This document provides discussion on the following approved email thread as part of RAN1#104bis-e Release 16 NR V2X maintenance discussion:

[104b-e-NR-5G\_V2X-03] Email discussion/approval on issue M2-1: Infinite loop due to excessive resource exclusion in step 5) till 4/15, with potential CRs till 4/19 – Sergey (Intel)

Outcome

TBD

Discussion

## Round 1

The issue of an infinite loop of the resource identification procedure captured in section 8.1.4 of TS 38.214 was highlighted in contributions [1][4][5][11] for this meeting.

The main concern is that due to “hard” exclusion in step 5), the X% resource ratio on the selection window may not be achievable after any number of RSRP threshold adjustments, that practically leads to unsatisfied loop completion condition, i.e. infinite loop behaviour.

It is noted that at this stage it is unwelcomed to debate on optimizations of the exclusion procedure, such as a smarter exclusion of period values, which were already discussed in past. From FL perspective, the main aim would be to introduce a simple and efficient loop breaking condition, rather than optimize the hard exclusion step.

Finally, the context of the discussion assumes that a UE could not find the selection window size that is valid and does not enter the infinite loop condition. If this condition is not met, then the UE is first expected to adjust the selection window size.

The first phase of the discussion is to express preferences about the possible solution approaches:

Approach 1:

* Introduce a loop breaking condition

Approach 2:

* Refine step 5 (and potentially step 6) to decrease or eliminate infinite loop chances

In contributions, and as identified by FL, the following flavours of the approaches 1 and 2 are presented:

Approach 1 conditions:

* Option 1-1 [1]: If the number of resources in is already less than or equal to after step 5), UE will report the current to high layers immediately and not perform other steps (i.e. step 6 and 7)
* Option 1-2 [1]: If the number of resources remaining in will be less than after performing step 5), UE will not perform / skips step 5)
* Option 1-3 [5]: Introduce a maximum RSRP threshold of 0 dBm to ensure UE does not enter an infinite loop when performing resource selection in Mode 2 operation
* Option 1-4 [FL]: If the number of resources in is already less than or equal to after step 5), UE will report the to high layers after performing steps 6 and 7 once
  + *FL comment: this option was added based on modified option 1-1, with the difference that a UE executes steps 6 and 7 at least once to preclude collisions above the initial RSRP threshold*

Approach 2 conditions:

* Option 2-1 [1]: A subset of the (pre-)configured periodicities for reservation should be used to exclude resources in slots not monitored during sensing
* Option 2-2 [1]: Replace with to avoid excessive exclusion
* Option 2-3 [4]: In step 5) of resource selection procedure, the number of hypothetical SCI format 1-A resulting from a non-monitored slot is extended with only a single period (letting Q=1) for all configured resource reservation periods
* Option 2-4 [11]: If the number of candidate single-slot resources excluded from the set is larger than (1-X)⋅, randomly selected resources from those excluded in step 5) are added to set until the number of the candidate single-slot resources remaining in the set is not smaller than

**Q1: Do you agree that the infinite loop breaking is expected to kick in only when there is no valid selection window size that can be chosen by the UE which does not lead to the infinite loop?**

