**3GPP TSG-RAN WG2 Meeting #130 R2-250xxxx**

**St Julian, Malta, 19th – 23rd May 2025**

**Source: vivo**

**Title: Remaining RLC open issues for XR Enhancements**

**Agenda Item: 8.7.1**

**Document for: Discussion and Decision**

1. Introduction

This document summarizes the discussion of the following email discussion and collects the RLC open issues for XR Enhancements according to the following email discussion.

* [POST129bis][505][XR] RLC running CR and open issues (vivo)

Scope:

* Update and review the CR
* List open issues related to the CR

Intended outcome:

* Running CR for endorsement in the next meeting
* List of open issues for discussion at the next meeting

Deadline: Long

Companies are invited to provide comments/additional open issues in the below table by 2nd May, 2025.

# Contact information

|  |  |
| --- | --- |
| Company | Name, Email |
| Vivo | Chenli, chenli5g@vivo.com |
| Ofinno | Hsin-Hsi Tsai, htsai@ofinno.com |
| OPPO | Zhe Fu, fuzhe@OPPO.com |
| ZTE | Eswar Kalyan Vutukuri, eswar.vutukuri@zte.com.cn |
| Xiaomi | Yujian, zhangyujian@xiaomi.com |
| Nokia | Benoist, benoist.sebire@nokia.com |
| LGE | Gyeong-Cheol, gyeongcheol.lee@lge.com |
| Sharp | Fangying.xiao@cn.sharp-world.com |
| Lenovo | Shwetha Sreejith, ssreejith1@lenovo.com |
| HONOR | Jian Zhang, zhangjian23@honor.com |
| Samsung | Vinay Kumar Shrivastava, shrivastava@samsung.com |
| Futurewei | Yunsong Yang, yyang1@futurewei.com |
| Apple | Ping-Heng Wallace Kuo, pingheng\_kuo@apple.com |
| Fujitsu | Sue Yi, [yisu@fujitsu.com](mailto:yisu@fujitsu.com) |
| Ericsson | Nithin Srinivasan, nithin.srinivasan@ericsson.com |

# Discussion

* 1. Issues related to running CR

**Open issue RLC-1 (essential): Terminology for avoiding unnecessary retransmission, e.g. “obsolete”, or “outdated”, or “discard”**

In the current RLC running CR, there is an EN as below:

Editor’s Note: FFS on the term, whether it should be “obsolete”, or “outdated”, or “discard”. Same as below.

Current RLC running CR use the term from RRC running CR “*stopReTxObsoleteSDU”*, “*t-RxDiscard*” . Meanwhile, some corresponding description is used, e.g. “This timer is used by the receiving side of an AM RLC entity in order to abandon an obsolete SDU”.

During the discussion, companies have different preference on the term regarding the enhancement to avoid unnecessary RLC retransmission, e.g. “obsolete”, or “outdated”, or “discard”.

**Companies are invited to provide your preference on the term to be used for the enhancement to avoid unnecessary RLC retransmission for both UL and DL operations, e.g. “obsolete”, or “outdated”, or “discard”, or others, please specify:**

