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

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

**Source: Moderator (Intel Corporation)**

**Title: Discussion on [104b-e-NR-5G\_V2X-03]**

**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 |
| NTT DOCOMO | Not support. No conclusion is fine for this aspect.  As commented above, in current spec UE sets window firstly and then step 1 to 7 are applied. There is no rule to update the window when the infinite loop is found. We think the new rule is an optimization and not essential. |
| Sharp | The proposed conclusion seems an optimization at this stage and thus not needed. |
| FUTUREWEI | As commented before, T2 is decided before the procedure. |
| Vivo | The intention to have this conclusion is not clear, since we are going to discuss the enh. in the following proposals. Do you mean, if we support this conclusion, no more discussion on the remaining proposals? |
| CATT, GOHIGH | Not support.  Principle of the above conclusion cannot be achieved by the current spec. And it is not a valid solution to the issue of infinite loop. |
| ZTE | Disagree, no conclusion is needed.  According TS38.214:  - selection of is up to UE implementation under , where is defined in slots in Table 8.1.4-2 where is the SCS configuration of the SL BWP;  - if is shorter than the remaining packet delay budget (in slots) then is up to UE implementation subject to remaining packet budget (in slots); otherwise is set to the remaining packet delay budget (in slots).  The selection window is up to UE implementation. Even if we consider partial sensing case, during selection window, there are two parts of sensed slots. Yet we should not regard them as two selection windows, but instead two sets of resource set Ys.  So we still think for sensing, there are only one selection window, which is up to UE implementation. This conclusion is not necessary. |
| Panasonic | Not support. As we commented before, the window is determined before entering the loop. We think it would be ok without concluding it. |
| OPPO | Disagree. As mentioned in Round 1, UE need to perform Step 5 to judge whether the current selection window will lead to the infinite loop or not. In order to select a valid selection window, Step 1-5 may be performed for many times. In the worst case, the valid window cannot be found but UE processing time increases. In our view, Conclusion 1 is not needed. |
| Samsung | This conclusion is not necessary |
| LG | The point is that we need to define a rule that prohibits the UE from performing the wrong behaviour **intentionally** which has the **negative impact on Mode 2 performance of other UEs** in the resource pool. Actually, we already made many agreements to prevent this possibility. For example, the minimum value of T2 was defined in order to resolve the problem that the collision probability is increased by the UE determining the selection window size excessively small. We would like to emphasize that defining this kind of rule is not the optimization, and when we introduce the mechanism to avoid the problem of “infinite loop”, it should be considered together. Note that all the options listed in Proposal 3 are to skip some parts of Normal Mode 2 Resource Selection Procedure, and these definitely have the impact on other UEs’ Mode 2 performance in the resource pool.    From our perspective, the simple way to resolve this issue could be that the solution to resolve the problem of “infinite loop” is applied only when the value of T2 is set to the remaining PDB. |
| Ericsson | As indicated by FL since up to this moment w/o fixing the loop, a UE is anyway expected to do this, then there is no need to have this conclusion. |
| NEC | Not necessary bacause this conclusion seems more like a UE implentatio issue. |
| Nokia, NSB | We don’t see the need for this conclusions. |
| Huawei, HiSilicon | The window size is up to UE implementation and decided before going to the loop. The proposed conclusion does not solve the infinite loop issue.  So there is no need to have this conclusion. |
|  |  |