|  |  |  |
| --- | --- | --- |
| Source | Answer | Comment |
| NTT DOCOMO | (Technically YES, but) NO | In current spec, it seems that UE determines one window [T1, T2] then step 1 to 6 are applied for this window. There is no description to update the window when the UE is in the infinite loop of step 5.  For better performance, the answer should be YES, but it seems ‘optimization’ as mentioned by FL. The window is selected by UE implementation, so we can expect smart UEs for this aspect. |
| Ericsson | Yes | There is a possibility of infinite loop, and therefore, a procedure to break the loop should be triggered under certain condition(s). |
| Huawei, HiSilicon | Clarification needed on Q1 | We are not sure if the proposed pre-condition can be real, in the sense that we only know the UE has arrived in the infinite loop under the condition of its implementation choices. It does not matter how many sizes of selection window it has first tried without success (e.g. a single size, or many sizes). It only matters what to do once the infinite loop is found.  We suggest not discussing Q1, and proceeding with Q2 directly. |
| QC | NO | [T1, T2] is chosen before entering the loop. If there is infinite loop, UE cannot break out of the loop to reselect another window of [T1, T2] |
| Sharp | Yes | We share similar view as FL.  Regarding the infinite loop issue itself, proponents mainly share the view that the percentage X could be configured larger than LTE V2X and reservation interval could be finer and more, which both lead to potential infinite loop. In our view, such differences cannot be quantified exactly and actually even in LTE V2X, such infinite loop issue may also happen w/o introducing solutions.  With selecting the resource selection window up to implementation, the issue could be addressed. To us, option 1-2 seems like a method which operates step 5) first and then discover the infinite loop issue, then, skip step 5) to address it. |
| OPPO | No | In our opinion, adjusting the size of selection window is just one of the solutions to jump out of the infinite loop and it should be treated same as other solutions (e.g., Option 1-1,1-2…) rather than prior to other solutions. In addition, UE still need to perform Step 5 again after adjusting the selection window to judge whether the selection window is valid or not. In the worst case, UE has to use another listed solution when it has already tried all the combinations of T1 and T2. The processing time and complexity will increase by doing this way due to performing Step 5 again and again. Finally, it is also hard to specify such behaviour because T1 and T2 are up to UE implementation. Therefore, we tend to agree with QC that the selection of [T1, T2] should not be adjusted since it is before entering the loop. |
| Vivo | Partially yes | There are some other reasons, e.g., too many periodicities configured for the pool, very large X%... The infinite loop issue exist when non-proper parameters are configured. |
| Futurewei | No | T2 is decided before the procedure. UE cannot change T2 to break the loop. |
| Panasonic | NO | We share similar view with QC that [T1, T2] is chosen before entering the loop and it should not be adjusted. |
| CATT,GOHIGH | NO | We share the similar view with QC. The resource selection window should be determined before resource exclusion is performed. Logically, if there is no break conditions for infinite loop, UE cannot stop the infinite loop to adjust the T1 and T2. If there is a break condition, no need to further adjust the T1 and T2. Additionally, even the size of resource selection window can be changed, it cannot ensure that the infinite loop can be avoided. |
| ZTE, Sanechips | Yes | Infinite loop could take place |
| LG Electronics | No  (see our additional comment) | First of all, we agree with other companies’ understanding on the current specification that T1 and T2 are selected before entering Step 5/6 and it is not possible to re-select the values of T1 and T2 after finishing Step 1. Considering this aspect, we think that **it should be clearly clarified under what conditions the UE is allowed to check whether it enters in the infinite loop**. In other words, it should be avoided that the UE checks whether it enters the infinite loop after intentionally setting T2 to a small value (e.g., T2min), which leads to make Normal Mode 2 Resource Selection Procedure impossible. **One possible solution to handle this issue is that it can be defined that only when the value of T2 is set to the remaining PDB, the UE is allowed to check whether it enters in the infinite loop.**  Secondly, the point of this email discussion is how/whether to handle the case where the remaining number of candidate resources after Step 5 is less than the (pre)configured minimum number of candidate resources. This is clearly clarified in the subject of this email discussion assigned by Chairman. **We would like to emphasize that further enhancement targeting other scenarios except the case mentioned above is not the scope of this email discussion.** |
| Nokia, NSB | No | As described in 38.214, selection window is chosen first. |

**Q2: Please answer which of the above approaches to fix the infinite loop issue should be pursued in R16 V2X maintenance?**

* **Approach 1: Introduce a loop breaking condition**
* **Approach 2: Refine step 5 (and potentially step 6) to decrease or eliminate infinite loop chances**
* **Combination (please specify)**

