3GPP TSG-RAN WG2 #113b-e R2-210xxxx

Electronic Meeting, 12th Apr – 20th Apr 2021

Agenda Item: 8.10.2.3

Source: Samsung

Title: Report of [POST113-e][107][NTN] RLC and PDCP Aspects (Samsung)

Document for: Discussion, Decision

# 1 Introduction

The purpose of this document is to collect companies’ views for the following email discussion.

* [POST113-e][107][NTN] RLC and PDCP aspects (Samsung)

Scope: Based on RAN2#113-e contributions, discuss RLC and PDCP aspects

Intended outcome: email discussion summary

Deadline: Long

This email discussion is divided into two phases.

(i) Phase I. Companies’ preliminary views are collected. The deadline to contribute to Phase I is March 23, 11:00 UTC.

(ii) Phase II. The proposals are finalized to facilitate discussions and decision-making during live sessions of the RAN2#113b-e meeting. The deadline to contribute to Phase II is March 26, 11:00 UTC.

# 2 Discussion

## 2.1 Previous RAN2 Agreements

The following RLC and PDCP objectives are mentioned for RAN2 in the latest NTN WID [RP-202908].

* RLC
  + Status reporting: Extension of the value range of *t-Reassembly*
  + Sequence Numbers: extension of the SN space only for GEO scenarios
* PDCP
  + SDU discard: Extension of the value range of *discardTimer*.
  + Sequence Numbers: extension of the SN space for GEO scenarios.

In the RAN2#112-e meeting held in November 2020, following agreements were made [R2-00001].

Agreements:

1. RLC t-Reassembly timer needs to be extended in NR-NTN.
2. There is no need to extend t-PollRetransmit Timer in NR-NTN.
3. There is no need to extend t-statusProhibit Timer in NR-NTN.
4. There is no need to extend RLC SN length in NR-NTN.
5. There is no need to extend PDCP SN length in NR-NTN.

## 2.2 RLC Enhancements for an NTN: Discussion and Proposals

RAN2 has agreed to update the RLC t-Reassembly timer for an NTN. Long propagation delays necessitate an adjustment in the t-Reassembly timer.

The t-Reassembly timer is used to detect loss of PDUs at the receiving RLC entity [3]. The t-Reassembly timer is started at the receiving RLC entity if at least one byte is missing from the PDU segment received from the lower layer and if such timer is not already running. When this timer expires, the receiving entity declares a loss of the PDU. Such detection of RLC PDUs is used in RLC Unacknowledged Mode (UM) operation and RLC Acknowledged Mode (AM) operation [3]. In Release 16, the t-Reassembly timer can be configured to be a fixed value ranging from 0 ms to 200 ms.

As explained in [8], the data delivery without HARQ feedback can benefit from relatively short t-Reassembly values to reduce overall RLC retransmission delays while possibly allowing for a blind HARQ retransmission per gNB implementation.

For the data delivery with HARQ feedback, a longer t-Reassembly timer facilitates HARQ retransmissions while avoiding too early RLC status reporting. However, a very long t-Reassembly timer increases the overall service or user experience delay when some packets are lost on the radio interface. If TN-like short t-Reassembly timer values are used in certain NTN scenarios (e.g., GEOs and high-altitude LEOs with a large number of HARQ retransmissions), the RLC PDUs would be retransmitted while HARQ processes have not yet completed their procedures, leading to waste of precious radio resources. The gNB needs to have adequate flexibility in an NTN to achieve a target balance between the overall delay and unnecessary RLC retransmissions [8].

In an NTN, the delay between the UE and the gNB changes from one instant to another in case of quasi-Earth-fixed beams and Earth-moving beams due to the movement of the NTN platform (e.g., a LEO satellite) and due to the UE movement.

The t-Reassembly timer adjustment can make use of at least one of the two general options.

**Option 1.** The UE utilizes the t-Reassembly timer value that keeps changing based on the time-varying UE-gNB delay.

**Option 2.** The UE utilizes the t-Reassembly timer value that does not depend on the time-varying UE-gNB delay.

Option 1 is based on the approach that RAN2 has considered for other adjustments such as RA contention Resolution timer and start of the RA Response Window. Following drawbacks of Option 1 were mentioned in [5].

1. Option 1 would require the UE to keep re-calculating the RLC t-Reassembly timer value, because the UE-gNB delay can keep changing. This would increase the processing burden on the UE and adversely affect the UE’s battery life.

