**3GPP TSG-RAN WG2 #110b-e *R2-200xxxx***

**E-meeting, June 2020**

Agenda Item: x.x.x.x

Source: OPPO

Title: Summary of [Post109bis-e][954][V2X] SIB12 Overhead Reduction (OPPO)

Document for: Discussion, Decision

# Introduction

RAN2 achieved following agreement at RAN2#109bis meeting:

Agreements on SIB12:

1: Introduce segmentation of SIB12 in RRC layer for both NR and LTE system.

Furthermore following email discussion is agreed to discuss potential optimization solution to reduce SIB12 size:

[Email discussion]: To discuss and conclude other options proposed in R2-2004075. If we cannot see majority companies’ views for each sub-options or if we cannot have reliable draft CR, we will rely on segmentation of SIB12 only in Rel-16 (OPPO)

# Discussion

The size of SIB12 is mainly affected by those IEs which are either too big or repeat too many times or both. How many times an IE will repeat depends on either the length of the list or how deep the IE is buried within the whole IE structures. Taking IE SL-QoS-Profile-r16 for example, it will repeat NrofSLRB-r16\* NrofSL-QFIs-r1 times (Note1). For IE SL-Priority-TxConfigIndex-r16 it will repeat NrofFreqSL-r16\* NrofSL-BWPs-r16\*( NrofTXPool-r16+1)\*8.

*Note: the parameter e.g.* **NrofSLRB-r16** *in the paper is the configured length of the IE list whose maximum number is* ***max*NrofSLRB-r16***. The same assumption is taken for other similar parameters.*



Figure 2-1 IE structure of SIB12 (partial IEs)

In Figure2-1 the IEs which impacts SIB12 size at most are listed. And the following table show the IE name, their repeat factors and estimated size:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **level 1 IE list** | **level n IE list** | **Size of IE list(bit)** | **Repeat factors** | **Minimum factors** | **Typical factors** |
| sl-FreqInfoList-r16 | IE size of sl-FreqInfoList-r16 except for below 5 IEs | 274 | 1 | 1 | 1 |
|  | SL-SyncConfig-r16 | 90 | NrofFreqSL-r16\*SL-SyncConfig-r16 | 1 | 4 |
|  | **SL-ResourcePool-r16(TX) except for**  **sl-CBR-Priority-TxConfigList-r16**  **and sl-ThresPSSCH-RSRP-List-r16** | **487** | **NrofFreqSL-r16\* NrofSL-BWPs-r16\*( NrofTXPool-r16+1)** | **2** | **5** |
|  | **sl-CBR-Priority-TxConfigList-r16** | **448** | **NrofFreqSL-r16\* NrofSL-BWPs-r16\*( NrofTXPool-r16+1)** | **2** | **5** |
|  | **sl-ThresPSSCH-RSRP-List-r16** | **496(Note1)** | **NrofFreqSL-r16\* NrofSL-BWPs-r16\*( NrofTXPool-r16+1)** | **2** | **5** |
|  | **SL-ResourcePool-r16(RX)** | **449** | **NrofFreqSL-r16\* NrofSL-BWPs-r16\*NrofRXPool-r16** | **1** | **1** |
| sl-UE-SelectedConfig-r16 | IE size of sl-UE-SelectedConfig-r16 except for below 3 IEs | 11 | 1 | 1 | 1 |
|  | sl-PSSCH-TxConfigList-r16 | 69 | PSSCH-TxConfig-r1 | 1 | 8 |
|  | sl-CBR-RangeConfigList-r16 | 7 | CBR-Config-r16\*CBR-Level-r16 | 1 | 64 |
|  | sl-CBR-PSSCH-TxConfigList-r16 | 46 | TxConfig-r16 | 1 | 16 |
| sl-NR-AnchorCarrierFreqList-r16 |  | 22 | FreqSL-NR-r16 | 1 | 1 |
| sl-EUTRA-AnchorCarrierFreqList-r16 |  | 14 | FreqSL-EUTRA-r16 | 1 | 1 |
| sl-RadioBearerConfigList-r16 | IEs except for below 2 IEs | 17 | NrofSLRB-r16 | 1 | 12 |
|  | **SL-QoS-Profile-r16** | **129** | **NrofSLRB-r16\*NrofSL-QFIs-r1** | **1** | **48** |
|  | sl-PDCP-Config-r16 | 9 | NrofSLRB-r16 | 1 | 12 |
| SL-RLC-BearerConfig-r16 |  | 59 | SL-LCID-r16 | 1 | 12 |
| sl-MeasConfigCommon-r16 |  | 98 | NrofSL-ObjectId-r16 | 1 | 1 |
| sl-CSI-Acquisition -r16 |  | 1 | 1 | 1 | 1 |
| sl-ZoneConfig-r16 |  | 11 | 1 | 1 | 1 |
| sl-OffsetDFN-r16 |  | 11 | 1 | 1 | 1 |
| t400 |  | 4 | 1 | 1 | 1 |
| sl-MaxNumConsecutiveDTX-r16 |  | 3 | 1 | 1 | 1 |
| sl-SSB-PriorityNR-r16 |  | 3 | 1 | 1 | 1 |

