3GPP TSG-RAN WG3 #114-e R3-216013

**E-meeting, 1st – 11th November 2021 Review of R3-215854**

Source: CATT (moderator)

Title: CB: # SONMDT5\_RACHOpt

Agenda Item: 10.2.1.7

Document for: Approval

# Introduction

**CB: # SONMDT5\_RACHOpt**

**- Turn the WA to agreement?**

**- Maximum number of PRACH Configurations?**

**- PRACH configurations encoding? approach 1,2,or 2 enhanced?**

**- DU can request further PRACH configurations from gNB-CU?**

**- Neighbour PRACH configurations over F1AP? X2 impact?**

**- Capture agreements and open issues**

**- Proceed to TPs if there are agreements.**

(CATT - moderator)

Summary of offline disc R3-215854 rev in R3-216013

The deadline for the first phase is 00:00 UTC on 5th November (Friday).

For 2nd Round:

* Find the compromise way on the maximum number of potentially-in-conflict served cell list

The deadline for the second phase is 00:00 UTC on 10th November (Wednesday).

# For the Chairman’s Notes

**Already agreed in the 1st round:**

**It should be possible for the gNB-CU to provide the gNB-DU with information indicating the CGI of the cells potentially in conflict and the neighbouring relation between these cells and their neighbour cells, along with the PRACH configurations of those neighbour cells, so as to prevent the gNB-DU from reconfiguring one of its cells from conflicting with one neighbour toward conflicting with another neighbour. How/whether gNB-CU do the filter is up to implementation.**

**Approach 2bis is adopted**

**Maximum number of potentially-in-conflict served cell list is FFS and neighbour cell list is 32**

**PRACH configuration is not included in F1 setup response message**

# Discussion (second phase)

In the second phase of discussion we will focus on two open issues:

* the maximum length of potentially-in-conflict served cell list,
* whether the gNB-DU can request the gNB-CU to provide neighbour’s PRACH configurations.

## Maximum length of potentially-in-conflict served cell list

The core of this issue is exact the same issue as Q1.3.2-1, i.e.:

What value should the maximum (i.e. upper bound) length of potentially-in-conflict served cell list (i.e. the outer layer) be defined as? Please provide the range of that maximum value you considered acceptable and your most preferred value.

**Questions 2.1-1**: Companies are invited to provide your views on the maximum value you considered acceptable and your most preferred value

| Company | Comment |
| --- | --- |
| CATT | Acceptable:512Preferred:512 |
| Huawei | 512.We think that a larger value is needed, especially in scenario that RRH is widely used. If a DU manages 512 cells which all are RRHs, in worse case, each cell in the DU may suffer PRACH configuration conflict from cells in neighbour nodes. |
| Nokia | 512 on both cases. In our view, if a DU can serve up to 512 cells, a CU should be able to provide with a single message to each of those cells neighbour PRACH Configurations. Of course, this is to be understood as a maximum value that may not be often needed. Other values (256, 128, etc.) are speculative that are “big enough” but no one can really be sure otherwise what the maximum needed value may be.  |

## Whether to support “pull” mechanism

In this meeting one company proposed that the gNB-DU should be capable to request the gNB-CU to provide PRACH related information [1].

Although its motivation is that the limit of 512 may be too low for some cases, making the gNB-DU has to retrieve more in a “segmented” manner (something similar to delivering a lengthy *RRCReconfiguration* message), the moderator believes that it can be used in other ways.

In summary, the moderator withes to confirm 2 major usages of such “pull” mechanism:

1. The gNB-CU may not deliver all of the neighbour PRACH configurations due to e.g. message size, but then the gNB-CU does not deliver more (i.e. contrast to the ordinary “segmentation” mechanism where multiple F1AP message should be sent automatically without any inserted feedback). If any PRACH conflict is seemingly still unresolved, the gNB-DU should trigger the “pull” mechanism. In order to support this usage, the gNB-DU must store the received PRACH configuration for a while at least.
2. The gNB-CU does not always “push” neighbour PRACH configurations when the gNB-DU sets up a new cell. This make the gNB-DU has to “pull” neighbour PRACH configurations.

