3GPP TSG-RAN3 Meeting #109-E R3-205993

E-meeting, 17 – 28 August 2020

Agenda Item: 9.3.5.1

Source: Ericsson

Title: SoD for MaxNofConnectedCells

Document for: Discussion, Decision

# Introduction

This is a summary of offline discussions for the topic of Cell Creation Rejection when max number of supported cells is exceeded at CU.

**CB: # 71\_MaxNofConnectedCells**

**- Acknowledge problem?**

**- Cause value agreeable?**

**- check details**

(E/// - moderator)

Summary of offline [R3-205593](C:\\Users\\eangcen\\Documents\\3GPP_ETSI\\RAN3\\RAN3-109\\EmailDiscussions\\Inbox\\R3-205593.zip)

## Summary of First Round of Discussions

There seem to be a general support for the problem statement which is formulated as follows:

**Problem Statement:** **3GPP specifications allow a gNB-DU to connect to any available gNB-CU. It is possible that the cells a gNB-DU wants to add at a gNB-CU causes that the maxium number of cells a gNB-CU can add is exceeded. The problem to solve is how to identify such failure and how to prevent that it would occur in the future.**

Solutions were discussed and one option found some company support:

**Solution Proposal: Adding a cause value in the F1 SETUP FAILURE message and in the GNB-DU CONFIGURATION UPDATE FAILURE message, which specifies that a failure occurred because the gNB-CU Cell Capacity was Exceeded**

# For the Chairman’s Notes

Following agreements were proposed on the first round of offline discussion:

Minute that the problem stated below is valid:

**Problem Statement:** **3GPP specifications allow a gNB-DU to connect to any available gNB-CU. It is possible that the cells a gNB-DU wants to add at a gNB-CU causes that the maxium number of cells a gNB-CU can add is exceeded. The problem to solve is how to identify such failure and how to prevent that it would occur in the future.**

Agree to the following solution:

**Solution Proposal: Adding a cause value in the F1 SETUP FAILURE message and in the GNB-DU CONFIGURATION UPDATE FAILURE message, which specifies that a failure occurred because the gNB-CU Cell Capacity was Exceeded**

# A CR capturing this solution is available in R3-20xxxxDiscussion

## Problem description

The problem raised in R3-204780 is described in the figure below:



The standard does not mandate that a gNB-DU needs to connect to the same gNB-CU every time it wants to setup an F1.

A simple and very plausible example where one cannot predict that a gNB-DU would connect to the same gNB-CU is when an ongoing connection between gNB-DU and gNB-CU fails, e.g. due to a failure at the gNB-CU.

In this case the gNB-DU is allowed to setup a connection to any other available gNB-CU. If this was not possible the system would be highly unreliable because at the failure of one gNB-CU, the cells of a gNB-DU would never be able to become operational.

**Observation 1: 3GPP specifications allow a gNB-DU to connect to any available gNB-CU. If this was not possible the system would be highly unreliable because at the failure of one gNB-CU, the cells of a gNB-DU would never be able to become operational.**

With the clarification above in mind, it is plausible that a gNB-DU would connect to a gNB-CU and in that case that the cells to be added at the gNB-CU would exceed the maximum number supported by the gNB-CU. It has to be noted that the limit in number of cells a gNB-CU has is implementation specific and could be below the current maximum of 16384 cells.

**Observation 2: Given that a gNB-DU can connect to any available gNB-CU, it is possible that the cells a gNB-DU wants to add at a gNB-CU causes that the maxium number of cells a gNB-CU can add is exceeded.**

The main problem R3-204780 wants to resolve is how to identify that a failure is due to the issue above and how to avoid that this problem occurs in the future.