**Please also express the views on the detailed handling option (1-1, 1-2, etc.)**

|  |  |
| --- | --- |
| Source | Comment |
| NTT DOCOMO | Approach 1 is preferred. Original concept of step 5 should be maintained and only addressing this issue is better. Approach 2 might lead to new issue, unless careful studies/evaluations.  Among approach 1, we prefer option 1-3/1-4. |
| Ericsson | Approach 1 is preferred at this point. Option 1-2 looks like a simple approach to avoid the infinite loop skipping step 5) under the condition that will be less than after performing step 5) |
| Huawei, HiSilicon | First of all, to avoid any misunderstanding, it’s necessary to clarify that all the options shall only apply to the “infinite loop” issue, i.e., if “infinite loop” issue does not happen, none of the options will be effective. Therefore, we suggest to add “If the number of the excluded resources in step 5) is larger than , …” to each option.  Otherwise, the options except Option 2-4 seem to change R16 Mode 2 behaviours in all cases, i.e., regardless of whether “Infinite loop” issue happen or not. Any such general change is out of the scope of this thread.  We support Option 2-4, which can well solve the “infinite loop” issue and provides MAC layer candidate single-slot resources.  The main drawbacks of other options are:   * For all the Option 1-1/2/3/4:   + One major problem is the final number of resources in SA could be much smaller than , e.g., when the periodicity value is very small. Consequently, MAC layer has very few candidate resources to be selected, resulting in large collision chance and some timing restrictions cannot be satisfied (e.g., HARQ RTT, chain reservation, etc.) * Option 2-1:   + If the sub-set is obtained by RRC configuration, then this solution shall not be considered due to RRC impact.   + If the sub-set is derived based some predefined rule, e.g., the first X periodicity values as proposed by [1], then it’s possible that even we apply this sub-set, the infinite loop issue still exists since the first X periodicity values may still include very small periodicity values. * Option 2-2, 2-3:   + This changes the basic design principle of sensing procedure, and could face serious collision issue since resources corresponding to Q>1 or are not considered. |
| QC | Approach 1 is preferred |
| Apple | Prefer a combination of Approach 1 and Approach 2. If the infinite loop does not occur, i.e., the number of candidate resources after step 5 is larger than , then nothing needs to be done. Otherwise, step 5 is modified (e.g., Option 2-x) so that the number of candidate resources after modified step 5 is larger than |
| Sharp | If the infinite loop is deemed as an issue following majority view, we support option 1-2/-1. |
| OPPO | We prefer Approach 1. Compared with Approach 2, only an independent behaviour needs to be specified in Approach 1 without modifying the details of Step 5 or Step 6. Furthermore, if Approach 2 cannot completely avoid the infinite loop, we still have to go to Approach 1.  Within Approach 1, Option 1-2 or a new Option 1-5 (adjust the value of X% to avoid infinite loop after step 5).  Option 1-1 will lead to the performance loss because UE doesn’t exclude any resource based on SCI decoding. In Option 1-3 and Option 1-4, UE should not continue exclude resources from the candidate resource set when excessive exclusion has already happened.  Our preference is Option 1-2. The intention of Step 5 is to solve the issue of half-duplex. If it doesn’t cause the infinite loop, we can keep it as LTE-V2X. Otherwise, skipping it is the simplest way to jump out of the infinite loop.  In addition, if the infinite loop is caused by a large value of X% (e.g., 50%), it is also a good choice to reduce the value of X when the number of resources in is already less than or equal to after step 5). We additionally propose this as Option 1-5 as follow   * A maximum value for X is selected from the set {0.2, 0.3, 0.5} such that is satisfied, where is remaining resources after step 5. |
| vivo | If after optimization, there is still possibility to incur infinite loop, we prefer to keep the spec. unchanged. Following the above principle, we think approach 1 or option 2-4 in approach 2 can be further considered to avoid infinite loop. |
| Futurewei | We prefer approach 2. Comparing the schemes listed under the two approaches, we do not agree that approach 1 is simpler than approach 2. Arbitrarily breaking loop also impacts the performance and could destroy the original design in Rel-16. For approach 2, we prefer approach 2-4 with a slight modification to avoid the infinite loop completely, as   * Option 2-4A: If the number of candidate single-slot resources excluded from the set is larger than (1-X)⋅, randomly selected resources from those excluded in step 5) are added to set until the number of the candidate single-slot resources remaining in the set is not smaller than where, =5% for example. |
| Panasonic | We prefer approach 1. Among approach 1, we prefer 1-3 and 1-4. |
| CATT,GOHIGH | Option 1-2 or option 2-1 are preferable.  In order to avoid the infinite loop essentially, solution to avoid excessive and unnecessary exclusion by step 5) can be supported. Both option 1-2 and option 2-1 can achieve that target. Additionally, if option 2-1 is accepted, the subset can be the periodicity value of the upcoming transmission and 100ms. |
| Samsung | We have raised the issue about “hard” exclusion in step 5) before and suggested to remove whole step 5) procedure. However, it was treated as minor issue. Now, if it is considered as issue to be handled, we first need to discuss about whether it is really critical or not (I think it is related to Q1 but we need to reformulate the question). If yes, we think that enough evaluation and analysis should be carried out to resolve the issue including the option for removing step 5). If not, we prefer to keep the spec without further optimization. |
| ZTE, Sanechips | We prefer option 1-3, which is a straightforward solution to address the issue. |
| LG Electronics | Our first preference is Option 1-4 of Approach 1, which is more aligned with the principle of sensing/resource selection procedure in Rel-16. Regarding Option 1-3 of Approach 1, we are not sure whether it can be a solution covered by the scope of this email discussion as already commented in Q1. |
| Nokia, NSB | At this stage, we should aim for a simple rather than for a perfect solution. Hence we prefer approach 1. Option 1-4 looks sensible. |

## Round 2

As per the discussion around Q1, it seems it is not possible to make such a conclusion. However, it seems the intention of the question needs to be clarified, like what was explained by LGE. In FL understanding, a UE should be discouraged to pick the selection window SW1 = which goes to the infinite loop if there is another valid combination SW2, which does not lead to the infinite loop.