2. It is not feasible for the gNB to know the correct RLC t-Reassembly timer value used by the UE at a given instant because of the inherent long propagation delay between the UE and the gNB and the practical constraint of the UE not being able to continuously update the gNB about the current Timing Advance (TA) or the current UE-gNB delay. Hence, the UE and the gNB are unlikely to have the same value of the RLC t-Reassembly timer, potentially complicating the gNB’s data transfer operations for the UE.

Option 2 avoids these drawbacks of Option 1.

**Question 1. Which option do you prefer- Option 1 or Option 2? If you have any specific comments, please provide those in the “Comments” column.**

**Table 1: Preferences for an Approach for t-Reassembly Timer Enhancement**

|  |  |  |
| --- | --- | --- |
| Company | Preferred Option | Comments |
| Ericsson | 2 | Option 1 is acceptable from a NW point of view.  It is not necessary for gNB to know the exact value for the t-Reassembly as the gNB anyway do not know that a HARQ transmission fails the decoding in the UE.  For both options, the gNB can control the trade-off between time until an RLC status report is triggered and the amount of unnecessary RLC retransmission. |
| MediaTek | Prefer Option 2. | We prefer Option 2. However, Option 1 can also be acceptable |
| OPPO | Option 2 | We think option 2 is simple. If t-Reassembly needs to be changed, gNB can reconfigure it at a proper time. |
| Lenovo | Option 2 | Option 2 is simple and avoids too-frequent calculation. No need to keep the same value for the UE and gNB in any time. |
| CMCC | Option 2 | Option 2 is simple and easy to implement. Option 1 needs to consider energy consumption due to rapidly delay changing. |
| ZTE | Option 2 |  |
| Huawei, HiSilicon | Option 2 | We don’t see strong need to keep tracking the RTT and align with timer length. |
| Sequans | Option 2 | We acknowledge the intent to try to mitigate long t-reassembly timer with an adjusted dynamic value (Option1).  However given the drawbacks (continuous update, possible desync with gNB, test complications) we prefer Option 2. |
| Thales | Option 2 | We prefer option 2 which is simpler |
| Qualcomm | Option 2 |  |
| Nokia | Option 2 | Both Option1 and Option 2 are workable. Option2 is prefered for simplicity. |
| CATT | Option 2 | Option 1 would increase handling complexit of t-Reassembly Timer. The option 2 is more simple. |
| Intel | Option 2 | We share the view that this option is simpler. |
| Apple | Option 2 | At this point there is no need to complicate the solution with a variable t-reassembly timer. |
| Magister | Option 2 | Option 2 is simpler. |

Following 4 proposals were made in RAN#113-e contributions to modify RLC t-Reassembly timer for an NTN.

**Proposal A1** [5]. A generic formula is used to update multiple timers including RLC t-Reassembly timer (and PDCP discardTimer and the PDCP t-reordering if and when needed).

NTN t-ReassemblyTimer= (minimum\_NTN\_delay + R16 t-ReassemblyTimer value)\*scaling factor.

where “minimum NTN delay” is the minimum expected UE-gNB round-trip-delay and “scaling factor” is used to fine tune the overall delay relative to “minimum\_NTN\_delay.”

**Proposal A2** [6]. The extension of RLC t-Reassembly timer is left to network implementation. The maximum value (or value range) of the extended timer is FFS.

**Proposal A3** [7]. Add the following set to the R16-specified set of values for t-Reassembly timer: {ms210, ms220, ms340, ms350, ms550, ms1100, ms1650, ms2200}.

**Proposal A4** [8]. The RLC t-Reassembly for NTNs is extended as (t-Reassembly + k\_reassembly \* RTT), where t-Reassembly is the legacy RLC parameter and k\_reassembly and RTT are new RRC parameters.

**A Closer Look at the “t-ReassemblyTimer” Proposals**

**Proposal A1**. According to [5], this generic formula/framework is reusable for various timers. Furthermore, this framework enables reuse of existing R16 timers and provides a better time resolution for a given NTN type compared to the case when timer values are extended by adding new numerical values. Additionally, the framework is more efficient from signaling and processing perspectives. For example, there is no need to keep recalculating and updating t-ReassemblyTimer due to the ever-changing propagation delay for quasi-Earth-fixed beams and Earth-moving beams. This framework also enables both the gNB and the UE to know the exact timer value. A side benefit of conveying minimum\_NTN\_Delay to UEs in the cell is that (i) a UE with pre-compensation capability but experiencing poor GNSS visibility and (ii) a UE without GNSS capability can make use of this delay to adjust uplink transmission time during the random access procedure. The main drawback of this proposal is that the gNB needs to broadcast one or two parameters (i.e., minimum\_NTN\_delay and scaling factor).