|  |  |  |
| --- | --- | --- |
| **Company** | **Preference(s)** | **Comments, if any** |
| Ofinno | x) *stopReTxSDU*  This parameter is used by the transmitting side of each AM RLC entity to determine whether to stop RLC transmission and retransmission of SDUs (see clause 5.2.3) .  x) *t-RxDiscard*  This timer is used by the receiving side of an AM RLC entity in order to discard AMD PDU(s) (see clause 5.2.3.2.x). … | For *stopReTxObsoleteSDU*, we prefer not to use the words “obsolete” and “outdated” unless the standard clearly defines what is “obsolete” or “outdated”. Regarding “discard”, since the discarding for “*stopReTxObsoleteSDU*” occurs at PDCP rather than RLC, “dicard” might not be the appropriate term to use. In our understanding, there’s no need to assign an adjective to this word considering that the exact UE behavior is clearly specified, thus *stopReTxSDU* seems already be clear.  For *t-RxDiscard*, there is no issue with the naming of this parameter as it is indeed related to the discarding operation within RLC. However, we also feel that “obsolete” is unnecessary for similar reasons stated earlier. Additionally, “abandon” could be replaced with “discard” since the terminology for the timer and actions in 5.2.3.3.x is “discard”.  Furthermore, the expiry of the *t-RxDiscard* may result in discarding multiple AMD PDUs instead of just a single PDU. We suggest updating it to AMD PDU**(s)**. |
| OPPO | Prefer to use discard for both Tx and Rx side |  |
| ZTE | Prefer use “discard” | Because the case is that PDCP has discarded the SDU for *discardTimer* expiring and indicate the RLC entity to discard the SDU. E.g. in TS 38.323,   |  | | --- | | If the corresponding PDCP Data PDU has already been submitted to lower layers, the discard is indicated to lower layers. |   The same terminology should be used for the same case. |
| Xiaomi | Same view as OPPO. For Tx side, we can use “*stopReTxDiscardedSDU*”, and for Rx side, we can keep “*t-RxDiscard*” | Tx side operation is based on discard indication from PDCP layer, as from running CR copied below. Using “discard” is straightforward and we don’t need to define terminologies like “obsolete” / “ outdated”.  If stopReTxObsoleteSDU is set to enabled, when receiving a discard indication for an RLC SDU with SN = x from the upper layer |
| LGE | Ok with discard for both Tx and Rx side |  |
| Sharp | Prefer to use discard for both Tx and Rx side | The intention for both sides is to abandon (discard) outdated SDUs. Discard is more intuitive than obsolete. |
| Lenovo | 1. *stopReTxSDU* 2. *t-RxDiscard* | No strong view but okay with Ofinno’s suggestion. |
| HONOR | “discard” | Both TX and RX. |
| Samsung | For Tx side, prefe to use “discard”  For Rx side, prefer to use “outdated” | For Rx side, it is about determining SDU as outdated and abandoning it, and there may be no discard needed when no byte-segment is actually received for such SDU. Rx side still determines such SDU as outdated (i.e. detection of obsolescence) and triggers SR. Therefore, *t-RxOutdated* could be more appropriate name |
| Futurewei | No strong view on “discard” vs. “obsolete” | However, if “discard” is to be used in the TX side terminology, it should be “DiscardedSDU”, not “DiscardSDU”, because the main action is “stop”, not discard.  Another question on the TX side terminology is, given the following text in the RLC running CR, whether we should change “stopReTx” to “stopTx” (because remaining segment that is pending the initial transmission is also stopped)?  5.2.3.1 Transmit operations  5.2.3.1.1 General  …  “If stopReTxObsoleteSDU is set to enabled, … the transmitting side of an AM RLC entity shall not consider the corresponding RLC SDU or RLC SDU segment for transmission or retransmission.” |
| Apple | “Discard” for both TX and RX |  |
| Fujitsu | Prefer obsolete/outdated in both TX and RX | For combined TX and RX apporach, the motivation is the same for TX and RX, which is to abandon obsolete/outdated SDU.  In RX side, “discard” is not appropriete since there may be no AMD PDU within the SN gap in the reception buffer and there is nothing to discard. The RX side behavior is to abandon the outdated SDU that is missing in this case. |

**Open issue RLC-2 (not essential, but important): whether further changes are needed for SR triggered by t-RxDiscard expires.**

In the current RLC running CR, there is an EN as below:

Editor’s Note: FFS whether any further changes are needed for SR triggered by *t-RxDiscard* expires. Companies are invited to provide views (if any) in the summary.

