3GPP TSG-RAN2 Meeting #113-e R2-20xxxxx

Elbonia, 25 Jan - 05 Feb, 2021

**Agenda Item: 6.16**

**Source: Apple**

**Title: [Post112-e][063][NR TEI16] RRC processing time with segmentation**

**Document for: Discussion and Decision**

# Introduction

In RAN2#112e meeting, RAN2 agreed to extend the RRC processing time for RRC and to discuss the detailed solution via post meeting email discussion.

|  |
| --- |
| **RAN2 agreement**   * [029] Extending RRC processing time for RRC message segmentation is supported, to discuss detailed solution via long term email disc until next meeting. |

This is the email discussion report on below email discussion:

* [Post112-e][063][NR TEI16] RRC processing time with segmentation (Apple)

Scope: Make progress based on R2-2009488 and related discussion at R2 112-e.

Intended outcome: Report, agreeable CR

Deadline: long

According to the chair’s guidance, this report will be based on the summary R2-2010985 and try to figure out the majority interest on the proposals. The document consists of phase-1 and phase-2, the deadline of each phase is outlined as follow:

- **Phase-1:** collecting views on the detailed proposals, deadline: **Jan. 6, 2021.**

- **Phase-2:** collecting views on the summary and the text proposal, deadline: **Jan. 12, 2021.**

# Contact Information

|  |  |  |
| --- | --- | --- |
| **Company** | **Name** | **Email** |
| Apple | Fangli XU | fangli\_xu@apple.com |
| Xiaomi | Yumin Wu | wuyumin@xiaomi.com |
|  |  |  |
|  |  |  |
|  |  |  |

# Discussion

## Background

### Current RRC processing delay requirement

In current 38.331 section 12 defines the processing delay requirements for RRC reconfiguration/resume procedure as following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Procedure title:** | **Network -> UE** | **UE -> Network** | **Value [ms]** |
| **RRC Connection Control Procedures** | | | |
| RRC reconfiguration | *RRCReconfiguration* | *RRCReconfigurationComplete* | 10 |
| RRC reconfiguration (scell addition/release) | *RRCReconfiguration* | *RRCReconfigurationComplete* | 16 |
| RRC reconfiguration (SCG establishment/ modification/ release) | *RRCReconfiguration* | *RRCReconfigurationComplete* | 16 |
| RRC resume | *RRCResume* | *RRCResumeComplete* | 6 or 10 |

### The impact of RRC processing time by DL RRC segmentation

In [1], it is observed that RRC processing time is increased when RRC message size increases. Figure-1 explains the relation between received configuration size and relative processing time.