**Proposal A2 [6]**. The benefit of this proposal is that the gNB may not need to broadcast any new parameters, because the set of the R16 values itself would be enlarged. The drawback of this proposal is that a desired resolution (i.e., difference between successive allowed values) may not be achievable when widely different propagation delays from HAPS to LEOs with varying altitudes to MEOs with varying altitudes to GEOs are considered. Additionally, the range extension would need to be defined separately for different parameters (e.g., RLC t-Reassembly timer, PDCP discardTimer, and the PDCP t-reordering timer).

**Proposal A3 [7].** This proposal assumes the number of HARQ retransmissions to be {1, 2, 3, 4, or 8} for LEOs and {1, 2, 3, 4} for GEOs. The benefit of this proposal is simplicity; the gNB does not need to broadcast any new parameters, because the set of the R16 values itself is enlarged. The drawback of this proposal is restricted resolution or granularity of values for different NTN platforms (e.g., HAPS, LEOs with different altitudes, MEOs with different altitudes, and GEOs). Additionally, the range extension would need to be defined separately for different parameters (e.g., RLC t-Reassembly timer, PDCP discardTimer, and the PDCP t-reordering timer).

**Proposal A4** (Reproduced from [8]). This proposal introduces a scale factor k\_reassembly in the RLC configuration that models the number of re-transmissions that the gNB would like to attempt as well as an RTT parameter according to possible satellite orbit, which can be a UE variable that is configured at RRC-level to be used by the RLC entity, similar to UE variables in MAC or RRC UE. This RTT can be reused for all timers that are needed as a function of the RTT in the UE, one example being drx-HARQ-RTT-TimerDL when UL and DL are not aligned in the gNB.

**Question 2. Which of the proposals do you prefer to adjust the RLC t-Reassembly timer? If you have a new proposal that you like RAN2 to consider, please briefly explain it in your response to Question 3.**

**Table 2: Preferences for Existing Candidate Proposals for the** **RLC t-Reassembly Timer**

|  |  |  |
| --- | --- | --- |
| Company | Preferred Proposal | Comments |
| Ericsson | A4 | A1: there is no need for gNB to know exact value on t-Reassembly. There is no way that minimum\_NTN\_Delay can be accurate enough during RACH if the UE do not have GNSS coverage without a major redesign in PHY layer. Apart from that, this is similar to A4.  A2: It is always up to the NW which parameter value to signal to the UE, but it is not possible to leave for NW implementation the values in the spec. The values in the spec for this method will have to be defined.  A3: This is a feasible method but will require adding much more values to cover all possible satellite orbit distances. Further using method A1 or A4, the added values can be reused for may other timers instead of each timer being extended with multiple values that fit all orbits from 600 km to 35786 km.  A4: Simple and enables reusing the new RTT RRC parameter value for other timers such as drx-HARQ-RTT-TimerDL and sr-ProhibitTimer. |
| MediaTek | A2 or A3 | Option A2 and Option A3 are almost equivalent as both suggest extending the value range of the timer. |
| OPPO | A2 or A3 | We prefer that gNB configures t-Reassembly for UE since HARQ retransmission is up to gNB to decide. With the extended value range, how to configure the length of t-Reassembly can be up to gNB. |
| Lenovo | A2 or A3 | No new parameter is broadcasted. |
| CMCC | A2 or A3 | Opt.2 and Opt.3 are similar and simple to standardize. |
| ZTE | A2 or A3 | A2 and A3 is the same approach which has least specs impact. The candidate values can be discussed further. |
| Huawei, HiSilicon | A2 or A3 | The exact length value for the timer can be FFS. |
| Sequans | A2 or A3 | Just extend the IE with new values might be good enough, and avoids to change the existing signaling framework.  Candidates values can be discussed, depending of the finetuning required (the smaller the granularity, the lower the unnecessary delay in reporting missing PDUs to gNB). |
| Thales | A2 or A3 | A2 and A3 have least specifications impact. FFS for exact timer values. |
| Qualcomm | A2 (configuration of value is up to network) | We can decide the exact values later. |
| Nokia | A2 or A3 | We prefer the simple solution to extend value range. |
| CATT | A2 or A3 | A2 and A3 are similar approach and both have small specs impact. |
| Intel | A2 (e.g. considering A3) |  |
| Apple | A3 | A3 has the least amount of spec impact. |
| Magister | (A2 or) A3 | The possible t-Reassembly timer values need to be specified and are FFS. |