**Questions 2.2-1**: What is your opinion on the “pull” mechanism? And what is your opinion on the two usages?

| Company | Comment |
| --- | --- |
| CATT | Neutral.Either supporting or not supporting the “pull” mechanism is acceptable for us. And either usage is acceptable for us as well. |
| Huawei | Neutral.Maybe useful, only if the CU does an over-filtering which leads to the DU having insufficient neighbouring cells’ configurations.- |
| Nokia | We do not support the “pull” mechanism. In our view, the gNB-DU can resolve RACH Configuration conflicts locally without the need of additional information. If a DU is given enough PRACH Configurations by the CU then it can resolve conflicts without the need of the additional message exchange to retrieve those.  |

# Discussion (first phase)

In this meeting we collected 6 documents on this agenda item from 4 companies [1–6], all of which focused on PRACH coordination.

How to perform PRACH coordination in gNB-CU/DU split architecture (and for EN-DC as well maybe) is one open issue left over from Rel-16. The progress we achieved last meeting makes the moderator kind of optimistic that we can finish this topic within Rel-17—maybe no one wishes delaying it once again into Rel-18.

But the limitation on time worries the moderator: there are only three meetings left (including this one) while quite an amount of Stage 3 details may need to be determine. In order to make progress as soon as possible, the moderator sincerely wishes every participant to compromise with one another.

The moderator understands that anyone is not so glad to change their opinion by a U-turn. So the moderator encourages using a “softer” manner when filling the blanks, e.g. “Preferred” / “Not preferred (but acceptable)” / “Not acceptable”, rather than a simple Boolean “Agree” vs “Disagree”.

## Confirm the WA last meeting

At first we would like to confirm the WA we achieved last meeting:

WA1: The gNB-CU should be possible to provide the gNB-DU with information indicating the CGI of the cells potentially in conflict and the neighbouring relation between these cells and their neighbour cells, along with the PRACH configurations of those neighbour cells, so as to prevent the gNB-DU from reconfiguring one of its cells from conflicting with one neighbour toward conflicting with another neighbour. How/whether gNB-CU do the filter is up to implementation.

One company proposed to turn it directly into an agreement [1], while another company proposed [4]:

|  |
| --- |
| With respect to the WA agreed, the neighbour cells for which the gNB-CU signals PRACH configurations to the gNB-DU are neighbors of the gNB-DU cells in potential PRACH conflict . |

The moderator thinks the sentence in [4] is literally covered by the WA, and thus simply confirming the WA is sufficient enough.

**Questions 1.1-1**: Do you agree to confirm the WA achieved last meeting?

| Company | Comment |
| --- | --- |
| CATT | Agree. |
| Huawei | Agree. |
| Nokia | Agree. |
| Qualcomm | Agree. Anyway, the filtering is up to implementation as captured in stage-2. |
| Ericsson | Agree.Our proposal in [4] was to make the agreement clear and not subject to interpretations. Anyhow we are ok with the WA. One editorial correction needed should be the following:“It should be possible for the gNB-CU to provide…” |
| Samsung  | Agree |
| ZTE | Agree.  |
| CMCC | Agree |

**Moderator’s summary**:
8 companies provided feedback.
8 agreed but one suggested an editorial change.

**Moderator’s Proposal:** **It should be possible for the gNB-CU to provide the gNB-DU with information indicating the CGI of the cells potentially in conflict and the neighbouring relation between these cells and their neighbour cells, along with the PRACH configurations of those neighbour cells, so as to prevent the gNB-DU from reconfiguring one of its cells from conflicting with one neighbour toward conflicting with another neighbour. How/whether gNB-CU do the filter is up to implementation.**

## Cell list structure

Then we would like to continue the discussion on cell list structure. During second phase of the offline discussion last meeting, the moderator listed two possible approaches based on companies’ input [7], aiming to cover the abovementioned WA:

