**3GPP TSG-RAN WG2 Meeting #123bis** **R2-231xxxx**

**Xiamen, China, 11-17 October, 2023**

**Agenda item: 7.5.1**

**Source: Qualcomm Incorporated**

**Title: Open issues in MAC running CR for XR enhancements**

**Document for: Discussion and Decision**

1. Introduction

This document aims to facilitate the discussion on open issues related to MAC CR for XR enhancements, as per the following e-mail discussion:

**[POST123bis][024][XR] 38.321 Running CR (Qualcomm)**

Scope:

- Review running CR

- Identify open issues

- Get inputs for subset of open issues (focus more detailed open issues that would help with CR finalisation.

Deadline: Nov 1st, 2023

In this discussion, companies may provide their input for (minor) open issues related to stage-3 design details in the MAC running (e.g. MAC CE format design, etc) that have not been officially agreed yet. The intention is to build consensus and prepare for easy agreement at the next RAN2 meeting.

The following are the deadlines for this discussion:

- Company feedback: by 2200 on Oct 30 UTC

- Rapporteur’s summary: by 0900 on Oct 31 UTC

- Final deadline: by 2200 on Nov 1 UTC

2. Contact information

Please provide your contact information in the table below.

|  |  |  |
| --- | --- | --- |
| **Company** | **Delegate** | **Email** |
| LGE | Hanseul Hong | hanseul.hong@lge.com |
| Apple | Ping-Heng Wallace Kuo | pingheng\_kuo@apple.com |
| Huawei, HiSilicon | Yinghao Guo | yinghaoguo@huawei.com |
| Samsung | Hyunjeong Kang | hyunjeong.kang@samsung.com |
| Nokia | Chunli Wu | Chunli.wu@nokia-sbell.com |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

3. Input for the open issues

3.1 Enhanced BSR MAC CE

First, let us discuss the format of the Enhanced BSR MAC CE.

One of the key questions in the format design is how to indicate which BSR table an LCG uses. The rapporteur thinks there can be at least two options (as illustrated in Figure 1):

* Option 1. Introduce a new 8-bit bitmap which indicates which BSR table an LCG uses; or
* Option 2. Add one bit indicator coupled with each Buffer Size field.

There may be other design options too. If you think so, please describe your preferred design in the Comments column.



**Figure 1. Options for the format of the Enhanced BSR MAC CE**

**Question 1: Which format for the Enhanced BSR MAC CE do you prefer?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Company** | **Option 1/2 or other** | **Comments** | |
| LGE | Option 1 | It looks simpler. | |
| Apple | Option 1 | Also, we prefer to model it as one additional BSR MAC CE format (can be dubbed as e.g. *Enhanced Long BSR*) in 6.1.3.1: 6.1.3.1 Buffer Status Report MAC CEs Buffer Status Report (BSR) MAC CEs consist of either:  - Short BSR format (fixed size); or  - Extended Short BSR format (fixed size); or  - Long BSR format (variable size); or  - Extended Long BSR format (variable size); or  - Short Truncated BSR format (fixed size); or  - Extended Short Truncated BSR format (fixed size); or  - Long Truncated BSR format (variable size); or  - Extended Long Truncated BSR format (variable size); or  - Enhanced Long BSR format (variable size); or  - Enhanced Long Truncated BSR format (variable size). | |
| Huawei, HiSilicon | Option1 |  | |
| Samsung | Option 1 | |  |
| Nokia | Option 1 | |  |
|  |  | |  |

**Summary:**

(to be added after the discussion)

In legacy, padding BSR includes a truncated version (i.e. long/short Truncated BSR MAC CE), which is used when there are not enough padding bits to accommodate a regular-sized BSR MAC CE. We may or may not need a truncated version of the Enhanced BSR MAC CE. Please share your preference and comments.

**Question 2: Is it necessary to introduce a truncated version of the Enhanced BSR MAC CE?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Company** | **Yes/No** | **Comments** | |
| LGE | Yes |  | |
| Apple | Yes | We think it is possible for the UE to send padding BSR with LCG using the new table. The UE can choose which format (legacy or new) to report according to the number of available padding bits. | |
| Huawei, HiSilicon | Yes | Same as legacy. Since padding BSR will be triggered as in legacy, the size of the BSR should be able to fit within the remaining space in the MAC PDU. It might need to be truncated in this case. | |
| Samsung | Yes | |  |
| Nokia | Could live without | | Legacy truncated BSR is anyway needed for the cases when the space is not enough to include the subheader and bitmaps of the Enhanced BSR MAC CE.  e.g. when there is only 2 bytes padding (with 1 byte subheader + 1 byte payload), it should be possible to report the LCG using the legacy 5-bit table, even if it is configured with new table and falls within the range. Otherwise, nothing can be reported.  Same for 3 or 4 bytes padding, better to use legacy table as well with 1 byte subheader + 2 or 3 bytes payload, since the two bytes bitmap in the enhanced BSR MAC CE does not provide any BS information.  If eLCID is used for Enhanced BSR, at least 5 bytes are needed for the 2 byte subheader + 2 byte bitmap + at least one BS.  Enhanced BSR could provide finer granularity, but legacy BSR can provide BS for two more LCG, so in that sense, can also live with legacy table only for truncated padding BSR. |
|  |  | |  |