RAN2 agreement is:

*When the t-RxDiscard expires, the expiration of t-RxDiscard triggers an SR. FFS whether this is just usual SR or some changes are needed, or if UE implementation can decide (to be discussed during CR review)*

Current RLC running CR captured the SR is triggered when *t-RxDiscard* expires as below:

|  |
| --- |
| Triggers to initiate STATUS reporting include:  ….  - Detection of obsolescence of an AMD PDU:  - The receiving side of an AM RLC entity shall trigger a STATUS report when *t-RxDiscard* expires. |

**Companies are invited to provide comments on any further changes are needed for SR triggered by *t-RxDiscard* expires.**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comments, if any** |
| Ofinno | Yes (further changes are needed) | In 5.2.3.2.x, when *t-RxDiscard* expires, it is possible that no AMD PDU is dicarded, for example, if there is no AMD PDU in the reception beffer with SN < RX\_Next\_Discard\_Trigger, so that’s why we added “if any” at the end. In this case, we think the SR is not needed to be triggered, i.e., the SR shall be triggered when at least one AMD PDU is discarded when *t-RxDiscard* expires.  5.3.4 Status reporting… “Detection of obsolescence of an AMD PDU” should also need to be fixed based on the outcome of the 1st issue.  We suggest the following changes:  Detection of discarded AMD PDU(s) when *t-RxDiscard* expires: …   * The receiving side of an AM RLC entity shall trigger a STATUS report when the AMD PDU(s) is discarded when *t-RxDiscard* expires.   [Fujitsu]: Disagree with the change. It is true that there may be scenario where no AMD PDU is discarded. However it may be that no segment has been received for an expected SDU. In this case, an SR should be triggered to let the TX side update the status and advance the TX window. We are fine with Rapp’s version. |
| OPPO | No |  |
| ZTE | Yes | Additional clarify that the SN of discarded AMD PDU should be included in the ACK\_SN of the SR. |
| Xiaomi | No | We don’t think further changes are needed. Upon expirty of *t-RxDiscard*, the RLC state variable RX\_Next is updated, and legacy status report can be used so that Tx side can move the window.  We don’t think the changes suggested by Ofinno is needed. In RLC layer, whenever a full RLC SDU can be reassembled, it is delivered to PDCP layer directly. So the discarded AMD PDUs are those SDU segments. Even if there is no such discarded AMD PDUs, Rx window is still moved upon expirty of *t-RxDiscard* since RX\_Next is updated. Status report is still needed to inform the Tx side to move its window accordingly. |
| Nokia | No further changes |  |
| LGE | No |  |
| Sharp | Yes | We see there is an issue of duplicate retransmissions due to close SR triggerings. Since two SRs are triggered by t-reasembly and t-RxDiscard independently, both triggered SRs could be close to each other. The second status report tiggered by t-RxDiscard could arrive during ongoing HARQ retransmission based on status report triggered by t-reassembly. This can be considered as duplicate retranmissions. That’s the main reason why we said there is a side-effect in the last meeting. |
| Lenovo | No | We have similar understanding as Xiaomi and think that the normal SR can be reused and we do not see the need for any additional changes to the SR triggering procedure beyond what is already agreed.  But “Detection of obsolescence of an AMD PDU:” will need to be aligned with the outcome of the previous open issue. |
| HONOR | Yes, RX\_Highest\_Status needs to be updated. | In TS 38.322:  c) RX\_Highest\_Status – Maximum STATUS transmit state variable  This state variable holds the highest possible value of the SN which can be indicated by "ACK\_SN" when a STATUS PDU needs to be constructed. It is initially set to 0.  When *t-RxDiscard* expires, it is possible that the value of ACK\_SN will be greater than RX\_Highest\_Status. If RX\_Highest\_Status is not updated, it dose not hold the highest possible value of the SN which can be indicated by "ACK\_SN" when a STATUS PDU needs to be constructed. There will be a problem of RX\_Highest\_Status and ACK\_SN not being aligned. |
| Samsung | No | We think changes proposed by Ofinno may lead to wrong interpretation. As we also pointed in previous comment this does not require an actual discard. When an outdated SDU is determined upon expiry of timer, a SR needs to be triggered. |
| Futurewei | No (and please see our response to ZTE in 2.3) |  |
| Apple | Yes | The usual status report can be reused.  However, considering that there is already a delay due to the timer at the RX side, we think transmission of this status report should not be constrained by a running *t-StatusProhibit*, it should be transmitted immediately once triggered by expiration of *t-RxDiscard*. |
| Fujitsu | Yes | Agree with Honor that RX\_Highest\_Status needs to be updated. |
| Ericsson | No |  |

**Open issue RLC-3 (essential): whether use the terminology of “autonomous retransmission” or others.**

In the current RLC running CR, there is an EN for autonomous retransmission procedure in clause 5.x as below:

Editor’s Note: FFS on the terminology of “autonomous retransmission”, as it was already used for NR-U.

