**3GPP TSG-RAN WG2 Meeting #121 R2-230xxx**

**Athens, Greece, Feb 27 – Mar 03, 2023**

**Agenda item:** 8.2.1

**Source:** Intel Corporation

**Title:** [AT121][411][POS] SLPP specification (Intel)

**Document for:**  Discussion and decision

# Introduction

This is the report of following at meeting offline discussion:

* [AT121][411][POS] SLPP specification (Intel)

Scope: Collect offline comments on the SLPP specification practices.

Intended outcome: Report to CB session

Deadline: Wednesday 2023-03-01 1900 EET

# Contact Information

Respondents to the email discussion are kindly asked to fill in the following table.

|  |  |
| --- | --- |
| Company | Contact: Name (E-mail) |
| Huawei, HiSiclion | Yinghao Guo (yinghaoguo@huawei.com) |
| Lenovo | Hyung-Nam Choi (hchoi5@lenovo.com) |
| Fraunhofer | Birendra Ghimire ([birendra.ghimire@iis.fraunhofer.de](mailto:birendra.ghimire@iis.fraunhofer.de)) |
| Nokia | Stepan Kucera (stepan.kucera@nokia.com) |
| MediaTek | Nathan Tenny (nathan.tenny@mediatek.com) |
|  |  |
|  |  |
|  |  |
|  |  |

# Discussion

As discussed in R2-2300409:

|  |
| --- |
| From procedure perspective, SL-PP is same as LPP protocol, which is different from RRC specification. Therefore the structure of LPP (TS37.355) can be reused as the baseline, e.g. section 4 Functionality of Protocol, section 5 LPP Procedures, and section 6 Information Element Abstract Syntax Definition. Further discussion is needed on what changes should be based on progress on Sidelink positioning.  **Proposal 2: Regarding the structure, e.g. general part, procedure part of SL-PP, Information Element Abstract Syntax Definition, LPP (TS 37.355) can be used as baseline for further discussion.** |

Rapporteur would like to check companies’ view .

**Question 1: Do companies agree the proposal 2 in R2-2300409, i.e.**

**Regarding the structure, e.g. general part, procedure part of SLPP, Information Element Abstract Syntax Definition, LPP (TS 37.355) can be used as baseline for further discussion.**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Remark** |
| Huawei, HiSilicon | Yes, See comments | We have the feeling that the current LPP spec put too much procedural text for the UE to the field description of the LPP ASN.1. it is better to separate these two and leave the field description to its original use |
| OPPO | Yes, but | At this stage, the IEs to be included in each msg should be assume zero from the beginning. |
| Lenovo | Yes with comments | For clause 4 and 5 we are ok to re-use LPP as baseline. Based on the future agreements we make for SLPP we have to sort out later whether we need to deviate from it or not.  For clause 6 we think more discussion is needed, see our comments to Q2. Basically, we prefer to adopt a simpler approach for SLPP ASN.1 compared to LPP or RRC, e.g. we are not convinced in applying delta signaling for SLPP messages. |
| Qualcomm | Yes | The reason why we have detailed field descriptions (instead of much procedural text) is that we have a plurality of positioning methods, which are all optional. Procedural description may get lengthy if handling of fields etc. is described via procedure description like in RRC. We believe the current LPP approach improves readability and reduces spec maintenance effort. |
| Ericsson | Yes, however | It would be good to develop the baseline as separate module within TS 37.355 to begin with. Later we can fork off in separate document.  It would give some insight as which clause can be clearly referred in LPP and which clause needs to be built separately. |
| Fraunhofer | Yes |  |
| Nokia | Yes but | We agree to reuse LPP as baseline but would like to define SLPP concretely in accordance with future agreements, not based on LPP legacy. |
| MediaTek | Yes | Agree with Qualcomm about the usefulness of procedural guidance in field descriptions.  We don’t have a strong view on a separate spec vs. a separate module within LPP, but we understand from plenary discussion that there was a general preference for a separate spec. |

