**3GPP TSG RAN WG1 #102-e- R1-200xxxx**

**e-Meeting, August 17th – 28th, 2020**

Source: moderator (vivo)

Title: Feature lead summary on ULFPTx

Agenda Item: 7.2.6

Document for: Discussion and Decision

1. Introduction

In this contribution, following issue is discussed according to agreement of preparatory email discussion

* [102-e-NR-eMIMO-07] TPMI grouping for mode 2 – Rakesh (vivo)

1. Remaining issues
   1. Issue 1: TPMI grouping for Mode 2

Alt1. Add new TPMI group(s)

Alt2. No change in TPMI group, and UE can report >1 TPMI groups

Alt3. Revise at least one TPMI group

|  |  |
| --- | --- |
| Company | Comments |
| ZTE | There are two questions ahead of us:   * Q1: The number of PA architectures corresponds to non-coherent and partial-coherent case is different, so how to make it in equal? * Q2: In order to enable the diversity of UE implementation with this functionality, how to support more newly typical PA architectures?   Here, we would like to provide a solution to address this issue, which consist of the following two aspects.  For Q1, compared to the existing G0-G6 for partial-coherent 4-port UEs, two PA architectures have been missed for non-coherent 4-port UEs, which are {20 17 20 17} dBm and {20 20 20 20} dBm. Correspondingly, two new groups (G4’, G5’) correspond to the two missed PA architectures should be added to Table 1 in RAN1 #99 agreement.  **Table 1’**   |  |  | | --- | --- | | 4Tx, nonCoherent | 4Tx, partialCoherent | | G0 | G0 | | G1 | G1 | | G2 | G2 | | G3 | G3 | | G4’ | G4 | | G5’ | G5 | |  | G6 |   **<Precoders of G4’ and G5’ correspond to non-coherent 4-port UEs>**   |  | | --- | | G4’:  G5’:{, , , , , }, |   **<PA architectures of Table 1>**   |  |  |  |  | | --- | --- | --- | --- | | 4Tx, nonCoherent | PA architecture | 4Tx, partialCoherent | PA architecture | | G0 | [23 17 17 17] dBm | G0 | [23 17 17 17] dBm | | G1 | [23 17 23 17] dBm | G1 | [23 17 23 17] dBm | | G2 | [23 23 23 17] dBm | G2 | [23 23 23 17] dBm | | G3 | [20 20 20 17] dBm | G3 | [20 20 20 17] dBm | | G4’ | [20 17 20 17] dBm | G4 | [20 17 20 17] dBm | | G5’ | [20 20 20 20] dBm | G5 | [20 20 20 17] dBm | |  |  | G6 | [20 20 20 20] dBm |   For Q2, based on the newly added G4’ and G5’ for non-coherent 4-port UEs, Alt2 can be regarded as a good way to address this issue, meanwhile it retains the flexible scalability of the PA architectures. Specifically, for the partial-coherent 4-port UE, it can report up to two TPMI groups, one from {G0~G2} and one from {G3~G6}. For the non-coherent 4-port UE, it also can report up to two TPMI groups, one from {G0~G2} and one from {G3, G4’, G5’}. We echo some examples as follow to elaborate this solution.   * Report G1+G4’ can enable the non-coherent 4-port UE with PA={23 17 23 17} dBm to implement full power with this functionality. * Report G1+G4 can enable the partial-coherent 4-port UE with PA={23 17 23 17} dBm to implement full power with this functionality. * Report G2+G5’ can enable the non-coherent 4-port UE with PA={23 23 23 17} dBm to implement full power with this functionality. * Report G2+G5 can enable the partial-coherent 4-port UE with PA={23 23 23 17} dBm to implement full power with this functionality. * Report G1+G5' can enable the non-coherent 4-port UE with PA={23 20 23 20} dBm to implement full power with this functionality. * Report G1+G6 can enable the partial-coherent 4-Tx UE with PA={23 20 23 20} dBm to implement full power with this functionality. * etc.   Due to the joint-reporting of TPMI groups, this solution can flexibly to enable many PA architectures of non-coherent & partial-coherent 4-port UEs to implement full power UL transmission. In addition, plenty of entries also have been remained to support some other PA architectures of non/ partial-coherent 4-port UEs in the future.  In conclusion, the intention of our solution is to ensure the number of supported PA architectures of non-coherent and partial-coherent 4-port UEs is equal, and enable more other PA architectures for the diversity of UE implement. |
