**3GPP TSG-RAN2 Meeting #129bis** **R2-25xx**

**Wuhan, China, Apr 7-11, 2025**

**Agenda Item:** 8.2.2

**Work Item:** Ambient\_IoT\_Solutions

**Source:** Qualcomm Incorporated

**Title:** Email discussion report: [POST129][035][AIoT] Paging

**Document for:**Discussion/Decision

# Background

RAN#106 approved WI for Ambient IoT in [1]. One of the objectives for RAN2 is listed as follows:

|  |
| --- |
| * + Specify the necessary functions and procedures for an Ambient IoT compact protocol stack and lightweight signalling procedure to enable DO-DTT and DT data transmission:     - A-IoT Paging, including subsequent paging for the same service. Support the options that a paging message contains one identifier, and that a paging message contains no identifier. Temporary identifier is not supported, unless required by SA WGs.   Note: RAN2 aims to design a paging message format such that multiple identifiers can be contained in one paging message, for forward compatibility purposes. |

RAN2#129 discussed on paging aspects and made some agreements, and to discuss further, RAN2#129 allocated following email discussion:

* [POST129][035][AIoT] Paging (Qualcomm)

Intended outcome: Discuss and address the remaining paging FFSs considering the agreements this meeting: FFS device behaviour if multiple requests are received in parallel (if needed), FFS how reader will generate “transaction ID”, FFS the size of transaction ID.

Deadline: Long

Below is the list of RAN2#129 agreements with yellow highlights added to the FFSes:

**Agreements**

1. Parallel service requests by the same reader is not supported.
2. The device is expected to only perform one procedure at a time. FFS device behaviour if multiple requests are received in parallel (if needed).
3. The “transaction ID” can be generated by reader based on CN corelation ID. FFS how reader will generate “transaction ID”. FFS the size of transaction ID
4. 1 bit solution is excluded. FFS the size. Aim to have a reasonable size.
5. RAN2 acknowledges that multi-reader scenario may exist but we will not specify something specific for this purpose. We can rely on transaction ID and implementation to handle it.

**Agreements on paging ID**

1. The “one identifier” in the paging message includes both the case of “one single device identifier” and “one group identifier”/”filtering criteria”, while the exact format of latter is supposed to be designed by SA2.
2. The current assumption is that the paging identifier is transparent to the A-IoT MAC Layer and carried by upper layer. FFS if there is really a need for visibility in the MAC layer

|  |
| --- |
| **Agreements**   1. The A-IoT paging message can include a number of msg1 resources 2. From RAN2 perspective, after initial paging message, the R2D transmission which determines the Msg1 resource(s), can be achieved by one of the below two ways, unless RAN1 concludes to use L1 signaling later:   **Way-1**: introducing new R2D message other than the paging message, e.g., QueryRep-like; or  **Way-2***:* reusing the same paging message, using field(s) to indicate it is only to determine the Msg1 resource(s) and omitting the paging identifier (device ID/group ID) field  3. The service type of A-IoT (e.g., inventory only, inventory + command) is not included in paging message. |

This document is the report of the email discussion [POST129][035][AIoT] Paging.

# Contact information

|  |  |  |
| --- | --- | --- |
| **Company** | **Contact Name** | **Email** |
| Lenovo | Jing HAN/Hyung-Nam Choi | hanjing8@lenovo.com  hchoi5@lenovo.com |
| vivo | LI Yuan | yuanli@vivo.com |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Discussion on device behaviour if multiple requests are received in parallel

## Multiple paging for Different Service Requests

Note that following is already agreed which anticipates a device **should not expect to receive** another service request **from the same reader** while there is one ongoing procedure.

1. Parallel service requests by the same reader is not supported.
2. The device is expected to only perform one procedure at a time.

However, *not expect to receive* does not guarantee it will not happen. What would be the device behaviour if another service request **is received from the same reader** while there is one ongoing procedure? Thus, RAN2 has captured the FFS on device behaviour if multiple requests are received in parallel (if needed).

In addition, the above agreement #1 only covers the same reader case, i.e. different reader case is still open. However, an agreement relevant to this scenario is the following:

5. RAN2 acknowledges that multi-reader scenario may exist but we will not specify something specific for this purpose. We can rely on transaction ID and implementation to handle it.