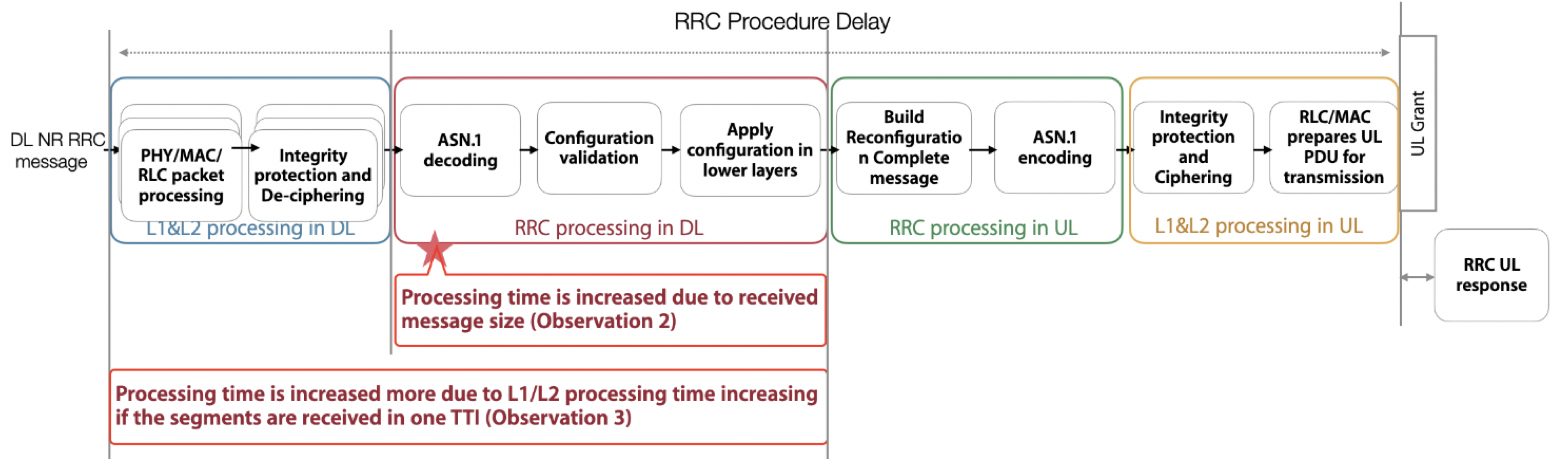


Figure-1. Impact on the UE processing on DL RRC message

## Detailed solutions

About the detailed solution to define the RRC processing time requirement for DL RRC message with segmentation, following options were proposed during the offline discussion [2]:

* Option 1: 16ms\*Nseg.
* Nseg is number of RRC segments
* Option 2: 16ms + (Nseg-1)\*X
* Nseg is number of RRC segments
* 16ms includes the processing time of UE functionalities which is needed only once for all received segments and no impact by the message size.
* X is the additional processing time per segment, e.g. DL processing, extra processing time for ASN.1 decoding, configuration application.
* X time in milli-seconds required to process an RRC segment.
* Option 3: fix value
* Define one fix value to cover all cases, including the case of the max segment number.
* The value should be the max value of the options in the function form.
* Option 4: (16+Y) + (Nseg-1)\*X
* Nseg is number of RRC segments
* Y = scale up delta to accommodate for the difference in size between max size for legacy RRC message (e.g. 9 KB) and max size of new RRC message (e.g. 45KB)
* X time in milli-seconds required to process an RRC segment

Take the RRC Reconfiguration message with 5 segments as the example, the processing time for each option is provided in the table below.

|  |  |  |
| --- | --- | --- |
| Options | Definition | Processing time |
| 1 | 16ms\*Nseg | 16\*5 = 80ms |
| 2 | 16ms + (Nseg\*X) | 16 + 4X |
| 3 | Fix value | Max of |
| 4 | (16+Y) + Nseg\*X | (16+Y) + 5X |

It is noted that legacy RRC processing time delay is applied for the RRC message without segmentation (i.e. not transmitted by DLDedicatedMessageSegment-r16).

#### **Question 1: Which solution do you prefer?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Preference** | **Comments** |
| Qualcomm | Option-2 but.. | We prefer option-2, but it seems that Nseq is defined as the number of segments. If no segmentation was initiated, then Nseq =1   * If RRC message wasn’t segmented, and if Nseq is assumed to be = 1 🡪 this will contradict with the current requirement, as processing delay would be = 16 + X * If RRC message wasn’t segmented, and if Nseq is assumed to be = 0 🡪 this is aligned with current delay requirement (= 16 ms)   + Based on this, we recommend modifying the equation to     - Nseq = number of segment -1, with Processing delay 16 + (Nseg\*X)   Or   * + - Nseq = number of segment with Processing delay = 16 + (Nseq-1)\*X |
| MediaTek | Option 3 (Accept option 4 and option 2) | Option 3 is simpler as we have 5 segments at most. Only 4 new cases (number of segments = 2, 3, 4, 5) need to be discussed.  For both option 2 and 4, Nseg should be “number of segment -1” as mentioned by QC.  Note that we also have DL segment in LTE, so both **LTE and NR** processing time should be updated. |
| Huawei, HiSilicon | Option 2 | We also agree with the observation from QC on Nseg |
| CATT | Option 2, but | We prefer to option 2, and we agree with Qualcomm, when no segmentation was initiated, the processing time should be 16ms, we prefer to define the Processing delay = 16 + (Nseq-1)\*X, Nseq = number of segment |
| Ericsson | Option 2 with comment | We also agree with the observation from Qualcomm. |
| Apple | Option 1, Option 2 | Option 1 is simple and we can avoid to discuss the X value. And Option 2 is acceptable to us.  For the no segmentation case, our intention is to keep the current processing time requirement, i.e. no change. And for the segmentation case, the segmentation number cannot be 1. |
| Xiaomi | Option 2 (Accept Option 1 and Option 4) | We think that Option 2 is probably more accurate to the UE’s internal processing delay budget. However it seems a little bit difficult to define the value for the segment processing time. If companies are not able to achieve the common value for the segment processing time, we could also have Option 1 to allow more relaxed UE processing time. |
| Samsung | Option 2 | We think option 2 makes sense when the value of X is not marginal, as otherwise option 3 would be fine.  Note: We assume N represents number of additional segments as commented by Qualcomm |
| Nokia | More like Option 2 | Option 1 is unacceptable as processing time cannot make sense linearly increasing with num of segments.  Option 2 would make sense but X cannot be 16 msec but quite some reasonably lower value additional overhead per segment?  Option 3 should be fixed upper limit?  Option 4 what is the order of Y, is it as worse as 16 msec? |

