3GPP TSG-RAN2 Meeting #121bis-e R2-2204394

eMeeting, 17~26 April 2023

Agenda Item: 7.5.4.1

Source: Qualcomm

Title: Summary of [AT121bis-e][212][XR] BSR solutions (Qualcomm)

Document for: Discussion and Decision

# **Introduction**

This report provides a summary of the following at-meeting email discussion:

* [AT121bis-e][212][XR] BSR solutions (Qualcomm)

Scope: Attempt to find out which among the BSR table solutions have most support and preclude those with least support (if possible). Should discuss pros and cons of each solution and determine which are acceptable to companies (and why). Can also discuss other general details (e.g. how the BSR tables are used).

Intended outcome: Discussion report in [R2-2304394](https://www.3gpp.org/ftp/TSG_RAN/WG2_RL2/TSGR2_121bis-e/Docs/R2-2304394.zip).

Deadline: Deadline 2

During the online discussion on Monday, three solutions for BSR table enhancements were discussed:

* [1] proposes that a basic set of BSR tables can be pre-defined to support common use cases. But it also allows network to RRC configure additional BSR tables on demand, e.g. based on UE’s traffic characteristics.
* [2] proposes that UE generates a new BSR table by applying a scaling factor to a pre-defined reference BSR table. The scaling factor is RRC configured by network.
* [3] proposes that UE can send up to two BSR MAC CEs in single PUSCH transmission for a pending BSR. The first BSR MAC CE indicates a coarse value of UE’s buffer size, and the second BSR MAC CE refines the value reported by the first BSR. The two BSRs may or may not use different BSR tables.

Although these three solutions share the same goal of reducing quantization errors of BSR, they do differ in various ways and have their own advantage and disadvantages. In the following, we first discuss their pros and cons, on aspects such as whether they are efficient in reducing quantization error (e.g. weighing their achievable levels of quantization error vs overhead they introduce), their impacts on network’s flexibility in scheduling and complexity of UE implementation, etc. In the second half of this discussion, we then discuss other general but related issues for new BSR tables.

# **Contact information**

|  |  |
| --- | --- |
| Company | Name (Email) |
| Qualcomm | Linhai He (linhaihe@qti.qualcomm.com) |
| Nokia | Chunli Wu (Chunli.wu@nokia-sbell.com) |
| ZTE | eswar.vutukuri@zte.com.cn |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# **Discussion**

One key difference between [3] and [1][2] is their overall approach in reducing quantization error. [3] uses more bits (up to two BSRs) to encode buffer size. Whereas [1][2] always sends only one BSR but UE may use a new BSR table with smaller quantization error.

**Q1. Which of the following two options do you prefer for reducing quantization error in BSR?**

* Option 1a. UE always sends only one BSR. UE may use either the legacy BSR table or a new BSR table with smaller quantization error. UE chooses which BSR table to use based on its buffer size, e.g. use a new BSR table if its buffer size is within the range of the new BSR table or use the legacy BSR table instead.
* Option 1b. UE may send up to two BSR MAC CEs in one PUSCH transmission. These two BSRs are coupled, i.e. the first BSR indicates a coarse value of UE’s buffer size, and the second BSR refines the value reported by the first BSR. *Without loss of generality, let us assume in this discussion that either of these two BSRs can be based on either the legacy or a new BSR table.*
* Option 1c. UE sends only one BSR MAC CE in one PUSCH transmission, but the UE may report the overall buffer sizes for one LCG with two buffer size values in the BSR MAC CE: the first buffer size value indicates a coarse value of the LCG’s buffer size, and the second BSR refines the first buffer size. *Without loss of generality, let us assume in this discussion that either of these two buffer size values can be based on either the legacy or a new BSR table.*

In addition, the rapporteur suggests companies to discuss the pros and cons of these two options in the comments, e.g. whether it is more efficient than the other in reducing quantization error, its impact on network’s flexibility in scheduling and complexity of network’s UE implementation, etc.

|  |  |  |
| --- | --- | --- |
| **Company** | **Your preference**  (Option 1a/b) | **Comments**  (e.g. Pros and cons of these two options) |
| Qualcomm | Option 1a | Regarding Option 1b:   * If only legacy BSR tables are used, the maximum quantization error of the first BSR is ~10%, then the use of 2nd BSR can reduce it down to 1%. According to SA4’s TR, at 4K and 90 fps, the range of burst size is 28~208 KB. 1% of that corresponds to 280B ~2KB, which is at most one full PDCP PDU. So we are not sure if such a fine resolution in reporting is necessary or not, especially considering the extra overhead it introduces. * If new BSR tables are introduced, then a single BSR can offer a sufficiently good performance. Using the example above again, if the new BSR table uses linear distribution of 256 code points, then the resulting quantization error is 109B~810B, which is smaller than that of using two BSRs.   Based on the above analysis, we can see that a single BSR with a properly designed new BSR table in Option 1a can offer sufficiently good performance or beats the performance of Option 1b that uses legacy tables. Moreover, Option 1a has less UL overhead and is easier for network to decode received BSRs. Therefore, in our view, it is a better option to consider. |
| Nokia | See comments | Maybe better to have separate discussions about whether to allow fallback to legacy table and whether to allow two BS for an LCG as they are different issues.  Fallback to legacy table is at least needed if the new table does not cover full range (depends on the answer to Q3).  Two BS for an LCG could be used to reduce quantization error on top regardless of which table is used since the quantization error always remains. |
| ZTE | 1b/1c | In our view, this provides a simple mechanism to ensure an upper bound on the quantization error regardless of the new table design.  Firstly a “properly designed” new BSR table considering just the most likely application data packet sizes may not be sufficient to minimize quantization errors because the buffered data also depends on past scheduled data and newly arriving data. This will unfortunately require optimisation across the entire BSR range. As such covering all ranges of data sizes (especially now that we go towards even higher data sizes) with a very fine granularity would be impractical.  The main point is that eventually there will be gaps in these tables and it can happen that UE’s buffered data falls in these gaps, there by leading to quantization errors. This quantization error can be reduced by increasing the number of tables, but there will still be worst case values which will result in high degree of over reporting.  So, we think the solution proposed here to reduce the quantization error, especially when the quantization error exceeds some threshold (chosen by the network), is a simple way of achieving the goal.  It should be noted that the second index need not be always present. But, if the error is high, by including very few bits, we can reduce it to almost zero. With this approach, we also think we don’t need to define a large number of additional tables. And we can also have a simple table design which just covers a few additional high data code points. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Summary

(to be added later)

There have been different proposals on how new BSR tables may be introduced. For example, they may be pre-defined in specifications, generated on demand based on parameters configured by RRC, or a combination of these two approaches.

**Q2. Which of the following option(s) do you prefer for introducing the new BSR table(s)?**

* Option 2a. They are pre-defined in the spec;
* Option 2b. They are generated on demand based on a pre-defined formula whose parameters are RRC configured by network;
* Option 2c. Option 2a + 2b, i.e. a basic set of BSR tables can be pre-defined in the spec to cover common use cases, but network can configure additional BSR tables using one of the methods in Option 2b.
* Option 2d. They are generated based on a reference BSR table and a scaling factor RRC configured by network.

You may choose more than one option from the above in your reply. If possible, please also include your analysis on the pros and cons of these four options in your comment.

|  |  |  |
| --- | --- | --- |
| **Company** | **Your preference**  (Option 2a/b/c/d) | **Comments**  (e.g. Pros and cons of these options) |
| Qualcomm | Option 2a or 2c | From UE’s perspective, Option 2a is the simplest for UE to implement and yet serves the purpose well. Since new BSR table(s) only need to cover the size range of common XR encoding rates and frame rate, which are known, BSR tables can be predefined accordingly.  We understand that Option 2b can provide more flexibility for network. And if done right, it may be able to achieve lower quantization errors too. However, given the fact that the target range for new tables are known, we are not sure how much gain (e.g. in term of capacity improvement) Option 2b can offer and whether that would justify the extra implementation effort by UE. And the worst concern for UE implementation is that it is uncertain how much computing cycles it needs to budget for dynamic BSR table generation, because we don’t know how often network may ask UE to generate a new BSR table.  Therefore, Option 2c can be a good compromise for UE and network, because pre-defined BSR tables can help handle most of the scenarios and UE only needs to generate a new BSR table occasionally.  Option 2d can be an alternative to 2c if all the parameters of the reference table (e.g. min, max, distribution of its code points) can scale in the same way when encoding/frame rate changes. But that assumptions needs to be fully vetted before it can be considered. |
| Nokia | Option 2b or 2d | More flexible to cover all typical data rate and frame rate than 2a, without too much UE complexity.  No simplification for UE implementation with 2c compared to 2b since the UE would anyway need to implement both. |
| ZTE | Option 2a | We prefer a single additional table with focus on larger data packet sizes. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Summary

(to be added later)

To either pre-define or RRC configure a new BSR table based on a formula, one needs to decide on three factors: the range of buffer sizes in a table, number of code points, and the distributions of code points within the range. Some of these factors may need to be considered together. For example, the choice in number of code points may affect the choice on the range of a table, and vice versa. Or the choice in the distribution of code points may depend on the choice in the range or number of code points, and vice versa. We discuss these issues in the following.

For the range, the rapporteur thinks that there can be at least two possible options: either reuse the same range of the legacy BSR table or define a narrower range, e.g. based on the sizes of data bursts produced based on common XR encoding rates and frame rates. In the first option, quantization error can be reduced through techniques such as use of more code points or more efficient distribution of code points.

**Q3. What range of buffer sizes should new BSR table(s) have?**

* Option 3a. Reuse the same range of the legacy BSR table;
* Option 3b. A narrower range, e.g. based on the sizes of data bursts produced based on commonly used XR encoding rates and frame rates
* Option 3c. It depends on other options. No need to impose anything for now.

|  |  |  |
| --- | --- | --- |
| **Company** | **Your preference**  (Option 3a/b/c) | **Comments** |
| Qualcomm | Option 3b | We think there are two possible dimensions in reducing quantization errors: reduce the range of a table vs increase number of code points. Between these two choices, we think increasing number of code points is less desirable, because it will increase UL overhead and make the design of new BSR MAC CE more complicated. On the other hand, reducing the range of a table has much less overall impact on the current BSR framework. |
| Nokia | Option 3b | Finer granularity with narrower range. No need to cover full range as legacy table can be used for smaller buffered data, otherwise if with 3a it would have worse granularity for some code points than legacy table if to have finer granularity for others? |
| ZTE | Option 3b | As shown in our contribution, the problem is higher towards higher BSR indices (when there is more buffered data). This is because the code points are sparser in this region. So, targeting these regions seems to make sense. On top, if we have some fail-safe mechanism to ensure that the quantization error never exceeds a given value (like including second index if it exceeds), then the design can simply focus on the higher end of the buffer sizes (and typical frame sizes for XR traffic etc). If we have no such fail-safe mechanism, some more detailed analysis may be needed to see how to optimise over entire range. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Summary

(to be added later)

For the number of code points, the rapporteur thinks that there can be at least two possible options (for both RRC configured and predefined tables): all new BSR tables have the same number of code points or different new BSR tables may have different number of code points. The first option would simplify the design and implementation of the enhanced BSR MAC CE, whereas the second option maximizes the flexibility in defining/configuring new BSR tables.

**Q4. Which of the following is your preferred option for the number of code points in a new BSR table?**

* Option 4a. All new BSR tables have the same number of code points;
* Option 4b. Different new BSR tables can have different number of code points (e.g. depending on their ranges);
* Option 4c. Other (Please provide details in your comment)

|  |  |  |
| --- | --- | --- |
| **Company** | **Your preference**  (Option 4a/b/c) | **Comments** |
| Qualcomm | Option 4a | This choice is a tradeoff between performance and complexity. Theoretically, Option 4b probably has a better performance in reducing quantization errors than Option 4a, because tables with large ranges can benefit from having more code points. However, if network can configure different LCGs to use different BSR tables, then having different BS field lengths for different LCGs can make the format of the new BSR MAC CE much more complicated 🡪 not desirable for UE implementation. |
| Nokia | Option 4a | 8-bit table(s) are enough. |
| ZTE | 4a | We think just one additional table would be sufficient. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Summary

(to be added later)

For the distribution of code points, three options have been proposed in contributions: exponential (as in legacy, the ratio between a step size and its associated buffer size is a constant across all code points), linear (step size for each code point is a constant), and truncated Gaussian [2]. A sensible choice in the distribution of code point may depend on factors such as range and number of code points of a BSR table, as well as traffic characteristics (e.g. size distribution of data burst).

**Q5. Which of the following is your preferred option for the distribution of code points for new BSR table(s)?**

- Option 5a. Exponential distribution, i.e. The same as in legacy;

- Option 5b. Linear distribution, i.e. equal interval between any two consecutive code points;

- Option 5c. Truncated Gaussian distribution;

- Option 5d. Other (Please provide details in your comments).

You may choose more than one option from the above. In that case, please provide the criteria for each selected option in your comment.

|  |  |  |
| --- | --- | --- |
| **Company** | **Your preference**  (Option 5a/b/c/d) | **Comments** |
| Qualcomm | Option 5a and 5b | In our understanding, exponential distribution is better suited for a large range (e.g. across several orders of magnitude), whereas linear distribution is better suited for a small range. Hence different new BSR tables may benefit from using different distributions.  Our understanding on truncated Gaussian distribution is that it is just a model used in RAN1’s evaluation study. It needs to be vetted whether it matches well with actual XR traffic generated by different codec algorithms and how forward compatible it can be. |
| Nokia | Option 5b or 5c | Depends on the option adopted for question 2.  5b with linear distribution is simpler for 2b with formula-based calculation.  5c could match with the traffic distribution better for 2d with scaling on top of the reference table without the need to define the Gaussian formula. |
| ZTE | 5a | Some exponential distribution optimising for higher data sizes would be suitable. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Summary

(to be added later)

There are a number of contributions on the granularity for using new BSR table(s). Most of them have proposed that network can configure on a per LCG basis which BSR table(s) UE should use, e.g. LCG #1 may use the legacy BSR table but LCG #2 may use one of the new BSR tables, and so on. On the other hand, it is also possible that in some solutions, it may be simpler for all LCGs in a BSR MAC CE to use the same BSR table.

**Q6. Which of the following is your preferred granularity for using new BSR table(s)?**

- Option 6a. Network can configure which BSR table(s) (either legacy or new) an LCG should use;

- Option 6b. All LCGs in a BSR MAC CE use the same BSR table;

- Option 6c. Other (Please provide details in your comment)

|  |  |  |
| --- | --- | --- |
| **Company** | **Your preference**  (Option 6a/b/c) | **Comments** |
| Qualcomm | Option 6a | Different LCGs can have different data rates and different burst sizes. It hence makes sense to configure BSR table on a per LCG basis. |
| Nokia | Option 6a | Per LCG makes sense as not all the LCGs are for XR and different LCHs/LCGs might have different data rate. |
| ZTE | Option 6c | We prefer that any new BSR mechanism would be per LCG (same as legacy). Perhaps this could be agreed as an independent agreement regardless of other enhancements.  But considering that the buffer size cannot be predicted accurately, which BSR table(s) is used should be selected based on the Buffer size to be reported.  And the table used is identified by the LC-ID. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Summary

(to be added later)

In legacy, short BSR and long BSR use different BSR tables, because they use different number of code points. If we are going to introduce new BSR tables, then we need to discuss whether/how new BSR tables should be designed for them.

**Q7. Which of the following is your preferred option for introducing new BSR table(s) for short/long BSR?**

- Option 7a. Only long BSR need to have new BSR table(s);

- Option 7b. Only short BSR needs to have new BSR table(s);

- Option 7c. Both short BSR and long BSR can have their own new BSR table(s), which are defined/configured separately;

- Option 7d. The same set of new BSR table(s) are used by both short BSR and long BSR.

- Option 7e. Introduce new BSR formats to accommodate new BSR table(s).

|  |  |  |
| --- | --- | --- |
| **Company** | **Your preference**  (Option 7a/b/c/d) | **Comments** |
| Qualcomm | Option 7a | First, we think it is useful to keep the 2B short BSR MAC CE. For small bursts, the current short BSR table is sufficient, because its distribution of code points has fine resolution at the low end.  To cover the case where a single LCG has a large burst, we think RAN2 should agree that UE is allowed to use long BSR in that case. Then new BSR table(s) designed for long BSR can be used for to provide better resolution for large bursts, if needed. |
| Nokia | Option 7a | Long BSR provides finer granularity since it provides a lot more code points compared to short BSR, leading to minimizing the quantization error. |
| ZTE | 7a |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Summary

(to be added later)

Last but not least, there was discussion near the end of the online session on whether new BSR table(s) is available only to XR UEs or to any UEs. Let us continue that discussion here to collect more views.

**Q8. Do you think new BSR table(s) is available only to UEs supporting XR services or to any UEs?**

- Option 8a. Only UEs supporting XR services;

- Option 8b. Any UEs

|  |  |  |
| --- | --- | --- |
| **Company** | **Your preference**  (Option 8a/b) | **Comments** |
| Qualcomm | Option 8b | We do not see any strong reasons why a new BSR table cannot be used by UEs not supporting XR services. Moreover, use of new BSR tables is fully under network control, i.e. if network does not want a UE to use a new BSR table, it can simply not enable or configure that BSR table for the UE. |
| Nokia | Option 8a for now | We can start with 8a when designing the parameters/values for the new table(s).  It can be discussed later if need to apply to other UEs. |
| ZTE | 8b | As normal, we assume the UE will indicate support for these and if supported, the network can configure the UE to use these whenever it is appropriate. Whether an XR service is running at this point or not may be irrelevant (what matters is the configuration that the UE receives). |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Summary

(to be added later)

# **Conclusions**

(To be added later)

# References

1. R2-2302515, BSR enhancements for XR, Qualcomm Incorporated.
2. R2-2303862, BSR enhancements for XR, Nokia, Nokia Shanghai Bell.
3. R2-2302851, BSR enhancements for XR, ZTE Corporation, Sanechips.