**Problem Statement:** **3GPP specifications allow a gNB-DU to connect to any available gNB-CU. It is possible that the cells a gNB-DU wants to add at a gNB-CU causes that the maxium number of cells a gNB-CU can add is exceeded. The problem to solve is how to identify such failure and how to prevent that it would occur in the future.**

Companies are invited to consider the problem describe above and to state whether the problem is acknowledged or whether, if this problem is not recognised, how it can be solved.

|  |  |  |
| --- | --- | --- |
| Company | Agree/Don’t Agree | Comments |
| Nokia | Generally not agree. | In our view there are two aspects to be considered.  \* Firstly, whether the max number of cells that has been specified for a gNB-CU in current specifications is adequate. Considering a cloud deployment, it is foreseeable that the current maximum number can be eventually exceeded. Such examples could be small cells deployments. Likewise, in RAN sharing deployments, a different cell identifier may be used (under discussion). With that under consideration, the maximum number of cells could be reached. However, the solution for such case is simple, and can be limited to just to increase the max number of cells from 14-bit to e.g., 17-bit that a gNB-CU can support in the specifications. This would also would remove this possible issue in long term.  **Proposal 1: Increase max # cells to be supported at gNB-CU to 17-bits.**  \* Secondly, there seems to be concern from some companies that in some deployments (likely non cloud based), the gNB-CU could run out of capacity to handle certain number of cells (let it be around 16 thousand, as with 14-bit limit, or a lower value). For this case, introducing a new cause value to indicate the issue is sufficient to address the concern. Nevertheless, we see this as a corner case.  **Proposal 2: Introduce new cause value over F1 to indicate that the gNB-CU has reached it s maximum cell handling value.**  Other enhancements we see as not needed. Likewise, as it was discussed at prior RAN3 meetings, for the second aspect, it is expected that proper dimensioning and OAM can handle this scenario.  **Proposal 3: Any changes, if any, should target Release 16.** |
| ZTE | Agree | We mentioned in the previous email that this problem is a deployment problem. For example, an alarm system can be used to warn in advance to prevent this from happening.  Due to the co-signature of the operator, we realize that this problem needs to be resolved.  For example, in some IAB scenarios, there may be very flexible DU deployment. |
| Ericsson | Agree | As explained it seems difficult to avoid the problem via planning, if flexibility wants to be maintained. We believe that at least a solution relying on cause values should be made available. This is the simplest solution, which will also allow for visibility at OAM level, so this would be in line with the “alarm system” solution approach from ZTE, but with reduced complexity |

## Possible Solutions

The solution proposed in R3-205523 to solve the problem above is made of two parts:

* **Solution part relative to problem detection**: this consist of adding a cause value in the F1 SETUP FAILURE message and in the GNB-DU CONFIGURATION UPDATE FAILURE message, which specifies that a failure occurred because the gNB-CU Cell Capacity was Exceeded. This element of the solution allows to have visibility over this type of failure and to determine any possible action to prevent a posteriori the failure in the future.
* **Solution part relative to problem prevention**: this consists of signalling in the GNB-DU CONFIGURATION UPDATE FAILURE message the maximum number of cells that are left to be added at the gNB-CU. This element of the solution allows to prevent the failure after it occurs. Namely, a gNB-DU that immediately re-attempts to connect to the gNB-CU would not generate a failure anymore, because it will be able to add a number of cells within the gNb-CU capacity.

Companies are invited to provide their comments to the solutions components above and to highlight how the problem would be solved if the solutions component described are not in place.

|  |  |  |
| --- | --- | --- |
| Company | Solution | Comments on solution |
| Nokia |  | See comment on section 3.1 |
| ZTE |  | 1: We still believe the solution is sub-optimal and introduce unnecessary complexity.  For example, when a CU is connected to multiple DUs, when a DU tries to connect and gets a failure message, the message informs that DU has 20 cells remaining. But other DUs will also try at the same time, for example, try to access 5 cells. Once the DU attempt is successful, there is a margin of 15 Cells left in the CU. When the DU that obtained 20 cells before is ready for select 20 cells to try again, at this time, there are only 15 cells left in the CU, so the DU fails to access again. If multiple DUs try to access at the same time, DU and CU will be in a 1-to many ping-pong-like scenario.  2: Secondly, it is the same as the issue we mentioned in the discussion at the last meeting. Whether the DU can intelligently select the cell to be coverage according to the margin number of CUs. Regardless of the choice, DU needs to discard some cell configurations. The DU may not so smart to distinguish whether the DU cell is used for coverage or offload.  3: Therefore we suggest that this problem can be considered and solved in the AI topic. With machine learning, maybe the issue can be solved better. |
| Ericsson |  | We would support both solutions components. However we can accept to have the addition of a new cause value as a problem detection mechanism |

# Conclusion, Recommendations [if needed]

If needed