#### **Question 2: If Option 2 (i.e. 16ms + (Nseg\*X)) is your preference, what do you think is the value of X?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Preferred X value** | **Comments** |
| Qualcomm |  | We prefer to resolve the issued raised in Q-1 first before providing a value. |
| MediaTek | 2ms | Prefer option 4 than option 2 |
| Huawei | 2ms |  |
| CATT |  | Agree with Qualcomm |
| Ericsson |  | Before discussing about what is the value of X, we need to clarify that the formula for Option 2 is not correct at the moment and need to be revised based on QC comment. |
| Apple | 12ms ~ 16ms | X is the additional delay per segment, which include the extra processing time for ASN.1 decoding, configuration validity and applying the configuration internally. |
| Nokia |  | Agree with QC but then disagree with Apple that 16 msec is an additional overhead per segment? What is the reason for such a low performance from UE?  Option 2 seems reasonable with order of around 1-2 msec as other companies mentioned not definitely more than 2 msec. |

#### **Question 3: If Option 4 (i.e. (16+Y) + Nseg\*X) is your preference, what do you think is the value of X and Y?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Preferred X, Y value** | **Comments** |
| MediaTek | Y=2ms, X=1ms | Y for extra delay in the concatenation of the segments and additional ASN.1 decoding time for larger message.  X for additional L1/L2 processing delay on the extra L1/L2 configurations. |
|  |  |  |
|  |  |  |
|  |  |  |

#### **Question 4: If Option 3 (i.e. fix value) is your preference, do you agree the value should cover the latency of the max segment number?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Agree or not?** | **Comments** |
| MediaTek | Agree | For NR, we believe that 20ms ~ 25ms should be enough  For LTE, we believe that 25ms ~ 30ms should be enough |
| Huawei | Agree | For us the value of 25 ms should be able to cover the worst case scenario both for NR and for LTE. |
|  |  |  |
|  |  |  |

## RAN5 Impact

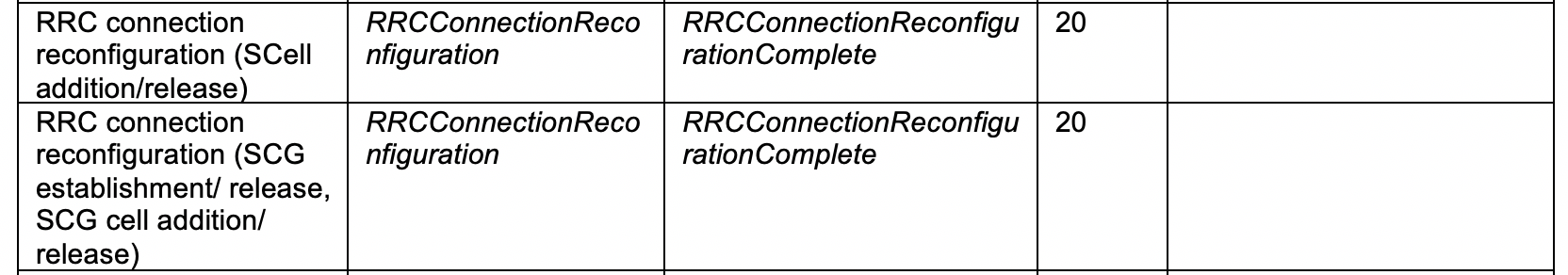
The test cases are currently defined in 38.523 for checking the RRC processing delay (see Annex). Since the RRC processing delay is extended for the RRC message with segmentation, the defined test case is not applicable for the RRC message with segmentation. Therefore, it’s better to inform RAN5 to exclude the RRC message with segmentation from current test cases.