During the discussion, some companies mentioned that we have existing “autonomous retransmission” procedure in Rel-16 NR-U. TS 38.300 uses “autonomous retransmission” terminology for the NR-U feature. RAN2 may need to consider to use a new terminology for this Rel-19 feature, and “autnomous retranmsision” for Rel-19 XR is a working teminology, a standardized terminology should be discussed later.

With this, rapporteur suggests to change it to another name, e.g. timer-based retransmission, etc.

**Companies are invited to provide comments on whether to change the “autonomous retransmission” to another term in the specification, if “yes”, please provide your suggestion.**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No to change** | **Suggestion on the term, e.g. timer-based retransmission** |
| Ofinno | Yes | timer-based may not be ideal as this retransmission relies on the PDCP timer rather than the RLC timer. We suggest using “**delay-based retransmission**”, as the retransmission is based on the remaining time of an RLC SDU, which is the delay information of the RLC SDU.  If the term of the procedure is changed, the term of the parameter *autonomousReTxThreshold* should also be changed accordingly. |
| ZTE | Yes | Considering that no timer is introduced for “autonomous retransmission”, we prefer to use “remaining time based retransmission” |
| Xiaomi | No strong view | If naming change is needed, we can use “RLC autonomous retranmission” to differentiate from CG autonomous retransmissions in MAC. |
| Nokia | Depends on RLC-4 | If we merge, perhaps there’s no need to distinguish the two. |
| LGE | Yes | “remaining time based retransmission” seems good to us. |
| Sharp | Yes | Remaining time based retransmission is also ok, if this term is used somewhere in the specification. |
| Lenovo | Yes | We also think that the standardized name of feature needs to be changed from its working terminology to avoid confusion between separate procedures. We suggest ‘**proactive retransmission**’ but also okay with Ofinno’s suggestion. |
| HONOR | No strong view | Keep using “autonomous retranmission” in RLC spec and use “RLC autonomous retranmission” in TS 38.300. |
| Samsung | Yes | Remaining time based retransmission seems okay |
| Futurewei | We are fine to keep it | But also fine with “remaining time based retransmission” (and can add “RLC” for 38.300). Prefer not to use “delay-based retransmission” because it sounds like the retransmission is delayed. |
| Apple | No strong view | Maybe we can call it “Autonomous RLC Retransmission” |
| Ericsson | Yes | It is not truly autonomous, anything related to remaining time is more accurate. |

**Open issue RLC-4 (essential): whether merge the autonomous retransmission procedure in clause 5.x into 5.3.2 or capture it separately.**

In the current RLC running CR, there is an EN for autonomous retransmission procedure in clause 5.x as below:

Editor’s Note: FFS on whether to merge this section into 5.3.2 or capture it separately.

During the discussion, some companies prefer to merge the whole autonomous retransmission procedure in 5.x into section 5.3.2 “retransmission”, noticing large portion of behaviour/content are same, and we can treat the autonomous retransmission trigger as same as receiving a NACK feedback. While some companies prefer to keep the autonomous retransmission procedure in 5.x as a separate section as autonomous retransmission does not rely on feedback or request, and it is a little odd to add autonomous retransmission to the ARQ section.