Table2-1

*Note1: the size is calculated assuming CBR-Level-r16=8 (maxCBR-Level-r16=16)*

*Note2: the IE sizes in table 2-1 are adjusted according to endorsed 38.331 CR in R2-2004072*

# Optimization of SIB12 size issue

As indicated in table 2-1, the IEs in **bold red** are either of big size and/or repeated many times. Following parameters are assumed to estimate the SIB12 size:

|  |  |
| --- | --- |
| **Configured** parameters | Typical values |
| NrofFreqSL-r16 | 1 |
| NrofSL-BWPs-r16 | 1 |
| NrofTXPool-r16 | 4 |
| NrofRXPool-r16 | 1 |
| NrofSLRB-r16 | 12 |
| NrofSL-QFIs-r1 | 4 |
| SL-LCID-r16 | 12 |

Table 2.1-1

The corresponding repeat factors are listed in the column “typical factors” of table 2-1. Based on this assumption SIB12 is estimated as 17360 bits. If some measures e.g. proposal 1/2/3 from paper [2] are taken into account then the SIB12 size can be reduced to be 8022 assuming:

1, IE SL-QoS-Profile-r16 is 33 bits assuming standardized PQI instead of non-standardized PCI is configured and IE sl-GFBR-r16 and sl-MFBR-r16 are both reduced from 33 bits to 6 bits as indicated in draft CR [4]

2, Table plus index approach is taken for IE SL-QoS-Profile-r16 where the length of the table is 16

3, IE SL-ThresPSSCH-RSRP-List-r16 and SL-CBR-Priority-TxConfigList-r16 is configured as cell level IE hence only one instance is counted instead of 5

The idea from [3] is to preconfigure *sl-RadioBearerConfigList-r16* and *sl-RLC-BearerConfigList-r16*, in that case the SIB12 size can be further reduced to 6474 bits.

Figure 2.1-1 SIB12 size reduction

The outcome in Figure2.1-1 shows there is a big room for size optimization. In future the number of SLRB may not increase dramatically but the number of frequencies and the number of BWP per frequency will like increase when more spectrum are available for sidelink operation. In that case SIB12 size will increase mainly because of IE like TX resource pool.