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 |
| NTT DOCOMO | Agree |
| Sharp | Agree |
| FUTUREWEI | We do not support this proposal if the loop stopping condition only include the approach 1 as in the current definitions of approach 1. Actually, the definitions of the two approaches are not mutually exclusive. For example, any proposal in Approach 1 will add one step after 5) which can be viewed as redefining 5). Our proposed 2-4A, 2-1, or 2-4 is also a process which will break the loop. The process can be added after 5) without changing 5), i.e., in the same manner for fixing the spec as other proposals in Approach 1. The reverting the excluded resource back in our proposal is similarly to approach 1-2 which is simply a special case of proposals 2-1, 2-4, or 2-4A, i.e., reverting all the excluded resource back.  Therefore, more discussions are needed before reaching conclusion on this proposal. |
| vivo | We can accept, if the majority prefer such optimization |
| CATT, GOHIGH | Agree |
| ZTE | OK |
| Panasonic | Support |
| OPPO | Support, our understanding to agree on this Proposal 2 is to go with Approach 1. |
| Samsung | We can accept if the majority support this proposal. |
| LG | Agree |
| Ericsson | Agree |
| NEC | Agree |
| Nokia, NSB | Support |
| Huawei, HiSilicon | Disagree. No need to have this proposal since the definition of “a loop stopping condition”is unclear and needs to be separately discussed.  For example, Option 2-4 or 2-4A can eliminate infinite loop chances, so it’s also a kind of loop stopping condition. Moreover, Option 2-4 or 2-4A is targeted to solve the infinite loop issue, and does not have more specification impact compared to other options, so it’s also inline with the spirit of fixing the issue w/o going into optimization. |
|  |  |