**Summary:**

(to be added after the discussion)

There are three types of LCID (legacy 6-bit LCID, one octet eLCID, or two-octet eLCID) that the Enhanced BSR MAC CE may use.

**Question 3: which type of LCID do you think the new Enhanced BSR MAC CE should have?**

* **Option 1: legacy 6-bit LCID;**
* **Option 2: one-octet eLCID;**
* **Option 3: two-octet eLCID.**

|  |  |  |
| --- | --- | --- |
| **Company** | **Option 1/2/3** | **Comments** |
| LGE | Option 2 | Similar to other BSR formats |
| Apple | Option 1/2 |  |
| Huawei, HiSilicon | Option2 | The discussion should only be on Option2/1. Between these two options, we prefer option2. With XR services ongoing, there shouldn’t be coverage issues. |
| Samsung | Option 2 |  |
| Nokia | Option 2 | Overhead should not be too big concern when Enhanced BSR MAC CE is used. |
|  |  |  |

**Summary:**

(to be added after the discussion)

The Enhanced BSR MAC CE needs to be assigned a logical channel priority. The rapporteur does not see any strong reasons for it to have a priority different from that of the legacy BSR MAC CEs. Please indicate if you would agree.

**Question 4: Do you agree that the Enhanced BSR MAC CE should have the same logical channel priority as the legacy BSR MAC CEs (the ones except the padding BSR)?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comments** |
| LGE | Yes |  |
| Apple | Yes | In our view this is a new BSR format instead of an independent MAC CE, so the priority of which certainly does not change from the legacy BSR (just like we do not differentiate the priority between Long BSR and Short BSR). |
| Huawei, HiSilicon | Yes |  |
| Samsung | Yes | Enhanced BSR should be one of the BSR formats, similar to the cases of long and short BSR. |
| Nokia | Yes |  |
|  |  |  |

**Summary:**

(to be added after the discussion)

## 3.2 DSR MAC CE

Based on the agreements so far, the DSR MAC CE should include at least the following fields:

* LCG bitmap, which indicates which LCG has delay information included in the MAC CE;
* remaining time for a reported LCG;
* Amount of data associated with the reported remaining time.

Let us first discuss how to encode the remaining time. Based on proposals submitted so far, there are at least the following two options:

* Option 1: Encode the remaining time field directly through some linear mapping, since the typical delay requirements for UL XR traffic are not stringent (e.g. 50msec). As an example, we may define that the value *r* of the field corresponds to remaining time in the range of 0.5 × (*r*, *r*+1] msec, for r ∈(0, 63]. This mapping covers remaining times from 0 to 64 msec.
* Option 2: One may define a lookup table in the spec, similar to how BSR table is used to encode buffer sizes. This option seems useful only if companies decide to use a non-linear distribution to encode remaining times.

**Question 5: Which option do you prefer to encode the remaining time field in the DSR MAC CE?**

**- Option 1: define in the spec a linear mapping between the values of the field and the values of the remaining time, e.g. the value *r* of the field corresponds to remaining time in the range of 0.5 × (*r*, *r*+1] msec, for r ∈(0, 63];**

**- Option 2: define a lookup table in the spec, similar to how BSR table is used to encode buffer sizes. Distribution and range of this table can be discussed based on companies’ input (please provide details in your comment).**

|  |  |  |  |
| --- | --- | --- | --- |
| **Company** | **Option 1/2** | **Comments** | |
| LGE | Option 2 | Agree that linear mapping seems enough and there is no need to further optimize the remaining time table, but it looks simpler to define a lookup table for remaining time, rather than making a new equation.  In our view, one 4-bit linear table for remaining time is enough for XR traffic, e.g., 1ms to 15ms. | |
| Apple | Option 2 | A look-up table is more straightforward | |
| Huawei, HiSilicon | Option1 | If the mapping is linear, no need for a table but a formula should be enough | |
| Samsung | Option 2 | | Lookup table-based solution looks simpler and more efficient considering real-world implementation for both UE and gNB. |
| Nokia | Option 2 | | In the granularity of ms should be enough considering the discard timer is in ms. |
|  |  | |  |