|  |  |
| --- | --- |
| **[Approach 1]**Neighbour item:>CGI: 21>Carrier list etc: XXX>PRACH Configuration: #1>Associated served cell list:>>CGI: 11Neighbour item:>CGI: 22>Carrier list etc: XXX>PRACH Configuration: #2>Associated served cell list:>>CGI: 11>>CGI: 12Neighbour item:>CGI: 23>Carrier list etc: XXX>PRACH Configuration: #3>Associated served cell list:>>CGI: 12 | **[Approach 2]**Served cell item / served-cell-specific message:>CGI: 11>Neighbour list:>>Neighbour item:>>>CGI: 21>>>Carrier list etc: XXX>>>PRACH Configuration: #1>>Neighbour item:>>>CGI: 22>>>Carrier list etc: XXX>>>PRACH Configuration: #2Served cell item / served-cell-specific message:>CGI: 12>Neighbour list:>>Neighbour item:>>>CGI: 22>>>Carrier list etc: XXX>>>PRACH Configuration: #2>>Neighbour item:>>>CGI: 23>>>Carrier list etc: XXX>>>PRACH Configuration: #3 |

Last meeting we failed to achieve any agreement on this topic (i.e. cell list structure). But the situation has slightly changed ever since: in this meeting two companies continued to express their preference on Approach 1 for F1AP [1][2], while the only company supported Approach 2 for F1AP last meeting switched to another method [4]:

|  |
| --- |
| Although the Approach 2 is simpler and more logical to be implemented, companies argue that Approach 1 is beneficial in term of signalling size/overhead, as PRACH configuration of a neighbouring cell may be forwarded to the gNB-DU multiple times in the Approach 2. However, in our understanding we can still use a version of Approach 2 that is simpler and understandable, while reducing the signalling overhead by making the PRACH configuration of the neighbouring cells as an optional IE. Based on this approach, a PRACH configuration of a neighbouring cells would not be duplicated (as it is optional) and it will only be added once for a given neighbour cell CGI. |

And another company also considered it acceptable [2]:

|  |
| --- |
| Nevertheless, there are also some other approaches which avoids such duplication as well, e.g. to deliver cell relation information separately from the configuration of neighbour cells, to omit the duplicated configuration on the basis of Approach 2 (the latter of which is in fact identical to what we propose for X2AP). We are generally open to these alternative approaches as long as they provide identical information. |

For convenience the moderator proposes to name this alternative version of Approach 2 as “Approach 2bis”. As no one supports the original version of Approach 2 now, what we need to do is to perform down-selection between Approach 1 and Approach 2bis:

|  |  |
| --- | --- |
| **[Approach 1]**Neighbour item:>CGI: 21>Carrier list etc: XXX>PRACH Configuration: #1>Associated served cell list:>>CGI: 11Neighbour item:>CGI: 22>Carrier list etc: XXX>PRACH Configuration: #2>Associated served cell list:>>CGI: 11>>CGI: 12Neighbour item:>CGI: 23>Carrier list etc: XXX>PRACH Configuration: #3>Associated served cell list:>>CGI: 12 | **[Approach 2bis]**Served cell item / served-cell-specific message:>CGI: 11>Neighbour list:>>Neighbour item:>>>CGI: 21>>>Carrier list etc: XXX>>>PRACH Configuration: #1>>Neighbour item:>>>CGI: 22>>>Carrier list etc: XXX>>>PRACH Configuration: #2Served cell item / served-cell-specific message:>CGI: 12>Neighbour list:>>Neighbour item:>>>CGI: 22 (All cell configuration information omitted)>>Neighbour item:>>>CGI: 23>>>Carrier list etc: XXX>>>PRACH Configuration: #3 |

**Questions 1.2-1**: Are Approach 1 and/or Approach 2bis acceptable for you? If both approaches are acceptable, which one do you prefer?