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 |
| NTT DOCOMO | Option 1-3 or Option 1-4 | Option 1-2 does not apply step5, this means there are potential resource collisions since identified resource set does not reflect potential reservations at slot where the UE did transmission. It will be undesirable.  Option 1-3 keeps current resource allocation behaviour and stops the loop after sufficient resource exclusion. Although no. of resources in the set is small, few resource collisions will be possible.  Option 1-4 applies once current resource exclusion of step 5 to 7. This option also will not lead to more resource collisions. |
| Sharp | 1-2 | If option 1-4 is going to be agreed, in our view, there is no need to perform step 7 (increment of threshold) once and only step 6 is enough. Regarding option 1-3, it seems “If the number of the excluded resources in step 5) is larger than ” in the other two options also applies, we suggest to add it, in order not to bring confusion. |
| FUTUREWEI | 2-4 or its modifications | Again, more discussions on proposal 2 are needed before moving on to this stage.  Also as commented before, our proposal 2-4A is also a loop breaking step and can be inserted as addition step without change step 5, same as other schemes in Approach 1, which should be included as an option in the proposal.  Since 3 sources mentioned 2-4 or its modification, 4 if including 1-2 as a special case of 2-4, we suggest including 2-4 or its modified versions, as an option for discussions. |
| vivo | 1-3/1-4 | 1-3/1-4 is more reasonable considering the resource with uncertainty interference would not be reported in SA |
| CATT, GOHIGH | 1-2 | For option 1-2, if step 5) causes the excessive exclusion, it can be skipped. Then resource exclusion are performed by the actual sensing results. Therefore, most of collisions can be avoided and sufficient candidate resources can be reported to MAC layer.  For option 1-3, it cannot solve the issue of excessive exclusion. If excessive exclusion in step 5) is serious, only a few candidate resources can be reported. They may not be sufficient for MAC layer to select transmission resources.  For option 1-4, resource exclusion is performed using the hypothetic SCI and actual SCI with the most severe SL-RSRP threshold. Then only a few candidate resources can be reported. Similar to option 1-3, the reported resources may not be sufficient for MAC layer. |
| ZTE | 1-3 | For 1-2, if step 5 is skipped, performance degradation is supposed to take place because no mechanism assure the detection of collision over these slots.  For 1-4, if step 7 is skipped, in that case, very few candidate resources may reside.  Option 1-3 seems a good compromise between the number of candidate resources of and collision/interference. |
| Panasonic | 1-3 or 1-4 | For resources with high interference, it’s not very meaningful to report to MAC layer thus limited iterations are preferred.  For option 1-3, a max RSRP threshold (0dbm or other values) is introduced as the stop condition, which is easier to adopted and has little spec impact.  For option 1-4, it runs the iteration once and not affect current procedure much. |
| OPPO | Option 1-2 | When the procedure of mode 2 goes into the infinite loop, it means that Step 5 has already excluded a lot of resources regardless how many times the RSRP threshold has increased. In option 1-3 or option 1-4, it is not proper for the UE continuing to exclude resources from the candidate set SA in such condition. Furthermore, option 1-3 and option 1-4 are more likely to cause re-transmission dropping compared with option 1-2 due to the small candidate resource set reported to the higher layer. And currently in the MAC spec, there is no procedure / mechanism to handle the case when there is insufficient amount of resources being reported.  Furthermore, with only limited amount of resources reported to the higher layer, it is also more likely that the two timing restrictions (i.e., a resource indicated by a prior SCI, HARQ RTT time gap) cannot be met.  In addition, in Option 1-3, we wonder how 0dBm is selected? Also, in our view, 0dBm is a very high value compared to -128dBm. If 0dBm is used, it means the RSRP thresholds for all priority combinations would have increased more than 40 times. As such, we can expect no resources in Step 6 would be excluded.  Therefore, our preference is option 1-2. |
| Samsung | Option 1-2 | We are not clear step 5 procedure is really beneficial but it seems that nobody considers removing step 5 as another option. Among the listed options, our preference is Option 1-2. |
| LG | Option 1-4 | In our reading of Option 1-3, it will be also applied even for the case when the remaining number of candidate resources after Step 5 is larger than or equal to the (pre)configured minimum number of candidate resources (i.e., X∙Mtotal). Again, this case is not the scope of email discussion. Note that the subject of email discussion assigned by Chairman is “Infinite loop due to excessive resource exclusion in step 5)”. Also even though the concept of limiting the maximum value of RSRP threshold increment is applied for the case when the remaining number of candidate resources after Step 5 is less than X∙Mtotal, at this moment, it is not clear with the difference in performance between Option 1-4 and Option 1-3.  Technically speaking, we don’t see any benefit of Option 1-2 compared to Option 1-4. |
| Ericsson | Option 1-2 | Step 5 can be skipped if the condition is fulfilled since most of the (potential) collisions can/may be detected by performing step 6. |
| NEC | Option 1-4 | Option 1-2 is adjusted too more by totally skipped step 5), too many possible reservations will skipped.  Option 1-3 is not deficated to hande the Infinite loop caused by step 5)  Option 1-4 to perform step 5 and 6/7 once is a compromise |
| Nokia, NSB | Option 1-3 or 1-4 |  |
| Huawei, HiSilicon | Option 2-4 or 2-4A | The key issue is to ensure MAC layer gets at least candidate single-slot resources. This should be the first goal of any fix to the problem, because when that is achieved, the problem itself is removed. Whereas other options leave the problem there, unfixed.  One clear benefit of Option 2-4/2-4A over other options is that it not only eliminate infinite loop issue, but also can provide MAC layer at least candidate single-slot resources.  Option 1-3 seems 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. So Option 1-3 should not be further considered. Moreover, in R16 maintenance phase, upper bound on RSRP threshold has already been discussed and precluded. This issue should not be pursued again.  Option 1-2 is too dangerous since step 5) is totally ignored and potential resource collisions cannot be identified.  A major problem of Option 1-4 is that the final number of resources in SA could be much smaller than , which is highly possible since infinite loop happens and lots of candidate resources will be excluded in step 5) due to small periodicity values. 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.). |
|  |  |  |

## Round 3

It seems almost a consensus that Conclusion 1 is not required, thus it will not be pursued further. Instead let’s focus on P2 and P3.

As for P2, it is observed that the majority, 13 sources, support or accept the proposal, while 2 sources do not support the proposal since it precludes their preferred option 2-4/2-4A.

From FL perspective, the claim that 2-4/2-4A avoid very small S\_A is not completely true, since step 6 and 7 is evaluated in addition, and exclude the resources further based on RSRP. In this case, the infinite loop is not resolved, since there is no loop breaking condition introduced. The enhanced option 2-4A improves the situation with delta\_X, but there is no universal delta\_X value which can be chosen to prevent all infinite loops.