**Summary:**

(to be added after the discussion)

It also needs to be discussed how to signal which BSR table is used to encode the buffer size field. Similar to the discussion on the Enhanced BSR MAC CE, the rapporteur thinks there can be at least two options: either use a bitmap such as the one used in the Enhanced BSR MAC CE or use one bit between the Remaining Time field and the Buffer Size field for the purpose. The formats of these two options are illustrated in Figure 2. Lastly, there is also the option to use only one particular table, e.g. use either only the legacy table or only the new table (if the range of the new BSR table that companies finally agree on is wide enough).



**Figure 2. Options for the format of the DSR MAC CE.**

**Question 6: which option do you prefer to indicate which BSR table is used to encode the Buffer Size field in the DSR MAC CE?**

* **Option 1: use a one-octet bitmap for the indication;**
* **Option 2: use a one-bit indicator for each reported LCG;**
* **Option 3: use only a specific BSR table (either only the legacy table or only the new table). Hence no indicator for is needed.**
* **Option 4: other (Please describe details of your preferred design in your comment).**

|  |  |  |  |
| --- | --- | --- | --- |
| **Company** | **Option 1/2/3/4** | **Comments** | |
| LGE | Option 3 | Given that DSR MAC CE only includes the data volume less than the configured delay threshold, the amount of data would not be large, so enhanced BS table is not needed and legacy BS table is enough.  However, if it is really needed to reduce the quantization error for DSR MAC CE, Option 1 is preferred. | |
| Apple | Option 3 | Agree with LGE. Also, DSR and the new BS table should be treated as two independent capabilities. We may have the cases where DSR is configured for a UE that does not support new BS Table. Thus, we prefer to keep it simple by not considering the new BS table for DSR in Rel-18. | |
| Huawei, HiSilicon | Option2 |  | |
| Samsung | Option 2 | | Option 2 is more concise than Option 1, i.e., no reserved bit and hence less overhead. |
| Nokia | Option 3 | | Agree with LG/Apple. Option 3 could be enough with the assumption that the data below delay threshold should be rather small and legacy table already provide good enough granularity for lower end, since otherwise the NW would not be able to schedule them on time and the two features of DSR and new BS table can be totally independent. |
|  |  | |  |

**Summary:**

(to be added after the discussion)

Next, let us discuss which type of LCID (legacy 6-bit LCID, one octet eLCID, or two-octet eLCID) the DSR MAC CE should have.

**Question 7: which type of LCID do you think the DSR MAC CE should have?**

* **Option 1: legacy 6-bit LCID;**
* **Option 2: one-octet eLCID;**
* **Option 3: two-octet eLCID.**

|  |  |  |
| --- | --- | --- |
| **Company** | **Option 1/2/3** | **Comments** |
| LGE | Option 2 | Similar to BSR MAC CE. |
| Apple | Option 2 |  |
| Huawei, HiSilicon | Option2 | Again, There is no coverage issue |
| Samsung | Option 2 |  |
| Nokia | Option 2 |  |
|  |  |  |

**Summary:**

(to be added after the discussion)

Please indicate which logical channel priority you think the DSR MAC CE should have. For example, if you think its priority should be below LBT Failure MAC CE but above MAC CE for SL-BSR, then please indicate “LBT failure” in the “Below” column and “SL-BSR” in the “Above” column.

**Question 8: which logical channel priority do you think the DSR MAC CE should have?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Company** | **Below** | **Above** | | **Comments** |
| LGE | Timing Advance Report | SL-BSR | |  |
| Apple | Timing Advance Report | SL-BSR | | We are also fine if the DSR has the same priority as SL-BSR or BSR. |
| Huawei, HiSilicon | LBT failure MAC CE |  | | Maybe it is beneficial to discuss whether the XR enhanced features can work together with NRU. Our thinking is that except for multi-PUSCH CG enhancement, other higher layer enchancement, like XR awareness, or PDU set discard are transparent to the lower layer transport. Hence, it should be possible to support them both  If both of them can be supported together, we think LBT failure MAC CE would be more important |
| Samsung | Timing Advance Report | | SL-BSR |  |
| Nokia | Timing Advance Report | | SL-BSR |  |
|  |  | |  |  |

**Summary:**

(to be added after the discussion)

## 3.3 PSI-based PDU discard activation/deactivation MAC CE

First, let us discuss which type of LCID (legacy 6-bit LCID, one octet eLCID, or two-octet eLCID) the PSI-Based PDU Discard Activation/Deactivation MAC CE should have.