It should be emphasized, that up to this moment w/o fixing the loop, a UE is anyway expected to do this.

I’m not sure whether/how to capture this expectation from the UE. One example is what LGE suggested, but it may not fly eventually. We can try a conclusion which is not intended to be captured in specifications:

**Conclusion 1**

* A UE is not expected to pick a valid selection window size which leads to the infinite loop of the identification procedure in section 8.1.4 of TS 38.214, if there is another valid selection window size, which does not lead to the infinite loop

**Q3: Is the above conclusion agreeable? Please indicate other suggestions if you don’t agree.**

|  |  |
| --- | --- |
| Source | Comment |
|  |  |
|  |  |

Regarding the distribution of opinions about Approach 1 vs Approach 2, the following is observed:

* Approach 1
  + 9 sources
* Approach 2
  + 3 sources (including Samsung option of removing step 5)
* Approach 1 or 2 (more than one preference indicated)
  + 3 sources

It seems the majority is for Approach 1, which is also more inline with the spirit of fixing the issue w/o going into optimization. I suggest we take a higher-level agreement first to confirm the intention of Approach 1.

**Proposal 2**

* Update the specification of identification of candidate resources for Mode-2 resource allocation in section 8.1.4 of TS 38.214 to introduce a loop stopping condition when X\*M\_total number of identified resources could not be reached after any number of loop iterations
* Note: The detailed condition is discussed separately

**Q4: Please indicate you support and/or modifications to P2 above.**

|  |  |
| --- | --- |
| Source | Comment |
| QC | Support |
|  |  |

Assuming P2 is agreed, between the options of Approach 1, the following distribution can be found:

* Option 1-1: 1 source
* Option 1-2: 4 sources
* Option 1-3: 3 sources
* Option 1-4: 4 sources
* Option 1-5 (see OPPO): 1 source

It seems options 1-2, 1-3, 1-4 have similar support, and can be further considered. I’ve refined the options to be more accurate and potentially ready for endorsement.

**Proposal 3**

* Adopt one of the following detailed condition to resolve the issue in Proposal 2 (to be down selected)
  + (Option 1-2) If the number of the excluded resources in step 5) is larger than , a UE skips step 5
  + (Option 1-3) In step 7, if all thresholds reached 0 dBm value, the UE reports to higher layers without checking that the number of elements in is than
  + (Option 1-4) If the number of the excluded resources in step 5) is larger than , a UE reports the to high layers after performing steps 6 and 7 once

**Q5: Please indicate which of the above options you prefer.**

|  |  |  |
| --- | --- | --- |
| Source | Option | Comment |
| QC | 1-3 | Other options has not been checked by evaluations |
|  |  |  |

References

1. R1-2102369 Remaining open issues and corrections for mode 2 RA OPPO
2. R1-2102589 Discussion and TPs on resource allocation in NR V2X CATT, GOHIGH
3. R1-2102941 Maintenance on NR sidelink mode-2 resource allocation mechanism vivo
4. R1-2103081 On Remaining Issues of Mode 2 Resource Allocation Apple
5. R1-2103143 Remaining Issues in Mode 2 Resource Allocation Qualcomm Incorporated
6. R1-2103467 Remaining issues on resource allocation for NR sidelink Sharp
7. R1-2103501 Draft CR of TS38.214 ZTE, Sanechips
8. R1-2103516 Remaining issues on resource allocation mode 2 NEC
9. R1-2103639 Remaining issues on sidelink mode 2 ASUSTeK
10. R1-2103750 Correction on resource exclusion for other TBs Huawei, HiSilicon
11. R1-2103751 Correction on step 5 of mode 2 resource allocation Huawei, HiSilicon
12. R1-2103765 Maintenance for Resource allocation for sidelink - Mode 2 Nokia, Nokia Shanghai Bell