**Companies are invited to provide comments on whether to merge autonomous retransmission procedure in 5.x into section 5.3.2 “retransmission” or keep it as a separate section.**

|  |  |  |
| --- | --- | --- |
| **Company** | **Merge or separate section** | **Comments, if any** |
| Ofinno | separate sections | UE behaviors differ for these two types of retransmission. Creating separate sections would improve understanding. |
| OPPO | Merge | To avoid duplicated text and also make it clear that the retransmission of a SDU will be triggered either by NACK or autonomously based on the timer. |
| ZTE | Prefer to merge | Because both are retransmission. |
| Xiaomi | Merge | We prefer to merge since the main difference is the triggering of retransmission. |
| Nokia | Merge | The ARQ section already contains retransmissions due to expiry of t-PollRetransmit, which are also not based on ACK/NACK feedback. |
| LGE | Merge | It would be good to specify all retransmission operations at one place. |
| Sharp | Prefer to merge | It seems that both legacy retransmission and autonomous retransmission have a big commonality. Merging them can avoid duplicate descriptions. |
| Lenovo | Separate | We have no strong view, but prefer to have separate section for ease of understanding and since the trigger condition for such retransmission is different from ARQ. |
| HONOR | Merge section | Because autonomous retransmission and ARQ retransmission are both retransmission and only differ in the triggering conditions. |
| Samsung | Merge | Avoid duplicate descriptions as far as possible |
| Futurewei | Perfer to merge | Not only to avoid duplication but also, through the merging, to identify and address potential co-existence issue, if any. |
| Apple | Merge | This is simpler to model it as one addition triggering condition of retransmission. |
| Ericsson | Merge | Should integrate into the current text as much as possible. At least, this does not warrant a new section. |

* 1. Open issue list

**Rapporteur provides the list of open issues as below, and the corresponding suggestions on how to address them. Some of them could be further discussed based on contributions or resoved based on further progress. Companies are invited to provide comments on whether it is open issue and whether the suggestions from reapporteur is accuracy enough.**

|  |  |
| --- | --- |
| **Company** | **Comments, if any** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### DSR enhancements

|  |  |  |
| --- | --- | --- |
| **Index** | **Issue description** | **Rapporteur suggestion** |
| RLC-5 | Editor’s Note: Same as PDCP open issue: it is FFS which delay-reporting RLC data volume shall consider RLC data PDUs to be retransmitted.  Editor’s Note: It is FFS which delay-reporting RLC data volume shall consider STATUS PDU to be transmitted. | **Issue Type:** Essential  **How to address it:** based on companies’ contribution |

### Avoid unnecessary RLC retransmissions

|  |  |  |
| --- | --- | --- |
| **Index** | **Issue description** | **Rapporteur suggestion** |
| RLC-6 | Editor’s Note: FFS whether there are any RLF detection impacts when avoiding unnecessary retransmissions is introduced. | **Issue Type:** Not essential but important  **How to address it:** based on companies’ contribution |

### Ensure timely RLC retransmissions

|  |  |  |
| --- | --- | --- |
| **Index** | **Issue description** | **Rapporteur suggestion** |
| RLC-7 | Editor’s Note: It is still open how Autonomous Retransmission coexists with ARQ procedures, i.e. whether/how to increment the RETX\_COUNT for Autonomous Retransmission. | **Issue Type:** Not essential but important  **How to address it:** based on companies’ contribution |
| RLC-8 | Editor’s Note: It is still open on how to avoid excessive polling for the polling enhancement, e.g. only one polling or multiple. | **Issue Type:** Not essential but important  **How to address it:** based on companies’ contribution |
| RLC-9 | Editor’s Note: FFS whether/what additional conditions are needed to prevent too early and/or unnecessary retransmission due to polling enhancement. | **Issue Type:** Not essential but important  **How to address it:** based on companies’ contribution |

* 1. Others, please specify

Companies are invited to describe any other identified open issues not currently included within this document.