It has been mentioned in [9] that multiple RLC STATUS reports are sent at the expiry of t-Reassembly timer when multiple RLC PDUs in a data burst are missing. To address this issue, the start of the RLC t-Reassembly timer can be delayed so that STATUS report can be consolidated for multiple RLC PDUS in a data burst [9].

**Question 3. Do you think RAN2 should discuss the issue of the RLC status reports?**

**Table 3: Preferences for the** **RLC Status Reports**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Comments |
| Ericsson | No | Adding a T-Reassembly-delay-timer to avoid additional and/or delay status reports is an unnecessary optimization.  It is not a serious problem that the SR for a missing second PDU is delayed up to t-StatusProhibit when a first PDU is already outstanding.  The SR for the first missing PDU may be delayed and then indicate also the second missing PDU. The next SR will anyway report both PDUs if they are still outstanding. |
| MediaTek | No | This does not seem like an NTN-specific problem and should not be discussed here. |
| OPPO |  | We are ok to discuss the issue, but the spec impact should be minimized when discussing solutions. |
| Lenovo | No | The issue mentioned is not NTN-specific. |
| CMCC |  | To discuss this issue is needed. |
| ZTE | No | Share the same view as Ercisson and MediaTek. |
| Huawei, HiSilicon | No | If the intention is to avoid multiple status report, we already have *t-StatusProhibit.* |
| Sequans | Yes | The main point is to **avoid unnecessary delay** for reporting missing PDU(s) to the gNB (in the common case where 1 PDU is eventually not recovered by HARQ, while at least a previous one was pending recovery).  In addition, in (comparatively rare) cases where 2 PDUs in the same burst are not recovered by HARQ, it also enables to send only one RLC SR, with as little delay as possible (contary to use of t-statusProhibit).  @Ericsson: it is an optimization indeed, which has the same ultimate goal as any finetuning of the t-reassembly timer discussed in Question 2 (not sure why delaying RLC SR would turn out not to be a problem in Question 3, whereas in Question 2 you support to change the existing signaling framework to finetune the t-reassembly value).  @Mediatek, Lenovo: what is NTN specific is that in NTN reporting may be unnecessarily delayed by up to around 2 seconds.  @Huawei: the main intention is not to avoid multiple RLC SRs but to reduce the unncessary delay to report missing PDUs to the gNB. Obviously t-statusProhibit on the contrary increases reporting delay. |
|  |  |  |
| Thales | No | We agree with Mediatek |
| Qualcomm | No | There is mechanism to avoid frequent RLC status reporting. Delay to send RLC status may not be huge problem in NTN considering already large delay is expected if retransmission is enabled. |
| Nokia |  | Considering the long RTT in NTN, we think the SR delay for second missing PDU can be further analysed to see if it is a big issue and if any optimisation needed. |
| CATT | No | It is not a NTN-specific issue and should not be discussed here. |
| Intel | No | We share the same view explained by Ericsson. |
| Apple | No | This is not a NTN specific issue. |
| Magister | No | Out-of-scope. |

## PDCP Enhancements for an NTN: Discussion and Proposals

Two PDCP timers, discardTimer and PDCP t-Reordering, may need to be revisited by RAN2 if QoS is adjusted for an NTN by SA2 to reflect long propagation delays.

The transmitting PDCP entity starts the discardTimer corresponding to a PDCP SDU upon receiving such SDU from an upper layer (e.g., IP) [4]. If the PDCP discardTimer associated with a PDCP SDU expires or if the successful delivery of a PDCP SDU is indicated by the PDCP status report from the receiving PDCP entity, the transmitting PDCP entity discards the PDCP SDU. R16 allows PDCP discardTimer to be from 0.5 ms to 1500 ms or infinity. The PDUs waiting for retransmission are discarded due to the expiry of the PDCP discardTimer if the PDCP discardTimer is smaller than RLC t-Reassembly timer [6].