**Question 9: which type of LCID do you think the PSI-Based PDU Discard Activation/Deactivation** **MAC CE should have?**

* **Option 1: legacy 6-bit LCID;**
* **Option 2: one-octet eLCID;**
* **Option 3: two-octet eLCID.**

|  |  |  |
| --- | --- | --- |
| **Company** | **Option 1/2/3** | **Comments** |
| LGE | Option 2 |  |
| Apple | Option 2 |  |
| Huawei, HiSilicon | Option2 | No coverage issue |
| Samsung | Option 2 |  |
| Nokia | Option 2 |  |
|  |  |  |

**Summary:**

(to be added after the discussion)

When specifying the handling procedures of DL MAC CEs, the MAC spec usually specifies the initial state of a feature upon its configuration and handover. The rapporteur thinks the same needs to be specified for the PSI-Based PDU Discard Activation/Deactivation MAC CE. Moreover, it is reasonable to assume that the PSI-based discard should be initially deactivated upon its configuration and handover.

**Question 10. Do you agree that the PSI-based PDU discard should be initially deactivated upon its configuration and handover?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comments** |
| LGE | No | Note that it should be PSI-based ‘SDU’ discard.  Similar to PDCP duplication, initial state for PSI-based SDU discard should be indicated by RRC. |
| Apple | No | We think it is useful to have some RRC-configured initial state per DRB, to handle the cases where congestion is already present when configured. The new RRC parameter *psi-BasedDiscard* (based on PDCP running CR) itself, if present, can be served as a binary flag for the initial state, so no additional overhead is needed. |
| Huawei, HiSilicon | No | The initial state should be indicated by RRC |
| Samsung | No | Indicated by RRC |
| Nokia | - | PSI-based discard is provisioned to be used in congested links and it should be initially deactivated if no explicit indication.  Ok with explicit indication in RRC as well. |
|  |  |  |

**Summary:**

(to be added after the discussion)

3.4 Modulus operation on non-integer DRX cycles

RAN2 have agreed to introduce non-integer DRX cycles for XR services and keep the modulus operations in the legacy DRX formula for the new DRX cycles. To minimize the mismatch between the start times of DRX on duration and XR traffic with non-integer DRX cycles, it is important that the modulus operation with non-integer divisor does not produce any rounding errors. At RAN2#123bis, it has been agreed that “We will have normative text to avoid rounding errors.”

Different options have been proposed in contributions. For example,

* We may only need to have a line in the normative text stating that “The MAC entity shall ensure no rounding error is generated when performing the modulus operation with drx-NonIntegerShortCycle or drx-NonIntegerLongCycle as the divisor.” The exact method to ensure no rounding error can be left to UE implementation. For example, some programming languages support fractional number data types or symbolic computations, which can represent and process non-integer values exactly without rounding errors. This option requires minimal changes to the legacy DRX formula and yet can avoid inter-operability issues.
* If one wants to have more details in the spec to ensure UEs do implement the modulus operation properly, a mathematical formula for modulus operation with non-integer divisor must be clearly specified instead of leaving it to UE implementation. For example, it is suggested in [1] that modulus (A, B) can be implemented by A - floor (A/B) × B. It is suggested in [2] that the least common multiples method may also be used, i.e. A modulus (B/C) = (A × C/C) modulus (B/C) = [(A × C) modulus B] / C, where both B and C are integers. For example, if frame rate is 60 fps or DRX cycle is 50/3 msec, then B = 50 and C = 3.

**Question 11. Which one of the following options do you prefer to capture the agreement that “We will have normative text to avoid rounding errors.”?**

* **Option 1. Add a line in the normative text after the DRX formula stating that “The MAC entity shall ensure no rounding error is generated when performing the modulus operation with drx-NonIntegerShortCycle or drx-NonIntegerLongCycle as the divisor.” The exact method to implement the modulus operation without rounding error is left to UE implementation.**
* **Option 2. Specify in the normative text that the modulus operation with non-integer DRX cycles shall be implemented by modulus (A, B) = A - floor (A/B)** × **B.**
* **Option 3. Specify in the normative text that the modulus operation with non-integer (ratio between integers) DRX cycles shall be implemented by modulus (A, B/C) = [(A** × **C) modulus B] / C.**
* **Option 4. Please describe your own preferred method in your comment.**

|  |  |  |
| --- | --- | --- |
| **Company** | **Option 1/2/3/4** | **Comments** |
| LGE | Option 1 or Option 3 | No strong view |
| Apple | Option 1 | We prefer not to impose too many restrictions on UE implementation. |
| Huawei, HiSilicon | Option1 |  |
| Samsung | Option 1 | This option allows different implementations. |
| Nokia | Option 3 | Option 1 is not enough as it is not easy to test. Option 2 has issues as explained in our Tdoc R2-2310686. |
|  |  |  |

**Summary:**

(to be added after the discussion)

## 3.5 Range of the new BSR table

For the maximum buffer size in the new BSR table, a number of options have been proposed in the contributions, which are listed below (the list may not be exclusive):

* It can be determined based on the maximum bit rate and lowest frame rate (e.g. which are specified in the SA4 TR) [3][7];
* It should be based on the maximum PDU size [5];
* It should be the same as the maximum of the legacy BSR table [6].

**Question 12: Please indicate which option you prefer for determining the maximum buffer size for the new BSR table?**

**- Option 1: it can be determined based on the maximum bit rate and lowest frame rate [3][4][7]. (Note: For now, we do not need to emphasize the exact formula for using these two parameters);**

**- Option 2: it can be based on the maximum PDU size [5];**

**- Option 3: it is the same as the maximum buffer size in the legacy BSR table [6];**

**- Option 4: other (please describe your preferred method in your comment).**

|  |  |  |
| --- | --- | --- |
| **Company** | **Option 1/2/3/4** | **Comments** |
| LGE | Option 1 | The exact value could be further updated based on the frame rate for AR UL traffic, depending on the SA4 discussion. |
| Apple | Option 3, but can follow majority | We prefer Option 3 as it minimizes specification efforts, and the new BS table could be used for different types of traffics other than XR, so it is a bit restrictive to specify the BS table with considerations of XR use cases only. |
| Huawei, HiSilicon | Option1 | Anyway, we have the legacy table to fall back to. The range should cater for the XR services |
| Samsung | Option 3 | Option 1 & 2 only reflect a single QoS flow case, but one LCG can include multiple LCHs, and hence multiple QoS flows. |
| Nokia | Option 1 | We should also consider the PDB and the number of full frames that can be in the buffer at any given time given certain data and frame rates. |
|  |  |  |

**Summary:**

(to be added after the discussion)

The following is a list of different proposals from the contributions (the list may not be exclusive) for determining the minimum buffer size of the new BSR table:

* Option 1: it can be determined based on the minimum bit rate and highest frame rate (e.g. which are specified in the SA4 TR) [3][4][7];
* Option 2: it should be the code point at which quantization error starts to ramp-up sharply or becomes intolerable [5][6].

**Question 13: Please indicate which option you prefer for determining the minimum buffer size for the new BSR table?**

**- Option 1: it can be determined based on the minimum bit rate and highest frame rate (Note: For now, we do not need to emphasize the exact formula for using these two parameters);**

* **Option 2: it should be the code point at which quantization error starts to ramp-up sharply or becomes intolerable [5][6];**
* **Option 3: other (please describe your preferred method in your comment).**

|  |  |  |
| --- | --- | --- |
| **Company** | **Option 1/2/3** | **Comments** |
| LGE | Option 1 |  |
| Apple | Option 2 | In our understanding, we introduce the new BS table because we cannot tolerate the quantization error caused by the legacy BS table. Also, there is no need to tightly couple the new BS table with XR traffics, as the new BS table could be applied for other use cases. |
| Huawei, HiSilicon | Option1 | Same rationale as above |
| Samsung | Option 1 |  |
| Nokia | Option 1 | Option 2 it is unclear to which point the “error starts to ramp-up sharply or becomes intolerable” since we did not define any target quantization error and for exponential the error rate is kind of fixed. |
|  |  |  |

**Summary:**

(to be added after the discussion)

# 4. Reference

1. R2-2310929, Remaining issues for C-DRX in XR, MediaTek Inc, Oct 2023.
2. R2-2309486, Power saving enhancements for XR, Qualcomm Incorporated, Oct 2023.
3. R2-2309487, BSR enhancements for XR, Qualcomm Incorporated, Oct 2023.
4. R2-2310068, Honor, Discussion on BSR and DSR enhancements for XR, Oct 2023.
5. R2-2310109, BSR enhancements for XR, ZTE Corporation, Sanechips, Oct 2023.
6. R2-2310331, BSR Enhancements for XR, Apple, Oct 2023.
7. R2-2310687, BSR Enhancements for XR, Nokia, Nokia Shanghai Bell, Oct 2023.