| Samsung | Thanks ZTE for the proposal. Although a solution based on Alt1+2 is not our preference, but for progress, we can be fine with it. It does provide more flexibility to non-cerement UEs by including G4’ and G5’. |
| Intel | Firstly, we think that if we use the existing TPMI groups for Mode 2, Alt 2 should be adopted for the 4-port partial coherent UE.  Then regarding ZTE’s proposal, we understand that more PA architecture can be supported. However, the singling design is more complicated and confusing since for non-coherent UE it also needs to report one group from the partial coherent TPMI group table.  The issue with the existing TPMI groups is that in the partial coherent TPMI table, both non-coherent and partial coherent TPMI groups are included.  Therefore, we have the following proposal. In our proposal, it can support the PA architectures supported by ZTE’s proposal. And the singling is clearer.  Note in the Table 1 and Table 2 below, for G(N, x), ‘N’ means non-coherent. For G(P, x), ‘P’ means partial coherent.  **Table 1: 4-port Non-coherent TPMI groups supporting full power in Mode 2**   |  |  |  | | --- | --- | --- | | # | TPMI Groups | Note | | G(N,0) |  | G(N,0) to G(N,3) are the same as G0 to G3 | | G(N,1) |  | | G(N,2) | , | | G(N,3) | , | | G(N,4) |  | G4’ | | G(N,5) | , | G5' | | G(N,6) | , | G0+G3 | | G(N,7) | , | G1+G3 | | G(N,8) |  | G0+G4’ | | G(N,9) | , | G0+G5’ | | G(N,10) | , | G1+G5’ | | G(N,11) | , | G2+G5’ |   **Table 2: 4-port Partial-coherent TPMI groups supporting full power in Mode 2**   |  |  | | --- | --- | | # | TPMI Groups | | G(P,0) |  | | G(P,1) |  |   **TPMI reporting:**   * For 4-port partial coherent UE   + It can report one non-coherent group from Table 1 {G(N,0) ~ G(N,11)} and one partial coherent group from Table 2 {G(P,0) ~ G(P,1)}.   + It can also report 2-port non-coherent group {2-bit bitmap} * For 4-port non-coherent UE   + It can report one non-coherent group from Table 1 {G(N,0) ~ G(N,11)}   + It can also report 2-port non-coherent group {2-bit bitmap} * For 2-port non-coherent UE   + It can report 2-port non-coherent group {2-bit bitmap} |
| QC | Regarding ZTE’s solution, we are not sure we follow the rationale to add G4’ and G5’. Maybe G4’ is added to make sure noncoherent UE can use [20 17 20 17] dBm. So when a partial coherent UE report it can support G4, it automatically fall back to support G4’ when this UE fall back to noncoherent UE? But why add G5’? The underlaying PA assumption is different for noncoherent and partial coherent, and fallback from partial coherent to noncoherent does not work anyway.  Regarding Alt 2, it is against the design philosophy of the TPMIs in this list. The original design concept of these TPMIs is that UE report a single entry in the table, not a combination of entries using a bit map. If we go with Alt 2, which is essentially the bit map approach, we again should redesign the TPMIs in the list by treating each entry as an atomic TPMI set or a building block TPMI set.  Again, the correct way to fix this problem is to modify the existing TPMI table, i.e. Alt 3. We don’t see the motivation to take other routes to fix an existing bug by introducing more new bugs.  If we take the Alt 3 to modify the TPMI list, then the design is very simple. We just need to identify the typical PAs combination UE want to support. For each PAs combination, list all the full power TPMIs that PAs combination can support, assuming this UE is a full coherent UE. Then if this UE report it as a partial or noncoherent UE, gNB just cross out certain TPMIs in the reported TPMI set. In this way, the TPMI reporting signaling is simple and clean. And the answer to RAN2 question is: Yes, full power TPMIs for lower coherence can be deduced from higher coherence. I don’t see why RAN1 want to pick a problematic solution over this clean solution. |
| LG | Thanks ZTE and Intel for the proposals. In our view, more flexibility for supporting various UE implementation can be achieved by both solutions. However, it seems Intel’s proposal requires one more bit than ZTE’s one, if partial coherent UE reports both non/partial coherent TPMI groups. Thus, we slightly prefer ZTE’s proposal. |