|  |  |
| --- | --- |
| **Company** | **Other identified open issues? (please describe) or other comments** |
| ZTE | For SDU discard indication from upper layer (e.g. PDCP), there are two branch to describe:  The first is:   |  | | --- | | 5.4 SDU discard procedures  When indicated from upper layer (e.g. PDCP) to discard a particular RLC SDU, the transmitting side of an AM RLC entity or the transmitting UM RLC entity shall discard the indicated RLC SDU, if neither the RLC SDU nor a segment thereof has been submitted to the lower layers. The transmitting side of an AM RLC entity shall not introduce an RLC SN gap when discarding an RLC SDU. |   The second is:   |  | | --- | | 5.2.3.1.1 General  ...  If *stopReTxObsoleteSDU* is set to enabled, when receiving a discard indication for an RLC SDU with SN = x from the upper layer (see TS 38.323 [4]), the transmitting side of an AM RLC entity shall not consider the corresponding RLC SDU or RLC SDU segment for transmission or retransmission.  Editor’s Note: FFS on the term, whether it should be “obsolete”, or “outdated”, or “discard”. Same as below. |   Based on these description, we can concluded that: if the RLC SDU or a segment thereof has been submitted to the lower layers, the transmitting side of an AM RLC entity shall not discard the indicated RLC SDU; the transmitting side of an AM RLC entity only stops the transmission or retransmission of the indicated RLC SDU.  Then for DL, if t-RxDiscard is not configured, the RLC SDU or RLC SDU segment(s) will stay in the transmitting window and lead the transmitting window stalling. If the transmitting window stalling is avoided based on t-RxDiscard mechanism, it should be specified that if DL *stop ReTx Obsolete SDU* is supported*,* the DL *t-RxDiscard* should be configured; and if UL *stopReTxObsoleteSDU* is configured, UL timer based Rx discard should be activated in NW.  Otherwise, transmitting window forward should be captured, such as:   |  | | --- | | 5.2.3.1.1 General  ...  If *stopReTxObsoleteSDU* is set to enabled, when receiving a discard indication for an RLC SDU with SN = x from the upper layer (see TS 38.323 [4]), the transmitting side of an AM RLC entity shall:  - discard the corresponding RLC SDU or RLC SDU segment;  - consider the positive acknowledgment has been received for the SDU with SN=x.  - set TX\_Next\_Ack equal to the SN of the RLC SDU with the smallest SN, whose SN falls within the range TX\_Next\_Ack <= SN <= TX\_Next and for which a positive acknowledgment has not been received yet.  Editor’s Note: FFS on the term, whether it should be “obsolete”, or “outdated”, or “discard”. Same as below. |   [FW]: it is welcoming to see the suggestion that this can be done on the UE side. In the previous RAN2 meeting (and R2-2502218), we basically suggested that this could be done at the gNB side (by implementation) without a need for the UE to trigger the SR immediately after its *t-RxDiscard* expires. If the concern is that the TX side should advance after the RX side, any smart implementation (of UE and gNB as the TX’er) can add a delay timer before the TX window is pushed forward. Because the time it takes the SN gap to stall the TX window is so much longer than PDB/PSDB (e.g., 156 msec vs. 10 msec), it is very easy for implementation to pick a delay timer value that is long enough to ensure the RX side has done the same already and still short enough with plenty safety margin before the SN gap can stall the TX window. The SR that we are currently mandating the UE to immediately transmit is completely wasteful and potentially harmful for delaying subsequent legitimate SR (due to the prohibit timer running). We should reverse that agreement made in meeting #129bis to avoid introducing such a blunder in the Rel-19 RLC spec. |
| Nokia | By the current CR, when t-PollRetransmit expires, if there are only SDUs buffered whose transmissions have been stopped due to discard indication from PDCP, there is no SDU to retransmit the poll with. |
| Apple | We have a similar concern as Nokia. An ongoing polling procedure seems to be no longer necessary when all RLC SDUs with SN between (and including) TX\_Next\_Ack and POLL\_SN are already positively/negatively acknowledged or discarded. |
|  |  |
|  |  |

# Conclusion

In this contribution, we collect the open issues for XR enhancements in RLC as below:

***DSR enhancements***

***Avoid unnecessary RLC retransmissions***

***Ensure timely RLC retransmissions***

# References

1. R2-25xxx, RLC running CR for XR, vivo.