**Q1: Should it be possible for the device to distinguish between whether the another (different) service request is received from the same reader vs the another (different) service request is received from a different reader?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comment** |
| Lenovo | No | Since A-IoT device is expected to perform only one procedure at the same time, no matter whether a different service request is received from the same reader or a different reader, the device cannot respond to it due to the capability limitation. Thus there has no need to distinguish between above two cases. |
| vivo | No | On the one hand, we see little possibility that the device is performing an ongoing procedure while encountering another service request, i.e., considering the low capability of device type 1, the same reader does not trigger parallel services toward one device at one time, different readers are expected to coordinate by implementation to avoid parallel service requests at the same time. In turn, there is no need for a device to identify readers.  On the other hand, similar view as above, it is unnecessary for the device to distinguish the paging message from the same or a different reader. As agreed that the device is expected to only perform one procedure at a time, it should not be involved in another (different) service request no matter if it is from the same or a different reader. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Summary:** TBD

**Q2: If answer to Q1 is yes, would the device behavior be different between: another (different) service request is received from the same reader vs another (different) service request is received from a different reader while there is one ongoing procedure?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comment** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Summary:** TBD

**Q3: What would be the device behavior if another (different) service request is received from the same reader while there is one ongoing procedure?**

|  |  |
| --- | --- |
| **Company** | **Comment** |
| Lenovo | Since it was agreed in last meeting parallel service requests by the same reader is not supported, we are wondering the motivation for this scenario. If this happens, we assumed the device will ignore the different service request received from the same reader if there is ongoing procedure. |
| vivo | We do not think this case will happen, since the agreement does not support “parallel (another, different) service requests by the same reader”. That is, the reader itself should ensure that it would not trigger paging related to another service request towards a device, which has not successfully finished an ongoing procedure. |
|  |  |
|  |  |
|  |  |
|  |  |

**Summary:** TBD

**Q4: What would be the device behavior if another (different) service request is received from a different reader while there is one ongoing procedure?**

|  |  |
| --- | --- |
| **Company** | **Comment** |
| Lenovo | Considering there has no specific priority or latency requirements for A-IoT related services. If there is ongoing procedure, we assumed that the device will ignore the different service request if received from a different reader. |
| vivo | As mentioned in Q1, we see little possibility of such case.  However, once it happens, from the device perspective, itself cannot tell it is from a different reader but only it is related to another service request. In this sense, if the ongoing procedure is not finished, the device should ignore this another (different) service request. |
|  |  |
|  |  |
|  |  |
|  |  |

**Summary:** TBD

The above questions deal with ‘different’ service requests while a procedure is already ongoing at the device. However, the same service request may be transmitted again in R2D direction (and this could be from the same or a different reader).

## Multiple paging for Same Service Request

Some companies have assumed that ‘transaction ID’ would be used by the device to detect repeated paging for the same service request, while others have indicated transaction ID may not be sufficient, especially if there is no coordination between the readers and the transaction ID is not large enough. Different views on reader ID, service ID etc. has been brough up before.

**Q5: Should it be possible for the device to distinguish between whether the same service request is received from the same reader vs the same service request is received from a different reader?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comment** |
| Lenovo | Yes | For resource and energy efficient consideration, the device needs to avoid the redundant responses from the same service request in principle. However, as discussed in RAN3 for locating purpose, CN may send a command to one or more readers associated with the A-IoT device to locate the device, in this case, the service request may be the same and whether the device needs to response it needs further discussion.  Therefore, the device is suggested to distinguish whether the same service request is received from the same reader or different reader. |
| vivo | No | RAN3 has agreed the case that CN allocates the same service request to different readers, possibly in order to improve the inventory efficiency. And the rational device behavior is only to successfully respond once when receiving the same service request from different readers for most scenarios. Moreover, one reader sending multiple paging messages for the same service request is to increase the successful inventory rate, where the device also only needs to successfully respond once.  Regarding the **multi-reader scenario** for the proximity service, the device is expected to respond to multiple readers for a same service. But as the agreement enlightened, the CN can assign readers with different correlation IDs and therefore reader can generate different transaction IDs for a single service. That is, we rely on CN but not the device to identify readers if the device is required to respond to different readers; and from the device point of view, the issue fallback to the device behavior of another (different) service request while there is one ongoing procedure as illustrated in Q4.  In this sense, we see no necessity for the device to distinguish whether the same service is from the same or a different reader. The device only needs to judge whether itself has successfully responded (i.e., successfully access to a reader) to such service request. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Summary:** TBD

Now let’s go to the device’s expected behaviour for the **same** service request received again.