* Answer to RAN2 LS: TPMIs for 4-port non-coherence can be deduced from the reported set of TPMIs for 4-port partial-coherence. 4-port full coherent UE follow the same way as 4-port partial coherent UE to report full power TPMIs.

|  |  |
| --- | --- |
| Company | Comments |
| ZTE | According to our above proposed solution of TPMIs reporting issue, we draft reply LS to RAN2 as below:  For a 4 port partial-coherent or full-coherent UE, the UE shall report 2-port {2-bit bitmap}, one 4-port non-coherent TPMI group from {G0~G2} in Table 1’ in Annex and/or one 4-port partial-coherent TPMI group from {G3~G6} in Table 1’ in Annex.  For a 4 port non-coherent UE, the UE shall report 2-port {2-bit bitmap}, one 4-port non-coherent TPMI group from {G0~G2} in Table 1' in Annex and/or one 4-port non-coherent TPMI group from {G3, G4', G5'} in Table 1’ in Annex.  For a 2 port UE, the UE shall report 2-port {2-bit bitmap}.  **Annex**  Table 1’   |  |  | | --- | --- | | 4Tx, nonCoherent | 4Tx, partialCoherent | | G0 | G0 | | G1 | G1 | | G2 | G2 | | G3 | G3 | | G4’ | G4 | | G5’ | G5 | |  | G6 |   Definition of G0~G6 and G4’, G5’ can be found in the table 2’ as below.  Table 2’   |  |  | | --- | --- | | TPMI group | Precoders | | G0 |  | | G1 | , | | G2 | ,, | | G3 | , | | G4 | , | | G5 | ,, | | G6 | ,,, | | G4’ |  | | G5’ | {,,,,, }, | |
| Samsung | Support with the clarification that for 4 ports, the UE reports x or y or (x, y) where x is a TPMI group from {G0-G2} and y is a TPMI group from {G3-G5} for NC or {G3-G6} for PC. |
| Intel | Same as our comment on Issue 1.  The TPMI reporting part is copied as below:  **TPMI reporting:**   * For 4-port partial coherent UE   + It can report one non-coherent group from Table 1 {G(N,0) ~ G(N,11)} and one partial coherent group from Table 2 {G(P,0) ~ G(P,1)}.   + It can also report 2-port non-coherent group {2-bit bitmap} * For 4-port non-coherent UE   + It can report one non-coherent group from Table 1 {G(N,0) ~ G(N,11)}   + It can also report 2-port non-coherent group {2-bit bitmap} * For 2-port non-coherent UE   + It can report 2-port non-coherent group {2-bit bitmap} |
| QC | Thank ZTE for formulating the reply to RAN2. Three questions for clarification. 1) why G3 is classified as a partial coherent precoder? Isn’t it a noncoherent precoder? 2) how many bits are needed to report 4-port non-coherent TPMIs for a 4-Tx UE? 3) A more important question is that: if we adopt this solution, how do we answer the RAN2 question in their LS, i.e., can full power TPMIs for lower coherence be deduced from higher coherence? Meaning, for a 4 Tx UE, can partial coherent full power TPMIs be deduced from full coherent full power TPMIs? Can noncoherent full power TPMIs be deduced from partial coherent TPMIs? I’d like to see the answer to this question with this proposed solution.  Same question 3) to Intel’s solution.  If we take the route to modify the TPMI list, then the design is very simple. We just need to identify the typical PAs combination UE want to support. For each PAs combination, list all the full power TPMIs that PAs combination can support, assuming this UE is a full coherent UE. Then if this UE report it as a partial or noncoherent UE, gNB just cross out certain TPMIs in the reported TPMI set. In this way, the TPMI reporting signaling is simple and clean. And the answer to RAN2 question is: Yes, full power TPMIs for lower coherence can be deduced from higher coherence. I don’t see why RAN1 want to pick a problematic solution over this clean solution. |
| LG | Support proposal from ZTE and also fine with Samsung’s clarification. |

1. Reference

[1] R1-2006979. “Summary#2 for Rel.16 NR eMIMO maintenance”, moderator (Samsung), RAN1#102-e, 17th -28th August, 2020