#### **Question 5: Do you agree to inform RAN5 about the RRC processing time extension for the RRC message with segmentation?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Agree or not?** | **Comments** |
| Qualcomm | Agree |  |
| MediaTek | Agree | Ok to inform RAN5 our conclusions. However, it seems not necessary to exclude RRC segmentation case. RAN5 could just take the processing delay into account. |
| Huawei | Agree |  |
| CATT | Agree |  |
| Ericsson | Agree |  |
| Apple | Agree |  |
| Xiaomi | Agree |  |
| Samsung | Agree |  |
| Nokia | Agree |  |

## LTE change

As companies indicated above, RRC message with segmentation is also supported in LTE in R16, and the processing time extention is also need to be considered.



#### **Question 6: Do you agree to apply the same rule on the RRC processing time with RRC segmentation in LTE?**

* Option 1: 20ms\*Nseg.
* Option 2: 20ms + (Nseg-1)\*X
* Option 3: fix value
* Option 4: (30+Y) + (Nseg-1)\*X

|  |  |  |
| --- | --- | --- |
| **Company** | **Agree or not?** | **Comments** |
| Apple | Yes |  |
| Xiaomi | Yes |  |
| Samsung | Yes |  |
| Nokia | Yes | Agree similar view as NR. |

# Summary

# Conclusion

The followings are proposed:

**Proposal 1: ?**

# Reference

1. [R2-2009488](file:///D:\Documents\3GPP\tsg_ran\WG2\TSGR2_112-e\Docs\R2-2009488.zip) Discussion on RRC processing delay Apple discussion Rel-16 TEI16
2. [R2-2011176](file:///D:\Documents\3GPP\tsg_ran\WG2\TSGR2_112-e\Docs\R2-2011176.zip) [AT112-e][029][NR TEI16] Misc Corrections II (ZTE) ZTE Corporation

# Annex: RAN5 test case in TS38.523

**8.1.5.8** Processing delay

8.1.5.8.1 Processing delay / RRC\_Idle to RRC\_Connected / RRC\_Inactive to

RRC\_Connected / Success / Latency check

8.1.5.8.1.1 Test Purpose (TP)

(1)

with { UE in NR RRC\_IDLE state and has sent an RRCSetupRequest message }

ensure that {

when { UE receives RRCSetup message and after 10ms receives an UL grant }

then { UE successfully transmits RRCSetupComplete message }

}

(2)

with { UE in NR RRC\_CONNECTED state }

ensure that {

when { UE receive a SecurityModeCommand message and after 5ms receives an UL grant }

then { UE successfully transmits SecurityModeComplete }

}

(3)

with { UE in NR RRC\_CONNECTED state }

ensure that {

when { UE receive a RRCReconfiguration message to establish DRB that is not part of the current UE

configuration and after 10ms receives an UL grant }

then { UE successfully transmits RRCReconfigurationComplete message }

}

(4)

with { UE in NR RRC\_CONNECTED state }

ensure that {

when { UE receives an UECapabilityEnquiry message and after 80ms receives an UL grant }

then { UE successfully transmits an UECapabilityInformation message }

}

(5)

with { UE in NR RRC\_INACTIVE state and has sent an RRCResumeRequest message }

ensure that {

when { UE receives RRCResume message and after 10ms receives an UL grant }

then { UE successfully transmits RRCResumeComplete message }

}

(6)

with { UE in NR RRC\_CONNECTED state }

ensure that {

when { UE receives an RRCReconfiguration message containing sCellToAddModList with a SCell

addition and after 16ms receives an UL grant }

then { UE successfully transmits RRCReconfigurationComplete message }

}

**8.2.6.2** Processing delay

8.2.6.2.1 Processing delay / PSCell addition / SCG DRB / Success / Latency check / ENDC

8.2.6.2.1.1 Test Purpose (TP)

(1)

with { UE in RRC\_CONNECTED state with EN-DC, and, MCG(s) (E-UTRA PDCP) only }

ensure that {

when { UE receives an RRCConnectionReconfiguration message to add PSCell with SCG DRB and after 20

subframes receives an UL grant }

then { UE successfully configures the PSCell with SCG DRB and sends an

RRCConnectionReconfigurationComplete message }

}