The PDCP t-Reordering timer at the receiving PDCP entity is started or reset when a PDCP SDU is delivered to an upper layer (e.g., IP) [4]. The PDCP t-Reordering timer helps detect loss of PDCP PDUs. The maximum configurable value for the PDCP t-Reordering timer is 3 s. It has been observed in [6] that the PDCP t-Reordering timer need to be at least longer than the RLC t-Reassembly timer to allow RLC procedures to complete.

**Question 4. Do you agree that** **PDCP discardTimer and PDCP t-Reordering timer need to be at least longer than the RLC t-Reassembly timer?**

**Table 4: Requirements for the PDCP discardTimer and t-Reordering timer**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Comments |
| Ericsson | Yes |  |
| MediaTek | Yes |  |
| OPPO | No | We are not sure why we need to discuss this issue, as these timers are all configured by the network. Network can handle.  For example, the start time between PDCP t-Reordering and RLC t-Reassembly can be different, hence the value ranges of them seem to have no causal relation. |
| Lenovo | Yes | NW can ensure it via configuration. |
| CMCC | Yes |  |
| ZTE | No need for this proposal | We understand the intention, but it seems we don’t need this proposal since the configuration of timers is under NW’s control and this proposal has no impact on specs or UE behavior. |
| Huawei, HiSilicon | Partially Yes | PDCP t-Reordering timer need to be at least longer than the RLC t-Reassembly timer, as they are both in receiving side.  PDCP discardTimer in UE side is configured by network, and we don’t have specified RLC t-Reassembly timer in network side. So there is no restriction on PDCP discard timer, because when to send status report is up to network implementation. |
| Sequans | Partially Yes | Same view as Huawei. |
| Thales | Yes |  |
| Qualcomm | Yes | We assume the intention of question is whether PDCP discard timer and re-ordering timer needs to be extended to the same range as t-reassembly timer. |
| Nokia | Partially Yes | Same view as Huawei. |
| CATT | Yes |  |
| Intel | Yes |  |
| Apple | Yes |  |
| Magister | Yes |  |
|  |  |  |

**Question 5. If the answer to Question 4 is yes, do you agree that such constraint can be addressed via network implementation without any changes to the specifications?**

**Table 5: Addressing the Constraint for PDCP discardTimer and t-Reordering timer**

|  |  |  |  |
| --- | --- | --- | --- |
| Company | Yes/No | Comments | |
| Ericsson | Yes |  | |
| MediaTek | Yes for Network implementation, but the timer-range needs to be extended in RRC | Given that we are extending the RLC t-Reassembly timer, the range of values for PDCP discard and reordering timers need to be extended equivalently. This will allow network implementation to meet the constraints. | |
| Lenovo | Yes |  | |
| CMCC | Yes | |  |
| ZTE |  | As commented, it is up to NW’s implementation. | |
| Huawei, HiSilicon | yes | It can be addressed by network implementation. | |
| Sequans | Yes |  | |
| Thales | Yes | This is up to NW implementation | |
| Qualcomm | Yes | Proper configuration is expected from the network. | |
| Nokia | Yes | It’s network implementation aspect. | |
| CATT | Yes |  | |
| Intel | Not essential | This behaviour it is a network guidance but final decision should be left up to network implementation. | |
| Apple | Yes |  | |
| Magister | Yes | Up to network implementation | |

Based on previous RAN2 discussions, it was found that the maximum values of 1500 ms for PDCP discardTimer and 3 s for t-Reordering timer would be adequate for an NTN. If SA2 updates QoS for an NTN, RAN2 may need to re-visit the extensions of PDCP discardTimer and t-Reordering timer. *The question below aims to collect initial views from contributing companies in an attempt to accelerate future progress on this topic.*

In case PDCP discardTimer and t-Reordering timer need to be extended based on SA2’s work on NTN QoS, the following proposal was made in [3].

**Proposal B1**. To update the PDCP discardTimer and the PDCP t-reordering timer per SA2 requirements, consider the generic framework of “NTN Timer Value= (minimum\_NTN\_delay + R16 timer value)\*scaling factor,” where “minimum NTN delay” is the minimum expected UE-gNB round-trip-delay and “scaling factor” is used to fine tune the overall delay relative to “minimum\_NTN\_delay.”