**Q6: If answer to Q5 is yes, would the device behavior be different between: the same service request is received from the same reader vs the same service request is received from a different reader after the device has previously responded to the same service request?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comment** |
| Lenovo | Yes | As commented in Q5, if location scenario is considered, then the device needs to distinguish whether the same service request is received from the same reader or different reader. Corresponding device behavior for responding can be different. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Summary:** TBD

**Q7: What would be the device behavior if same service request is received from the same reader after device has previously responded to the same service request?**

|  |  |
| --- | --- |
| **Company** | **Comment** |
| Lenovo | As discussed in SI phase, if the same service request is received from the same reader, the device skips to respond to the same service request to avoid duplicate responses. |
| vivo | It needs clarification on “device has previously responded to the same service request”, since the device behavior can be different depending on whether the previous response is successful or not.  If the device considers the previously-responded service request is successfully responded before receiving the same service request, it should ignore this same service request. On the contrary, if the previous service is not finished yet and the device receives this same service request, the device should response to such same service request. |
|  |  |
|  |  |
|  |  |
|  |  |

**Summary:** TBD

**Q8: What would be the device behavior if same service request is received from a different reader after device has previously responded to the same service request?**

|  |  |
| --- | --- |
| **Company** | **Comment** |
| Lenovo | For locating scenario, CN may indicate one or more readers associated with the A-IoT device to find the A-IoT device by transmitting the A-IoT paging message. From the A-IoT device perspective, it may receive multiple A-IoT paging messages from different readers (e.g., BS, intermediate node etc.) associated with the same service request. In this case, our view is that device needs to respond to the service request at least for location purpose.  Regarding to how the device to distinguish the same service request is from a same or different reader, reader ID is introduced or based on transaction ID can be further discussed. If transaction ID is used, the coordination among readers may be needed, or rely on CN implementation to generate different correlation ID for the same locating service request. |
| vivo | As mentioned in Q1, we see little possibility of deploying such case.  Once it happens, with no differentiation on reader, the device behavior is illustrated in Q7. |
|  |  |
|  |  |
|  |  |
|  |  |

**Summary:** TBD

# Discussion on transaction ID

Note that RAN2 has agreed the following:

5. RAN2 acknowledges that multi-reader scenario may exist but we will not specify something specific for this purpose. We can rely on transaction ID and implementation to handle it.

Following question is to confirm the companies understanding on use of the transaction ID.

**Q9: Is transaction ID sufficient for the device to confirm that the received service request is the same service request from the same/different reader that the device has already received/responded to?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comment** |
| Lenovo | Yes | Device can determine whether to respond the received paging message with the assistance of appropriate transaction ID generation mechanism and CN implementation. One of the possible mechanisms is as follows:   * Different readers generate same transaction ID based on the same correlation ID from CN. In this way, device doesn’t send duplicated response to different readers for the service request with same correlation ID. * For the proximity/locating service, CN sends service requests with different correlation ID to different readers. So readers generate different transaction ID, and device responds to each reader. |
| vivo | Yes | Based on the agreement “the “transaction ID” can be generated by reader based on CN corelation ID”, the transaction ID is specific to a single service, where the CN correlation ID is used to identify a single service request between network interfaces.  In this sense, the device is able to identify the currently-received paging message is related to the same or different service request from the one it has already received/responded to, by comparing whether the transaction ID is same or different. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Summary:** TBD

**Q10: If your answer to Q9 is no, what else is needed for the device to confirm that the received service request is the same service request from the same/different reader that the device has already received/responded to?**

|  |  |
| --- | --- |
| **Company** | **Comment** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Summary:** TBD

## Generation of Transaction ID

RAN2 has agreed that the transaction ID can be generated by reader based on CN correlation ID. However exact details have not been discussed. Furthermore, whether there is a standard-specified coordination mechanism between the readers is more relevant for RAN3. Following questions are to understand from RAN2 point of view, how do companies envision the transaction ID being generated by the readers.

**Q11: Is there a need for coordination between the readers when generating transaction ID?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comment** |
| Lenovo | Depends | It depends on whether the transaction ID generation mechanism is specified or not.   * Opt 1: Explicit signaling between readers may not be needed if generation mechanism is specified, i.e., readers can achieve alignment based on the specified generation mechanism, e.g., use the LSB X bits of correlation ID as the transaction ID. * Opt 2: If it’s up to reader implementation to generate transaction ID or reader randomly select X bits among all bits of correlation ID to generate transaction ID, different readers may generate same/different transaction ID for different/same service, causing misunderstanding on device. To avoid the situation, explicit signaling between readers is needed in this case.   We prefer Opt1 which does not introduce signaling overhead between readers and has less impact to RAN3. |
| vivo | See comments | To make transaction ID unique to a service is to ensure the generation method is unified for all readers per AIOTF. With the same method towards the same correlation ID, the reader itself is able to generate the same transaction ID for the same service without coordination with other readers.  From RAN2 perspective, we prefer to avoid reader coordination, and it can be more like a requirement to SA2/RAN3 to specify a unified generation method of correlation ID among readers. But the final decision is up to other WGs. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Summary:** TBD

**Q12: How is the transaction ID generated by the Reader based on CN correlation ID?**

|  |  |
| --- | --- |
| **Company** | **Comment** |
| Lenovo | As commented above, we prefer to specify a simple generation mechanism. Using the unique part of the correlation ID as the transaction ID is simple and efficient, e.g., use the LSB bits (last X bits). |
| vivo | We consider this issue is more related to SA2/RAN3’s design, which should take into account that the uniqueness of CN correlation ID within a certain period over a certain area. |
|  |  |
|  |  |
|  |  |
|  |  |

**Summary:** TBD

## Size of Transaction ID

RAN2 has captured the following regarding the size of Transaction ID:

4. 1 bit solution is excluded. FFS the size. Aim to have a reasonable size.