**Q1: Do you agree that SIB12 size need be optimized?**

If you disagree, please elaborate your detail reason

|  |  |  |
| --- | --- | --- |
| Company | Agree/disagree | comments |
| Ericsson | Disagree | Our view is that the segmentation introduced for SIB12 it solve the problem. Optimizing the size of SIB12 is an optimization that is not required at this stage, considering that we have only one meeting left to close the release and freeze the ASN.1. This mean that whatever is not done/agree in the next meeting, it will result in a painful not backward compatible change that we want to avoid.  We believe that the optimization of SIB12 can be left to TEI17 or Rel-17. Since this topic it deserve an extensive discussion and an impact analysis of all the mentioned options, there is no rush to have a not so efficient solution now. |
| MediaTek | Yes | As discussed in RAN2#109bis-e, we consider it important not only to make it physically possible to deliver the SIBs over the air (which segmentation can do), but to restrain the total size for overhead and acquisition reasons.  We don’t think this issue can be left to Rel-17, because Rel-16 UEs still need to be able to interpret SIB12. If we e.g. raise some IEs from resource pool level to cell level in Rel-17, we would still have to signal them at resource pool level for the benefit of Rel-16 UEs. |
| Qualcomm | See comments | Since RAN2 has agreed to segmentation, we have a solution to SIB12 size. Given the limited time, we recommend not pursuing further optimizations at this point.  Discussion of additional optimizations can be deprioritized until remaining open issues have been resolved. |
| Futurewei | Low priority for Rel-16 | While we appreciate the downside of current SIB12 structure, we don’t think the limited time remaining in Rel-16 would afford us a thorough consideration in optimizing SIB12. We are, hence, only open to some low hanging fruit, i.e., some targeted, high impact (on size reduction) modification, if it can be agreed without consuming too much RAN2 time.  Given the segmentation approach is supported for SIB12, further optimization of SIB12 can be postponed to Rel-17. Therefore, any effort on SIB12 optimization in Rel-16 should be of low priority. |
| Intel | Disagree | We tend to agree with the comments made by Ericsson and Qualcomm that since we already discussed and agreed to segmentation (even if it is subpotimal) and the fact that we have very little time to discuss detailed solutions, we can deprioritize optimization discussions to future releases |
| vivo | Disagree | Although we have some sympathy with companies’s efforts to further optimize the situation here for SIB size reduction, that only one meeting left makes it difficult to have some strightward and efficient solutions, let alone that some of these proposed solutions have to be coordinated and confirmed by RAN1 (e.g. if we would like to change some parameters from pool-level to BWP/carrier-level)  Therefore we suggest not to pursue this optimization at this stage and leave it to further release. |
| Nokia | Disagree | In R2#109bis meeting there was consensus on “no further optimization in Rel 16”. Furthermore, on the SIB12 size issue it was agreed in last R2 meeting (109bis) that SIB12 segmentation is supported, which fully resolves the problem. We are open to discuss and analyze further optimization for SIB12 size reduction in Rel. 17. |
| CATT | Yes | Considering the maximum SI message size is 2976 bits, assuming no optimization is performed, the SIB12 should be segmented into 17360 bits/2976 bit≈6 segments. Considering the number of frequencies and the number of BWPs per frequency should be increased in the further, more segments will be needed. In order to ensure the reception reliability of each segment, it will consume much more time. Hence, SIB12 optimization to reduce the segment number is preferred. |
| Huawei | Disagree | Share the view with Ericsson, Intel, Vivo, Nokia, Qualcomm, Futurewei. As we already agreed to apply SIB segmentation, any oher size optimization is then pure “OPTIMIZATION” on top of that. This is against what we agreed in the last meeting that any further optimization is not pursued in this Rel-16.  Also, we are not convinced that even with segmentation, the SIB size is still serveral times bigger than the SI size limitation. How fine granularity do companies expect for the configurations included in the SL specific SIB? Note that SIB just provides SL configurations at cell level which are just used by IDLE/INACTIVE UEs with mode-2. No matter how many configurations are included in the SIB, the performance of IDLE/INACTIVE mode-2 anyway cannot be guaranteed (especially for the GBR services). To this end, what sense does it make to include in SIB much more configurations than just necessary/sufficient for basic mode-2 operations (e.g. those already supported in LTE SIB21)?. |
| Samsung | Yes | According to the analysis upto now, segmentation is a mandatory for NR SL. Then UE should keep its buffer to cover the maximum number of segments. Also more segments will cause more latency in SL operations.  We understand the limited time cannot cover all the possibilities but RAN2 may consider some options in 2.2 |
| Lenovo | Disagree | As the discussion in RAN2#109b e-meeting, the total size of SIB 12 still cannot be controlled in accepteable range by taking both the optimization into account, so segmentaion of SIB 12 has been introduced in RRC layer for both NR and LTE. From our view, the approved segmentation can solve the SIB size overhead already, there seems no reason to specify two solutions for the same problem, then SIB size optimization cannot be a essential issue in Rel-16, we prefer not to consider optimization of SIB12 size in this relaese. |
| LG | Disagree | In R2#109bis meeting there was consensus on “no further optimization in Rel 16”. Further optimization for SIB12 size reduction can be discussed in Rel 17. |
| ZTE | Disagree | We think the SIB12 segmentation already solve the problem. And due to the limited time budget, we can discuss the optimization in next release. |
| OPPO | Agree | We think segmentation approach resolve the problem of broadcast of SIB12, but it doesn’t reduce any signaling overhead. On one hand we agree not so much time left for optimization discussion. On the other hand, this is a “go or no go” moment i.e. optimization in later release is not feasible due to backwards compatibility issue. |
| Apple | Agree | Optimization in later release(s) will be even harder. RAN2 is responsible to reduce the “fat” and have a lean version of SIB in the first release of NR sidelink |

# Options to reduce SIB12 size

The key IE impacting SIB12 size are SL-QoS-Profile-r16, sl-CBR-Priority-TxConfigList-r16 and sl-ThresPSSCH-RSRP-List-r16 etc. During offline discussion [1] there are several solutions are listed on the table and they are not exclusive with each other by the nature.

Option 1: To introduce table + index approach for e.g. SL-QoS-Profile-r16

Option 2: To adjust the granularity of value range for e.g. sl-GFBR-r16 and sl-MFBR-r16

Option 3: To modify the depth of the IE within ASN.1 structure e.g. to configure sl-CBR-Priority-TxConfigList-r16 and/or sl-ThresPSSCH-RSRP-r16 per cell IEs or per BWP or per frequency IE

Option 4: to preconfigure IE e.g. for SL-QoS-Profile-r16

Option 5: to preconfigure IE but with default value in case of absence in the SIB e.g. for SL-QoS-Profile-r16

All the solutions try to keep balance between signalling overhead and flexibility.

**Q2: Among listed 5 options, which option(s) do you prefer? Please also explain which IE(s) do you prefer for that selected option**

|  |  |  |
| --- | --- | --- |
| Company | preferred option(s) | Detail explanation |
| MediaTek | Versions of 1/3/5, with comment (apologies for length) | We have some doubts about option 2 because it reduces the signalling flexibility; it would be OK to investigate but may be difficult to agree in the limited time available. And we think option 4 is too dependent on predicting all possible service configurations; we should keep the ability to signal a “custom” configuration when needed.  Regarding the problematic IEs one by one:   * For SL-QoS-Profile-r16, we are not sure if it’s a huge problem in the SIB, because as noted by some companies previously, mode 2 operation for idle/inactive UEs cannot really support GBR services. We see some benefit in a table+index approach for this IE, but it may not be the most critical item. * For sl-CBR-Priority-TxConfigList-r16 and sl-ThresPSSCH-RSRP-r16, it would be necessary to confirm with RAN1 if we can change these to be per cell/BWP/frequency. We assume companies will coordinate internally so that this issue is not a surprise to RAN1 and an understanding can be reached in the May meeting cycle. If the IEs cannot be raised to cell level, we see three options:   + Option 5 above, with preconfiguration as a default value and signalling the IEs when they need to be different from the default.   + Signal the IEs at cell level as default values, and include the whole IE per pool only when it diverges from the default value (like option 5, but with cell-level signalling instead of preconfiguration).   + Table+index approach: Signal a list of values for these IEs at cell level, and signal indices into the list per pool (option 1 described above). * We should also look at SL-PSFCH-Config, a 275-bit string that needs to be included for each resource pool with PSFCH. This probably cannot be raised to cell level, but we could use a table+index approach. Furthermore, even though RAN1 described it as a fixed-length BIT STRING, we understand that it could be coded as variable length (1..275), since the number of PRBs used for PSFCH cannot be larger than the actual size of the pool. This would save significant overhead for smaller pools.   + Further optimisations for this IE could be considered, e.g. indicating a range of PRBs rather than a bitmap (but this assumes that the PRBs for feedback are always contiguous, which would require RAN1 to take the decision). As noted above, a table+index approach could be implemented, but it may have limitations for this IE, since pools of different sizes in the frequency dimension will never have the same string for the PRB bitmap. |
| Qualcomm | Option 1, 4, 5 | Of the suggested options, our preference is to allow pre-configuration. Option 3 is not extensible to future releases, given multiple frequencies may be supported.  More generally, as RAN2 has agreed to support SIB12 segmentation, given the limited time remaining to close the release, we don’t think these optimizations need to be pursued at the present time. |
| Futurewei | 1/3/5 | We don’t think it is wise to limit the signaling flexibility/granularity without comprehensive consideration, e.g., to consult other WGs by LS, which seems not feasible at the very late stage of Rel-16.  This leaves elegant ASN.1 structure optimization as best possibility in Rel-16. This should be of low priority, and really straightfoward for easy agreement. |
| Intel | 4/5 | If we do need to make some optimizations, we think pre-configuration seems to be easiest to consider. Other solutions either require input from other WGs or require more extensive discussion. So, while we think at this stage that segmentation would have to suffice, if there is a majority support for performing further optimizations, pre-configuration can allow to alleviate some of the burden. |
| CATT | Option 3 and 4 | Generally, for all the possibility solutions, our preference is to allow pre-configuration, i.e., Option 4 which is to preconfigure the sl-RadioBearerConfigList-r16, sl-RLC-Bearer ConfigList-r16.  For Option 3, we realize the IEs of sl-CBR-Priority-TxConfigList-r16 and/or sl-ThresPSSCH-RSRP-r16 consume a lot of bits. Considering the number of frequencies and the number of BWPs per frequency should be increased in the further, these IEs within ASN.1 structure should be optimized.  For Option 5, it’s unclear if SL-QoS-Profile-r16 is already pre-configured, why there is still need a default value in SIB? Relying on the pre-configuration for SL-QoS-Profile-r16 is enough for the idle/inactive UE. |
| Samsung | Option 1, 5 | We think that ASN.1 restructuring with table will reduce overhead of SIB. Also, since pre-configured IE is already defined we think that utilizing the pre-configuration is a possible option when default value is needed. |

|  |  |  |
| --- | --- | --- |
| Lenovo | Option1 | If some optimization is inevitable, considering option2 and option3 may sacrifice the configuration flexibility and the coorresponding impacts also should be coordinate with other group to confirm the feasibility; For option4 and option 5, the pre-confoguration cannot detremine all service parameters, they might not be the pratical way to handel this issue. Therefore, option1 is the accepted option for us if we do need to perform the SIB size optimization in this release. |
| LG | 3, 5 if necessary |  |
| ZTE | 5 | To include default SLRB configuration in pre-configuration can help to solve the issue when there is no default SLRB configuration in SIB. However, RAN2 should further dig out how to use pre-configuration together with SIB. I.e. when UE is in-coverage, it should read cell SIB, however, the default SLRB configuration is in pre-configuration. Thus, when UE needs to use default SLRB configuration, it needs to read pre-configuration. Such UE behaviour may cause some 331 spec impact. |
| OPPO | Option1 ,2 and 3 | For option3, we think one compromise is lift it as frequency level IE. Not sure why it is related to RAN1. It is kind of tradeoff between signaling flexibility and overhead hence RAN2 can make decision. |
| Apple | Option 1, 3,4 and 5 |  |

# Conclusion

There are totally 14 companies joined the discussion. 4 companies agree to optimize SIB12 size, 2 companies would take it as low priority issue and the rest 8 companies are not convinced this is essential and hence disagree to do anything in this release. Rapporteur just want to point out that it is not possible to optimize it in future release due to backwards compatibility issue.

Among the solutions, here is statistics:

|  |  |  |
| --- | --- | --- |
| Solutions | Supported companies | Note |
| 1 | 6 | Except for SL-QoS-Profile-r16, Mediatek also mentioned it could be also applicable to SL-PSFCH-Config and sl-CBR-Priority-TxConfigList-r16 and sl-ThresPSSCH-RSRP-r16 |
| 2 | 1 |  |
| 3 | 5 | Mediatech mentioned that RAN1 maybe consult if we change the IE level in the structure |
| 4 | 3 | CATT believe the proper IE should be RadioBearerConfigList-r16, sl-RLC-Bearer ConfigList-r16 |
| 5 | 7 | ZTE think SLRB configuration is right candidate and Mediatek think sl-CBR-Priority-TxConfigList-r16 and sl-ThresPSSCH-RSRP-r16 could be right candidate. The rest companies agree in general |

Table 3-1

Except for listed opinions Mediatek also believe SL-PSFCH-Config could be coded as variable length.

**Recommendation: No optimization on SIB12 size is introduced in Rel16.**

# Reference

[1] R2-2004075 [AT109bis-e][704][V2X] SIB12/28 (OPPO) OPPO

[2] R2-2002651 Open issues on system information OPPO

[3] R2-2002828 Further Discussion on RRC Remaining Issues CATT

[4] R2-2002652\_38331\_CRyyyy\_(REL-16)\_Correct to fix SIB12 size issue for NR V2X.docx OPPO