Regarding ASN.1 structure, R2-2300409 discussed:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| For ASN.1 itself, LPP used LTE RRC ASN.1 principle, e.g. Need code, naming convention, extension.   |  | | --- | | The ASN.1 in this clause uses the same format and coding conventions as described in Annex A of TS 36.331 [12]. |   However, there are some differences between LTE RRC and LPP:   * **Difference 1:** constraint is defined within the IE instead of common section in RRC (RRC multiplicity and type constraint values);   From readability perspective, it is easy to check if we put all constraint in the same place;   * **Difference 2**: Fields in the field descriptions is sorted based on presence order instead of alphabet order;   No big difference between RRC approach and LPP approach. Alphabet order is slightly better from readability perspective.   * **Difference 3**: Regarding extension, only “Ellipsis” and “spare” (only message level) are used instead of “nonCriticalExtension” ;   “nonCriticalExtension” in message level can group the IEs introduced in a new release together with less overhead, especially for size critical message, e.g. system information, initial setup, etc. But “nonCriticalExtension” can only be used at the end of the message. We do not see the reason why LPP should abandon it.   * **Difference 4**: setup/release, addition/modification are not used in LPP; However we may consider this in SL-PP since it is related to how to handle the SL-PRS resources.   **Observation 1: RRC approach on common session for constrains, Fields in the field description are sorted based on alphabet order and “nonCriticalExtension” at message level is better. FFS on whether setup release structure should be introduced in SL-PP.**  In Rel-15, follow enhancements are introduced in NR RRC compared with LTE RRC:   * **Difference 1:** Define ASN.1 elements for UE capabilities in a dedicated section (i.e. “UE capability information elements”) separate from “Other Information elements” - section. * **Difference 2:** Need code   Need codes defined for NR RRC should be clearer and unambiguous. (More discussion needed on when modules are used)  3 The use of need codes should be clarified to ensure consistent usage, in particular  a) Need codes should reflect the action performed upon receiving a message with the field absent (rather than the action when the field is not configured)  b) Need codes should distinguish one-shot and regular configuration parameters e.g. by introducing an additional need code.  Finally new Need codes were introduced in NR RRC as   |  |  | | --- | --- | | Need S | *Specified*  Used for (configuration) fields, whose field description or procedure **specifies** the UE behavior performed upon receiving a message with the field absent (and not if field description or procedure specifies the UE behavior when field is not configured). | | Need M | *Maintain*  Used for (configuration) fields that are stored by the UE i.e. not one-shot. Upon receiving a message with the field absent, the UE maintains the current value. | | Need N | *No action* (one-shot configuration that is not maintained)  Used for (configuration) fields that are not stored and whose presence causes a one-time action by the UE. Upon receiving message with the field absent, the UE takes no action. | | Need R | *Release*  Used for (configuration) fields that are stored by the UE i.e. not one-shot. Upon receiving a message with the field absent, the UE releases the current value. |   It would be good to make specification clear, therefore from editor perspective, SL-PP can follow NR RRC on these two new things.   * **Difference 3:** SetupRelease type is introduced. FFS on whether SL-PP needs setup/release compared with LPP.  |  | | --- | | – *SetupRelease* *SetupRelease* allows the *ElementTypeParam* to be used as the referenced data type for the setup and release entries. See A.3.8 for guidelines.  -- ASN1START  -- TAG-SETUPRELEASE-START  SetupRelease { ElementTypeParam } ::= CHOICE {  release NULL,  setup ElementTypeParam  }  -- TAG-SETUPRELEASE-STOP-- ASN1STOP |   **Observation 2: NR RRC approach on Need code, Define ASN.1 elements for UE capabilities in a dedicated section (i.e. “UE capability information elements”) is better.**  **Proposal 3: Regarding the ASN.1 part of SL-PP, follow NR RRC approach, e.g.**   * **Need code** * **Define ASN.1 elements for UE capabilities in a dedicated section (i.e. “UE capability information elements”)** * **Common session for constrains** * **“nonCriticalExtension” at message level** * **Fields in the field description are sorted based on alphabet order** * **FFS on whether setup release structure should be introduced in SL-PP** |

Rapporteur would like to check companies’ view .