**Q13: What should be the size of the ‘transaction ID’? (Note that 1 bit is already excluded and RAN2 aim is to have a reasonable size. So, please clarify why/how more/less bits are needed/sufficient.)**

|  |  |
| --- | --- |
| **Company** | **Comment** |
| Lenovo | Unfortunately, we don’t know yet the structure and length of the correlation id.  As reference the length of the LCS Correlation ID is minimum 1 character and maximum 255 characters. And depending on the encoding scheme one or multiple bytes are needed to encode a character, e.g. 1 to 4 bytes may be needed to encode a character in case of UTF-8.  Therefore, it may be better to defer the discussion on the size of the transaction id and coordinate first with SA2 on the details of the correlation id (structure and length). |
| vivo | From AS perspective to avoid duplicated responses, we consider at least 3-bit is a reasonable size of the transaction ID.  To be specific, the size depends on the maximum number of services that the CN may trigger towards a device within a certain time period and the flexible adjustment room for likely wrap-around case. Recurring RFID, 4 parallel services are allowed for a device. If we totally reuse, 2-bit length can be too extreme which requires CN limitation. To be more competent than RFID, 3-bit transaction ID means that no more than 8 services can be triggered towards a device, which allows CN more flexibility to reduce time latency at a tolerable expense and avoids the possibility of wrap-around.  Most importantly, agree with Lenovo, we still believe the size of transaction ID requires SA2 evaluation. To be specific, RAN2 should have consensus on its usage and requirement in AS layer, keep SA2/RAN3 informed ANS ask them about Q11-13 on transaction ID design. |
|  |  |
|  |  |
|  |  |
|  |  |

**Summary:** TBD

# Discussion on visibility of paging identifier to the MAC layer

The relevant agreement from RAN2#129 is copied again:

1. The “one identifier” in the paging message includes both the case of “one single device identifier” and “one group identifier”/”filtering criteria”, while the exact format of latter is supposed to be designed by SA2.
2. The current assumption is that the paging identifier is transparent to the A-IoT MAC Layer and carried by upper layer. FFS if there is really a need for visibility in the MAC layer

Separately, the reply LS from SA3 in R2-2501502 indicates the following SA3 conclusions captured in S3-251048:

|  |
| --- |
| The following aspects and principles are agreed for the conclusion on KI#3  - A mechanism to protect AIoT device ID based on the use of temporary ID shall be supported.  - Mechanism shall allow unambiguous identification of the AIoT device  - A mechanism to re-synchronize de-synchronized temporary IDs shall be supported.  Editor’s Note: Additional conclusions on solution are FFS |

The above seems to imply that the temporary ID, instead of the AIoT device ID, is to be used as AIoT paging identifier. However, RAN2 has not discussed this in detail.

**Q14: Please provide your view one whether the paging identifier needs to be visible to the MAC layer. If yes, why?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes/No** | **Comment** |
| Lenovo | No | According to above information provided by the Rapp, we understand this temporary ID that used as paging ID, is generated by the core network e.g. AIoTF based on the device ID. It is not suitable to let AS layer to further process the id e.g. filter or re-group. To us it is unnecessary to let paging identifier visible to the MAC layer. |
| vivo | No, see comments | The primary thing is that the paging identifier is recognized and handled by A-IoT NAS layer or MAC layer. From our perspective, the paging identifier is allocated by AIOTF and maintained at A-IoT NAS layer. If the paging identifier is visible to AS layer, it violates the end-to-end design criterion. In this sense, we see no need to make it visible to MAC layer.  However, when temporary ID is supported as a form of paging identifier, it can be viable that such paging identifier to be visible to MAC layer from technical perspective. But we think the benefit of visibility needs to be proved further and depends on other WGs’ progress. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Summary:** TBD

# Misc/Other

**Q15: Please list below if there are other open issues which should be addressed in this email discussion.**

|  |  |  |
| --- | --- | --- |
| **Company** | **Issue/Question** | **Comment/Details** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Summary:** TBD

# Summary

TBD

# References

[1] RP-243326, New Work Item: Solutions for Ambient IoT (Internet of Things) in NR, RAN#106, Dec 2024