Meanwhile, there are some technical aspects which are highlighted by FL:

* **Some technical questions/concerns from FL to 1-2**
  + **Skipping step 5 is not completely following Approach 1 (this is probably my mistake of classification in the first round), but it seems a too radical option, since it gets back all the “potential collision due to half-duplex” resources, and ignores these collisions completely. Note, that the group does not want to make a conclusion similar to C1 in round 2, that means the UE is free to take a small selection window and then face the over-exclusion issue, which skips step 5.**
* **Some technical questions/concerns from FL to 1-3**
  + **While introducing 0 dBm threshold solves the issue, it seems the stopping in this case happens in the most extreme state of the RSRP thresholds, i.e. almost no consideration of soft collision metrics.**
* **Some technical questions/concerns from FL to 2-4/2-4A**
  + **According to the TP provided by Huawei/HiSilicon in [11], it introduces a step 5-1 which randomly adds back excluded resources to reach X\*M\_total (or (X+delta\_X)\*M\_total). This means there is no explicit loop breaking condition, which leads to executing existing steps 6-7.**
  + **In the non-initial iteration, it seems the random operation of getting back some resource to S\_A will lead to another outcome (due to randomness), and can completely change the S\_A from iteration to iteration.**
  + **It seems the steps 4-7 will be repeated until there is a random outcome of step 5 which gets back resources with RSRP less than the threshold(s).**

As for the options in P3, the following distribution is observed:

* 1-2
  + 5
* 1-3
  + 6
* 1-4
  + 6
* 2-4/2-4A
  + 2

It is observed that there is no clear majority. Since there were some concerns on formulations, the last attempt is made to present the refined options, and then select by majority.

Updated proposal (P2 + P3):

* Update the specification of identification of candidate resources for Mode-2 resource allocation in section 8.1.4 of TS 38.214 to handle the case ~~introduce a loop stopping condition~~ whenX\*M\_total number of identified resources could not be reached after any number of loop iterations
* Down-select this meeting:
  + (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 without increasing RSRP thresholds
  + (Option 2-4/2-4A): 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

**Q6: Please indicate whether you support the main bullet and which of the sub-bullets you prefer. More than one option possible. Also pay attention to FL technical comments in the beginning of Round 3 section.**

|  |  |  |
| --- | --- | --- |
| **Source** | **Option** | **Comment** |
| NEC | Option 1-4 | Also support main bullet. |
| LG | Option 1-4 | As already commented in the previous round, according to the current wording of Option 1-3 (see below), it can be interpreted that Option 1-3 is applied even when the remaining number of candidate resources after Step 5 is larger than or equal to the X∙Mtotal. We have strong concern on this point because this case is not the scope of email discussion.  *(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*  In order for Option 3 to remain within the scope of email discussion, we think that the wording of Option 3 should be modified as follows:  *(Option 1-3) If the number of the excluded resources in step 5) is larger than and all thresholds reached 0 dBm value in step 7), the UE reports to higher layers without checking that the number of elements in is than* |
| OPPO | Option 1-2 or (1-2 + 1-4) | First of all, we don’t need the first bullet, as it has no spec impact, and it is sufficient knowing that we are here to solve the infinite loop problem.  To clarify, Option 1-2 is not the same as conclusion 1, where it was proposed to find a suitable selection window even before entering the loop, which was the main concern from the group. In Option 1-2, it is trying to resolve the infinite loop problem when it occurs.  For Option 1-3, our previous concerns remained, where the RSRP threshold will increase more than 40 times and nearly no resource can be excluded in Step 6.  As for Option 2-4/2-4A: In 2-4, the consequence is that step 6) is effectively not performed because the minimum number of candidate resource is reached and no more resources can be excluded. And as such collisions cannot be avoided with reserved resources. In 2-4A, how to determine the delta-X such that there is sufficient number of remaining resources in SA for resource exclusion in step 6). Furthermore, in both schemes, there would be problem with the re-evaluation checking, where an original selected resource was from one of the added-back resources from step 5-1) during the initial selection but was not added-back again during re-evaluation and causes the resource to be re-selected which should not happen.  I think we need to bear in mind that the infinite loop problem is not always caused by not monitor many slots during the sensing window. It can also be caused by small reservation periodicities configured for the RP or a large configured value for X even when small number of slots were not monitored.  Our preference is still Option 1-2 as there are still some problems exist with Option 1-4.  If to find a compromised solution, then we see there are some cases can be solved by Option 1-2 and others by Option 1-4. One method is that:   * When X = 0.2 or 0.3, use Option 1-2 * Else, when X = 0.5, use Option 1-4.   The reason for this is that since option 1-4 excludes both step 5) and step 6) resources, it is not suitable when the configured X value is small, where it will lead to the problem of small number of available resources for MAC selection and not be able to satisfy the 2 timing restrictions. On the other hand, Option 1-2 would be able to compliment this short coming by skipping step 5). Note that, when the number of non-monitored slot is low, by skipping step 5) will not harm the PRR performance. |
| CATT,GOHIGH | Option 1-2 | As clarified by FL, the issue is to handle the case when X\*M\_total number of identified resources could not be reached after any number of loop iterations. The reason that leads to this case is excessive exclusion in step 5) and the key is to alleviate the excessive exclusion. And we think the exclusion operation based on actual received SCIs is more important than that based on hypothetic SCIs.  For option 1-3 and option1-4, excessive exclusion based on hypothetic SCIs cannot be alleviated but exclusion based on actual SCIs are alleviated. They cannot achieve the aforementioned target.  For option 2-4/2-4A, after randomly selecting and adding some excluded resources to set A, a part of resources would be excluded by the received SCI and SL from remaining resource. Then iteration of step 6) and 7) would be performed and the worst case is that all excluded resources are based on hypothetic SCIs. It is not a valid solution.  Therefore, we support option 1-2. |
| Huawei, HiSilicon | Revised Option 1-4 (Combination of Option 1-4 and 2-4) | In previous rounds of replies, quite a few companies already mentioned PHY needs to provide enough candidate resources to MAC layer. This should be the first goal of any fix to the problem. If MAC layer has very few candidate resources to be selected, there will be large collision chance and some timing restrictions cannot be satisfied (e.g., HARQ RTT, chain reservation, etc.).  Since we are dealing with infinite loop issue, it is expected a lot of candidate resources will be excluded in step 5). So one major problem of Option 1-4 is that the final number of resources in SA could be much smaller than , and faces the issue we mentioned at the beginning.  We would like to find a compromise solution out of the technical discussion and inputs, as we see risks in just voting towards an outcome. As a way forward, we suggest that a combination of Option 1-4 and 2-4 can solve this issue (see “Option 1-4 revised” below).  That is, if infinite loop issue happens after step 5), some resources are added back to ensure there are sufficient number of resources to be further checked in step 6-7. Then, a UE reports the to MAC after performing steps 6 and 7 once without increasing RSRP thresholds   * (Option 1-4 revised) If the number of the excluded resources in step 5) is larger than , 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 , then a UE reports the to higher layers after performing steps 6 and 7 once without increasing RSRP thresholds   On Option 1-2: We share similar view with FL and other companies that skipping step 5) is too radical. There could be serious interference since such collisions are totally ignored.  On Option 1-3: We share similar view with other companies that 0 dBm is a very high value and thus leading to serious interference. |
| Futurewei | 2-4/2-4A | We are ok with the 1st main bullet  For option 1-3, with 0dBm RSRP threshold, it may stop at a small number of available resources, which could lead to a large collision rate. Also as Huawei commented, this option changes R16 mode 2 behaviour and the upper bound on RSRP threshold has already been discussed and precluded.  For option 1-4, similarly, without checking the criterion of , it could also result in a small number of resources even with steps 6-7 once, which leads to a large collision rate potentially.  Moreover, both options 1-3 and 1-4, as well as modified 1-4 versions other companies brought up, have a serious issue. Without checking the criterion of | , not only just resulting in a small , these schemes could also possibly **lead to an empty**  . which will break the system. Then another fix will be needed. Also, with option 1-3 or 1-4, we are reversing the R16 agreement on | The behaviour of the final outcome from the entire exclusion procedurescompletely changes. There could be many additional issues from subsequent processes.  Therefore, option 1-3 and 1-4, as well as modified 1-4 should not be agreed, and should be excluded from further discussion.  Option 1-2, as highlighted by FL, skipping step 5) is a too radical option. Therefore, it is not preferred in its current form. Another problem of option 1-2 is that step 5 is not skipped. Based on current fix, it is executed in each iteration and then revert all the excluded resources back. If step 5 is just run in the first iteration, more changes are needed in multiple places, e.g., with iteration index, in the spec.  For 2-4/2-4A, here are the answers to the comments to 2-4/2-4A from FL.  1st bullet, with RSRP increasing in each iteration, it is guaranteed that the loop will stop within a finite number of iterations, with or larger number of available resources in before step 6 in each iteration.  2nd bullet, it is ok to have different outcome on after proposed step 5-1 in each iteration as loop-stopping is guaranteed. A random outcome might be good actually if the random reversion in previous iterations is not good. The procedure may settle in a better opportunistically by avoiding reverting some bad resources that will be excluded again with step 6. This benefit becomes more obvious if an inner iteration of steps 5, 5-1, and 6, for multiple trials with same RSRP threshold. The procedure can be certainly improved but more discussions are needed. We are also ok to consider a predefined order if the randomness draws many concerns.  3rd bullet, increasing RSRP threshold in existing spec is a way to get back resources with RSRP less than a (new) threshold. A random outcome of step 5-1 helps too. As addressed to the 2nd bullet, the randomness may help to settle at a better . Again, some enhancements can be done with some additional changes, but more discussions are needed. Option 2-4A should be a good choice with no or less concerns from this comment, as well as other two comments.    So we propose to support 2-4A, which could cover 2-4. We are also willing to discuss a possible revision to cover 1-2, as well as using a pre-defined order to add the excluded resources back. Note that the option 2-4/2-4A in proposal does not include 2-4A, i.e., the term . The correct 2-4A is provided below.  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 can be simply a fixed value e.g., 5%, or configured from a predefined list, e.g., [0, 5%], or a range, e.g., 0<= <=20%. With =0, it becomes 2-4. If 1*-X* is included as a choice of , then 1-2 is included in this fix. |
| Ericsson | Option 1-2 | A simple way to end the loop is to skip step 5) which in our view does not provide so critical information about potential reserved resources when compared with step 6). Moreover, we think that in order to avoid any extra specification and potential extra changes going for Option 1-2 is the easiest way. |

## Round 4

It was interesting discussion.

Option 1-2

* 3 sources

Option 1-4

* 2 sources

Option 1-2 + 1-4

* 1 source

Option 1-4 + 2-4

* 1 source

Option 2-4/2-4A

* 1 source

There is no clear majority, except that 1-2 gets slightly more support.

There is a valid point brought by OPPO regarding 2-4/2-4A:

* **in both schemes, there would be problem with the re-evaluation checking, where an original selected resource was from one of the added-back resources from step 5-1) during the initial selection but was not added-back again during re-evaluation and causes the resource to be re-selected which should not happen**

In FL observation, this makes 2-4/2-4A not suitable/valid for our issue, since those don’t work with pre-emption and re-evaluation properly. To fix this, more complications need to be added, e.g. expect from the UE to add back exactly same resources in the selection window as were added in the previous iterations and re-evaluation/pre-emption checks. This does not make much sense at the late CR stage.

Based on this, it seems we narrowed down to two options only: 1-2 and 1-4. Further, the combination between 1-2 and 1-4 proposed by OPPO may eventually resolve the issue of down-selection but may not be easily acceptable.

Since it is already the last day of the technical discussion on this issue, it is proposed to go with the combined 1-2 + 1-4 option.

Updated proposal

* (please see email)

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