| Company | Comment |
| --- | --- |
| CATT | Acceptable: Both approach.We don’t have strong preference between the two approaches. Each approach has its own pros and cons: Approach 1 is cleaner while Approach 2 is easier to understand and more aligned with X2AP. |
| Huawei | Solution 1 is preferred. But can accept solution 2bis if the max no. is set to 512. Which means that at least each serving cell in DU may receive PRACH configuration of neighbour cells. This is the meaning of the maximum value. |
| Nokia | Solution 1 is preferred. Solution 2bis is not preferred (but acceptable). |
| Qualcomm | Both approaches are acceptable. Slightly prefer Option 1, but Option 2 is also acceptable.Agree with CATT that Option 2 is more easy to understand. But Approach 1 allows a natural filtering of the neighbour PRACH Configurations a CU may send to its DU. For example, if gNB-CU wants to send a filtered set of PRACH configurations of neighbouring cells (e.g., only to those cells in conflict), it can just send those neighbors whereas in Approach 2, it has to send PRACH configurations for every serving cell (or we have to limit the outer list to a small number like 32) |
| Ericsson | Approach 2b is preferred. Approach 2b is exactly the same structure used over Xn, hence easier to understand for those who will use the specifications. In reply to Qualcomm, in Approach 2b the list of neighbour per cell has negligible message size weight because PRACH configurations can be omitted, if already included elsewhere. Hence listing of neighbours per cell in conflict becomes less impacting (if at all)  |
| Samsung | Both ok for us. Slightly prefer Approach 1.  |
| ZTE | We prefer definitions that are easy to understand, so we prefer solution 2bis. (solution 1 is also acceptable). |
| CMCC | Both are fine with us |

**Moderator’s summary**:
8 companies provided feedback.
7 accepted Approach 1, whereas 1 did not express clearly.
8 accepted Approach 2bis but one among them only accepted with condition.
4 preferred Approach 1.
2 preferred Approach 2bis.

## Maximum length of cell lists

And the next question is on the maximum limit of length of the abovementioned cell lists.

Since the cases for the two different approaches are different, the moderator suggests discussing them separately—and so do the two layers of the cell lists respectively.

NOTE: In order to make progress as much as possible, companies are expected to provide their feedback on both approaches regardless of their preference between the two approaches.

### In Approach 1

For the outer layer of cell list in Approach 1, i.e. list of neighbour cells, both companies proposed to define its maximum length as 512 [1][2]. And in the moderator’s understanding, another company also preferred so, although they did not clearly express their support on Approach 1 [5]. The moderator believes that this is a very old topic and companies are expected to be familiar with the reasons provided by one another, and thus no need to copy the reasons here.

**Questions 1.3.1-1**: If Approach 1 is adopted, what value should the maximum (i.e. upper bound) length of neighbour cell list (i.e. the outer layer) be defined as? Please provide the range of that maximum value you considered acceptable and your most preferred value.

| Company | Comment |
| --- | --- |
| CATT | Acceptable: 512 or so.Preference: exactly 512.RAN3’s tradition is to define the max length as the max possible value regardless of the probability (e.g. we defined the max number of cell groups as 4, even though it is still not supported). And in practice when the gNB-DU is just setup all of its served cells are potentially in PRACH conflict, having every intra-frequency neighbour’s PRACH configuration be delivered. |
| Huawei | 512 |
| Nokia | 512 (we agree with CATT). |
| Qualcomm | Acceptable: 512 or any lower value (if there are concerns on message size) |
| Ericsson | We prefer a value lower than 512, for example 128. |
| Samsung | 512 |
| ZTE | 512 |
| CMCC | 512 |

**Moderator’s summary**:
8 companies provided feedback.
7 accepted 512, whereas 1 did not express clearly.
2 accepted 128, whereas 4 did not express clearly. 2 did not accept 128.
6 preferred 512.
1 preferred 128.

While for the inner layer of cell list in Approach 1, i.e. associated served cell list, the two companies showed different opinion. One company thought the maximum length should be 32 [1] while another proposed 512 [2]. The reason for defining it as 32 is not provided in [1], while the reason for 512 is provided in [2] as below:

|  |
| --- |
| the neighbour cell is a macro-cell but may of the served cells are pico-cells sharing the same spectrum with the macro-macro cell (see in Figure 2 as well). Since the meaning of the length of the inner layer list is “how many served cells neighbour to the neighbour cell”, it can be several hundred. The most extreme case is that all of the cells served by the gNB-DU are within the cover of a huge macro cell, making the maximum length 512. |

**Questions 1.3.1-2**: If Approach 1 is adopted, what value should the maximum (i.e. upper bound) length of associated served cell list (i.e. the inner layer) be defined as? Please provide the range of that maximum value you considered acceptable and your most preferred value.

| Company | Comment |
| --- | --- |
| CATT | Acceptable: any value from 32 to 512.Preference: exactly 512.The reason is already quoted by the moderator. But in order to make progress we can accept a much smaller value. |
| Huawei | For approach 1, it indicates the no. of cells in the DU that is neighbouring to a neighbour cell. Although we are also OK to 512, but in reality, a cell is not likely neighbouring to all the cells in a DU? A smaller value may also address the concern on message size from other company although it’s questionable. |
| Nokia | Preferred: 512Even though we agree that it may not always happen that all 512 served cells are in RACH Conflict, we should be able to signal this in a situation when this is necessary. |
| Qualcomm | Preference: 32 (as the reason pointed by Huawei; there can’t be 512 cells neighboring to a neighbor cell).Acceptable: any other value from 32 to 512 |
| Ericsson | Strong preference for 32. As Huawei and Qualcomm point out, there will be a limited number of neighbouring cells sharing the same spectrum. If it was not so, interference issues would become unbearable. On top of this, there will be even less neighbouring cells sharing the same spectrum AND in RACH conflict. |
| Samsung | 512. Agree with Nokia. We should provide enough space for all cases.  |
| ZTE | Preference: 32Acceptable: any other value from 32 to 512 |
| CMCC | Preference: exactly 512.Acceptable: any value from 32 to 512. |

**Moderator’s summary**:
8 companies provided feedback.
7 accepted 512, whereas 1 did not express clearly.
6 accepted 32, whereas 2 did not express clearly.
4 preferred 512.
4 preferred 32.

### In Approach 2bis

For the outer layer of cell list in Approach 2bis, i.e. the list of cells potentially in PRACH conflict, the company proposed to define its maximum length as 32 [4]. The reason is provided as following:

|  |
| --- |
| In addition, we think that signalling up to 32 serving cells in potential PRACH conflict in one message would suffice to convey all possible and potential conflicts to the gNB-DU. In case of any additional cells in conflict need to be signalled, a new message can be initiated. |

**Questions 1.3.2-1**: If Approach 2bis is adopted, what value should the maximum (i.e. upper bound) length of potentially-in-conflict served cell list (i.e. the outer layer) be defined as? Please provide the range of that maximum value you considered acceptable and your most preferred value.

| Company | Comment |
| --- | --- |
| CATT | Acceptable: exactly 512.RAN3’s tradition is to define the max length as the max possible value regardless of the probability (e.g. we defined the max number of cell groups as 4, even though it is still not supported). And in practice when the gNB-DU is just setup all of its served cells are potentially in PRACH conflict. So this value should be identical to the max value of served cells in gNB-DU. |
| Huawei | Prefer 512 for the upper list for 2bis. 32 is obviously not enough. Considering that a DU hosts 512 cells, how could only 32 cells are in the boundary of the coverage of the DU? |
| Nokia | Preferred: 512Acceptable: 512 (same reasons described by HW) |
| Qualcomm | Either we define 512 and ask CU to send a limited list if desired or only allow a smaller number like 32. Either is fine. |
| Ericsson | We prefer 32. We could accept a higher number such as 128. We would like to point out that having a maximum list of cells in RACH conflict equal to 512 implies the acknowledgement of a use case where ALL cells served by a gNB-DU are in conflict! This would imply that no planning at all has been carried out. Indeed this looks like someone planned a network to make the cells conflict on purpose 😊We do not deny that there could be many cells in conflict, but if they are the list should be smaller than 512…In light of this, the logic in Huawei´s comment is a bit misleading because it let´s us believe that ALL the cells of a gNB-DU that are neighbouring with co-channel sharing cells should be reported. Instead, only the (hopefully small) number of cells in RACH conflict should be reported.We believe that in a well planned network PRACH conflicts will be very few. In a badly planned network, PRACH conflicts should not exceed 32 per DU. We cannot take into account as a reference scenario a network where no planning at all is carried out and where all 512 cells served by a gNB-DU are in conflict.  |
| Samsung  | 512 |
| ZTE | Acceptable: 512 (same reasons described by HW) |
| CMCC | 512, same reason as HW |

**Moderator’s summary**:
8 companies provided feedback.
7 accepted 512. 1 did not accept 512.
2 accepted 32 or 128, whereas 1 did not express clearly. 5 did not accept 32 or 128.
6 preferred 512.
1 preferred 32.

For the inner layer of cell list in Approach 2bis, i.e. the list of associated neighbour cells, the company proposed to define its maximum length as 32 [4]. The reason is provided as following:

|  |
| --- |
| it is proposed that the list of neighbour PRACH configurations includes 32 PRACH configurations, because such number is sufficiently large to include PRACH configurations of neighbouing cells. Signalling more PRACH configurations would imply a substantial increase of message size without any obvious benefit. It is in fact extremely unlikely that a cell has more than 32 neighbouring cells on the same operating frequency. |

**Questions 1.3.2-2**: If Approach 2bis is adopted, what value should the maximum (i.e. upper bound) length of associated neighbour cell list (i.e. the inner layer) be defined as? Please provide the range of that maximum value you considered acceptable and your most preferred value.

| Company | Comment |
| --- | --- |
| CATT | Acceptable: at least 16.Preference: exactly 32.We agree with Ericsson’s analysis quoted by the moderator. The case for this question is different from the one for Q1.3.1-2:If the served cell is a macro one neighbouring to a lot of intra-frequency pico-cells, we should change the PRACH configuration of pico-cells to avoid conflict as this is easier (i.e. far less neighbours than the macro cell). As the result, there is no need to deliver those neighbouring pico-cells’ PRACH configurations toward the macro served cell. |
| Nokia | Preferred: 32In this scenario we can consider that the number of possible intra-frequency neighbours of a given served cell is limited to 32 for all practical purposes.  |
| Qualcomm | Preferred: 32 |
| Ericsson | Preferred 32 |
| Samsung | Here there should be same logic with Approach 1. As the sample in Approach 1, DU 1 has 512 pico-cells sharing the same spectrum with a macro cell of neighbor DU 2. If Approach 1, on F1 interface of DU 1, inner layer should be 512 because the served cells is 512. If Approach 2bis, on F1 interface of DU 2, inner layer should be 512 also because the neighbor cells is 512. If companies agreed the number of possible intra-frequency neighbours is up to 32, inner layer in Approach 1 should be limited to 32 as well. We hope to provide the enough information to avoid that DU can ask more PRACH configuration if DU think it’s not enough, so we prefer 512. |
| ZTE | Preferred: 32 |
| CMCC  | Preferred: 32 |

**Moderator’s summary**:
7 companies provided feedback.
2 accepted 512, whereas 5 did not express clearly.
6 accepted 32, whereas 1 did not express clearly.
1 preferred 512.
6 preferred 32.

## Other issues (low priority)

There are also some other valid issues over PRACH coordination within companies’ documents, either in the form of proposals or only shown in the TP. In order to save time the moderator only selects 2 “general” issues to be discussed. The issues on detail IE design can be discussed after we determine the cell list structure.

NOTE: Even for the questions listed below, “to wait the output of other questions first” is also considered to be a reasonable feedback by the moderator.

### Can an F1 Setup Response message contain information related to PRACH coordination?

This issue has been discussed many times and fortunately companies’ view on it seemingly has softened a lot ever since. In this meeting two companies seemingly preferred “yes” [1][2] while two others preferred “no” [4][6].

**Questions 1.4.1-1**: Are the answer “yes” and/or “no” acceptable for you, over the issue whether an F1 Setup Response message can contain information related to PRACH coordination? If both answers are acceptable, which one do you prefer?

| Company | Comment |
| --- | --- |
| CATT | Acceptable: any of “yes” or “no”.Preference: “yes”.We don’t think this an essential issue…but we have to make a decision anyhow. |
| Huawei | No strong view. |
| Nokia | Preferred: no,Not preferred (but acceptable): yes, or no In order to support progress and conclude on this topic, we can support either option. |
| Qualcomm | No strong view. |
| Ericsson | Preferred “no”. We do not want to increase the size and therefore the vulnerability, of the F1 Setup procedure. We had issues with the F1 Setup message size in the past and these issues are likely to come back if we keep on adding info to these messages. Without a successful F1 Setup, cells cannot be turned operational, so this is a major upset we would like to avoid. |
| Samsung | No strong view. |
| ZTE | Preference: “yes”. (but “no” is acceptable). |
| CMCC | No strong view |

**Moderator’s summary**:
8 companies provided feedback.
7 accepted “yes”, whereas 1 did not express clearly.
8 accepted “no”.
2 preferred “yes”.
2 preferred “no”.

### Should the gNB-DU store the received information on neighbour cells?

This issue has not been discussed yet. From the TPs of 2 companies the answer seemed to be “yes” or at least “up to implementation” [1][2].

The moderator believes three answers are reasonable: “always yes”, “up to implementation” and “always no”. The network behaviour should be different depends on this answer in order to prevent inter-vendor operation problem, e.g.:

* If the answer is “always no” the gNB-CU should always send the entire list once upon the addition of a served cell or a neighbour cell, or the change of PRACH configuration, while if the answer is “always yes” the gNB-CU can send only the “delta” part.
* If the answer is “up to implementation”, it may be beneficial to introduce a method so that the gNB-DU can request the gNB-CU to provide neighbour’s PRACH configuration, as proposed in [1], although this IE may also have other use.

**Questions 1.4.2-1**: Are the answer “always yes”, “always no” and/or “up to implementation” acceptable for you, over the issue whether the gNB-DU should store the received information of neighbour cells? If all answers are acceptable, which one do you prefer?

| Company | Comment |
| --- | --- |
| CATT | Acceptable: any of “always yes”, “always no” or “up to implementation”.Preference: “always yes”.We believe the answer is “always yes” for X2AP and XnAP. Here’s why:If a gNB-CU does not store the PRACH configurations it received from neighbours, when a new gNB-DU is deployed and connected toward this gNB-CU, the gNB-CU will have no means to provide the neighbour cell’s PRACH configuration toward this new gNB-DU (there is currently no “PRACH config retrieval” over X2AP or XnAP).So the gNB-CU has to store them.And letting F1AP to align with X2AP/XnAP seems reasonable. The cost is usually negligible as analysed in [2]. |
| Huawei | The answer is of course “always yes”? Because the following are specified in F1AP gNB-CU configuration update”*“The updated configuration data shall be stored in the respective node and used as long as there is an operational TNL association or until any further update is performed.”*And a question, does the current F1AP support delta configuration to DU now?The proposal from 1 is for the case when the DU cannot resolve the PRACH conflict issue based on neighbour cells PRACH configurations received in gNB-CU configuration update. |
| Nokia | Always yes. |
| Qualcomm | Always yes (as pointed by Huawei). But this has no spec impacts right? |
| Ericsson | This can be left to implementation. We do not accept an “Always yes” because there is no point for a gNB-DU to maintain a large neighbour context given that a gNB-DU is not in charge of mobility or indeed in charge of any operation that requires coordination with neighbour clels (outside its served cells). PRACH conflict is a punctual event that can be resolved on the spot. After that neighbour information can be discarded.Agreeing in the specifications that the gNB-DU shall always store neighbour information would imply to place unnecessary implementation requirements on the gNB-DU. |
| Samsung | Always yes.  |
| ZTE | Always yes (as pointed by Huawei). |
| CMCC | Always yes. |

**Moderator’s summary**:
8 companies provided feedback.
7 accepted “always yes”. 1 did not accept “always yes”.
2 accepted “up to implementation”, whereas 3 did not express clearly. 3 did not accept “up to implementation”.
7 preferred “always yes”.
1 preferred “up to implementation”.

**Moderator’s Proposal: The gNB-DU should store the received information on neighbour cells (this has impact on Section 8.x of specs).**

# Conclusion, recommendations [if needed]

# Reference

[1] R3-214864; (TP for SON BL CR for TS 38.473): Left overs on RACH Optimization Enhancements; Huawei.

[2] R3-215053; Discussion on Rel-16 leftover issues for PRACH coordination; CATT.

[3] R3-215054; (TP on SON for 36.423) TP on PRACH coordination for X2AP; CATT.

[4] R3-215447; (TP for SON BL CR for TS38.473-g70): RACH conflict resolution procedure; Ericsson.

[5] R3-215475; Further discussion on down-selection of options for RACH optimization; Nokia, Nokia Shanghai Bell.

[6] R3-215476; (TP for SON BL CR for TS 38.473) Enhancement of RACH Conflict Resolution; Nokia, Nokia Shanghai Bell.

[7] R3-214333; CB: # SONMDT6\_RACHOpt; CATT (moderator).