**Question 2: Do companies agree the proposal 3 in R2-2300409, i.e.**

**Regarding the ASN.1 part of SLPP, follow NR RRC approach, e.g.**

* **Need code**
* **Define ASN.1 elements for UE capabilities in a dedicated section (i.e. “UE capability information elements”)**
* **Common session for constrains**
* **“nonCriticalExtension” at message level**
* **Fields in the field description are sorted based on alphabet order**
* **FFS on whether setup release structure should be introduced in SLPP**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Remark** |
| Huawei, HiSilicon | Yes, but | In general, a better baseline for the SLPP should be PC5-RRC rather than UU-RRC in TS 38.331.  For the need code, the issue is that it is absent for UL and mandatorily needed for optional DL fields for Uu. While for PC5, there is no UL/DL difference. So the current PC5-RRC is quite unclear in this aspect. Hope the issue can be well addressed in SLPP  For the setupRelease, it should be discussed together with NeedM and AddModList mechanism that allow delta signaling. From spec point of view, SetupRelease has to be supported if we support need M. We also find it beneficial to support delta signaling  The others are OK |
| Lenovo | Partly, see comments | We agree to follow NR RRC approach for:   * Grouping of UE capability IEs into a dedicated section to improve readability. * Use of common/separate section for multiplicity and type constraint values. * Use of “non-critical extension” approach at message level to be code-efficient. * Sorting of field descriptions in alphabetical order.   For the following aspects we need further discussion:   * Other useful grouping of IEs to improve readability. * To introduce procedural descriptions to improve readability. * Delta signaling/need codes. Before going into ASN.1 details we should discuss first whether we should support delta signaling for SLPP messages at all or to which extent. Referring to discussions in the past for NR RRC, specifying need codes can be really a pain. So far, only “Provide assistance data message” may make sense for using delta signaling. We don’t see the need to apply delta signaling for UE capability enquiry/transfer etc. Furthermore, if we support groupcast/broadcast signaling (under the condition that security issues for these cast types can be solved by SA3) at least for broadcast transmission we don’t see the need for delta signaling. In general, compared to NR RRC we prefer to adopt a simpler approach for SLPP ASN.1. |
| Qualcomm | partly | Need Codes: We feel the need codes used in LPP are sufficient for positioning ASN. However, this can be revisited during the work. It may also be sensible to define SLPP specific need codes (i.e., not reuse RRC definition).  UE capabilities: We prefer to do it like in LPP. Each positioning method section is self-contained, including capabilities.  Common section for constrains: O.K. If the spec is getting bigger over time, this improves readability.  “nonCriticalExtension” at message level: Not sure yet, but O.K. as "working assumption."  Fields in the field description are sorted based on alphabet order: O.K. At the beginning of LPP, we didn't anticipate that all the IEs get so many fields added over time…  FFS on whether setup release structure should be introduced in SLPP: O.K. with the FFS. We will see during the work if this is sensible or not. |
| Ericsson | Yes | Agree we should reuse and also agree with Huawei that we need to be careful on need code as it was used for DL message. We should then also follow SL RRC message to follow some guidance on Need code. |
| Fraunhofer | Yes | Agree with the general principle.   * **Common session for constrains**   We presume this was meant as section and not session. We are fine with it.   * **“nonCriticalExtension” at message level**   We are fine with the LPP approach, as well as the proposed RRC-like approach.  Regarding the setup/release: We think this is a useful functionality and support FFS on this. |
| Nokia | Yes but | Views similar to Lenovo |
| MediaTek | Generally yes; see comments | Most aspects are fine. We think the NR need codes (M/R/N/S) are a bit more usable than the LTE need codes (there seems to be a tendency to misinterpret “OP” as “OPtional”, for example).  SetupRelease should be driven by use cases. If we find ourselves writing ASN.1 with the “setup/release branch” idiom, of course it would be good to introduce a parametrized shorthand for it. In general, we should feel free to use parametrized types if we have repetitive or complex ASN.1 structures—we don’t need to be constrained by how RRC does it.  Regarding the section for capability IEs, today we effectively have this per method (the Method-ProvideCapabilities subsections). We tend to think this is a clearer organization, because many capabilities are method-specific and a common capability section will end up with method-specific subsections anyway. Isn’t a reader more likely to want to find capabilities for a specific method than all capability-related IEs? |

# Summary

Based on the input from companies, we have the following proposals:

# Reference

[1] RP-223549 New WID on Expanded and Improved NR Positioning

[2] TS 36.331

[3] TS 37.355

[4] TS 38.331