**Proposal B2**. Enlarge the set of allowed values separately for the PDCP discardTimer and the PDCP t-reordering timer. The exact sets of values and the target scenarios (e.g., combinations of the number of HARQ retransmissions and the altitudes of LEOs and MEOs) are FFS.

The benefit of Proposal 1 is that a general framework is used for multiple timers but the drawback is the need for the gNB to broadcast one or two parameters.

The benefit of Proposal is that the gNB does not need to broadcast any additional parameter in System Information. The drawback of this proposal is that the granularity or time resolution may reduce the flexibility if the set of new values is small or a large set of new values would be needed to ensure a fine resolution similar to that in a Terrestrial Network.

**Question 6. If SA2 decides to update QoS to allow longer delays for an NTN, do you prefer Proposal B1 or Proposal B2? If you like to make a new candidate proposal, please briefly explain it in the “Comments” column. If you like to postpone any decision-making for this topic at this time, feel free to mention “Postpone” in the “Preferred Proposal” column.**

**Table 6: Preferences to Enhance the PDCP discardTimer and t-Reordering timer**

|  |  |  |
| --- | --- | --- |
| Company | Preferred Proposal | Comments |
| Ericsson | B2 | Only if SA2 adds new 5QIs, the 5QIs will not vary with constellation orbit height so it is fine to extend the value range with higher values.  Currently it is possible to configure RLC and PDCP in a way that RLC t-Reassembly <= PDCP t-Reordering <= PDCH discardTimer |
| MediaTek | Proposal B2 | As mentioned in our previous response, the timer-range can be extended to ensure that it is more than the extended RLC t-Reassembly timer. |
| OPPO | Postpone | We can postpone the discussion for this topic until SA2 has a decision on new QoS for NTN. |
| Lenovo | Postpone | SA2 decision is the precondition for this discussion. |
| CMCC | B2 with comments | Whether SA2 will introduce new requirements should go first. |
| ZTE | Postpone | The discussion can be postpone until SA2 has conclusions. |
| Huawei, HiSilicon | Postpone | We can further discuss this issue after we get reply LS from SA2. |
| Sequans | Postpone | We should wait for SA2 feedback |
| Thales | Proposal B2 | SA2 added a new non-GBR 5QI with PDB=832ms as per CR S2-2101669. |
| Qualcomm | B2 | This timer needs to be extended. |
| Nokia | Proposal B2 | RAN2 needs to discuss the LS from SA2 first to decide on the setting of the extension value. |
| CATT | Postpone | This issue should be postponed until SA2 has decision. |
| Intel | B2 |  |
| Apple | Postpone | And wait for SA2 feeback. |
| Magister | Proposal B2 |  |
|  |  |  |

# 3 Conclusion

Based on the discussion in Section 2, the following candidate proposals are suggested. [To be completed during Phase II]

# 4 References

1. RP-202908, NTN WID.
2. R2-200001. RAN2#112-e Meeting Report.
3. TS38.322, RLC.
4. 3GPP, TS 38.323, PDCP.
5. Samsung, R2-2100253.
6. MediaTek, R2-2100357.
7. Thales, R2-2101259.
8. Ericsson, R2-2101492.
9. Sequans Communications, R2-2101518

# Annex

To facilitate possible offline discussions, all delegates who have provided input for this document are requested to fill the following table.

|  |  |  |
| --- | --- | --- |
| Company | Name | Email Address |
| Samsung | Nishith Tripathi | nishith.t@samsung.com |
| Ericsson | Robert S Karlsson | robert.s.karlsson AT ericsson.com |
| OPPO | Haitao Li | lihaitao@oppo.com |
| Lenovo | Xu Min | xumin13@lenovo.com |
| ZTE | Zhihong Qiu | qiu.zhihong@zte.com.cn |
| Sequans | Olivier Marco | omarco at sequans.com |
| Thales | Camille Bui | Camille.bui@thalesaleniaspace.com |
| Qualcomm | Bharat Shrestha | bshrestha@qti.qualcomm.com |
| Nokia | Ping Yuan | Ping.1.Yuan@nokia-sbell.com |
| CATT | Jianxiang Li | lijianxiang@datangmobile.cn |
| Intel | Marta Martinez Tarradell | marta.m.tarradell@intel.com |
| Magister | Jani Puttonen | jani.puttonen@magister.fi |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |