**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 |
| OPPO | Liu Yang | liuyangbj@oppo.com |
| Huawei, HiSilicon | Yulong | shiyulong5@huawei.com |
| CMCC | Ningyu Chen | chenningyu@chinamobile.com |
| CATT | Jianxiang Li | lijianxiang@catt.cn |
| Spreadtrum, UNISOC | Huifang Fan | Huifang.fan@unisoc.com |
| Apple | Zhibin Wu | Zhibin\_wu@apple.com |
| Xiaomi | Yi Guo | Guoyi16@xiaomi.com |
| Tejas Networks | Sushmita Ghosh | [sushmitag@tejasnetworks.com](mailto:sushmitag@tejasnetworks.com) |
| ZTE | Lu Ting | [lu.ting@zte.com.cn](mailto:lu.ting@zte.com.cn) |
| InterDigital | Martino Freda | martino.freda@interdigital.com |
| MediaTek | Nathan Tenny | [nathan.tenny@mediatek.com](mailto:nathan.tenny@mediatek.com) |
| Nokia | Jakob Buthler | Jakob.buthler@nokia.com |
| ETRI | Seungkwon Baek | [skback@etri.re.kr](mailto:skback@etri.re.kr) |
| Qualcomm | Umesh Phuyal | uphuyal@qti.qualcomm.com |
| NEC | Zonghui Xie/  Satoaki Hayashi | [xie\_zonghui@nec.cn](mailto:xie_zonghui@nec.cn)  satoaki-hayashi@nec.com |
| Samsung | Weiwei Wang  Sangyeob Jung | [ww1016.wang@samsung](mailto:ww1016.wang@samsung)  sy0123.jung@samsung.com |
| HONOR | Xiaoxuan Tang | tangxiaoxuan@honor.com |
| Sharp | Chongming Zhang | Chongming.zhang@cn.sharp-world.com |
| Futurewei | Yunsong Yang | yyang1@futurewei.com |
| Ericsson | Emre A Yavuz | emre.yavuz@ericsson.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:

1. 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. |
| OPPO | No | No. agree with Lenovo |
| Huawei, HiSilicon | No | Device can differentiate the “different service” based on transaction ID, but cannot differentiate the same or different reader. It is agreed to not consider using reader ID to differentiate the multiple reader case. |
| CMCC | No | From our point of view, the device only need to distinguish whether the received service request is for a new service or for a previous one that it has already responded successfully but **do not** **need to distinguish whether the another (different) service request is from the same reader or from a different reader.**  In terms of multi-reader scenario, it may cause serious interference. Firstly, from the perspective of signal interference, devices located in overlapping areas covered by different readers may simultaneously receive R2D messages from different readers, resulting in low SINR and incorrect reception by the devices. Secondly, from the perspective of affecting the normal procedure, the device that is currently undergoing the corresponding procedure (procedure A) with a certain reader (reader A) may mistakenly treat R2D messages sent by other reader as sent by reader A, thereby affecting the progress of the procedure A. However, enabling the device to distinguish between different readers would bring additional message overhead and additional complexity to the devices, so we prefer to avoid the occurrence of multi-reader scenarios or mitigating the impact as much as possible through implementation.   1. In R19, only D1T1-B where CW node is out of the Topology 1 is supported. So it is very likely that when one reader transmits R2D messages and receives D2R messages, adjacent readers transmit CW waves in actual deployment scenarios. Therefore, multi-reader scenario may also be naturally avoided as much as possible in practical scenarios. 2. For Topology 1, each TRP acts as one reader as discussed in RAN3. To this end, it can be achieved through gNB implementation to control the transmission power of each TRP, thereby avoiding overlapping coverage between adjacent readers. With this, the device(s) is less likely to receive the another (different) service request from different reader. |
| CATT | No | There is no such case that **the another (different) service request is received from the same reader** as following reasons: It has been agreed that parallel service requests by the same reader is not supported, which means the case of device receives another service request from the same reader will not happen. And this **can be guaranteed by reader implementation**, i.e., the reader initiates another service request only if the ongoing service is finished.  We doubt the case that **another (different) service request is received from a different reader** will happen as following reasons:  -If the device receives **another (different) service request is received from a different reader** during its own service procedure at a time, it means the device receives R2D messages in the overlap in time domain from other readers which do not coordinate with other readers. It seems that there is an interference issue during the service procedure because the target device receives and just happens to decode **another (different) service request from a different reader** correctly. This interference will bring service failure to the target device, even when the target device just happens to decode one of R2D messages (the **service request**).  In our view, if the device happens to decode one of the R2D messages from other readers, there is an interference issue to the device in the overlap area. If the interference issue is not controlled in this release, it doesn’t make sense for RAN2 to solve one of the impacts that device happens to decode correctly parts of the interference. On the other hand, there won’t be such issue if the interference between readers is controlled well, for example, the coordination between readers.  So there is no expected device behavior to solve the issue that **another (different) service request is received from a different reader.** |
| Spreadtrum, UNISOC | No | There is not necessary to distinguish the service request is received from the same reader or different reader. We have agreed that the device is expected to only perform one procedure at a time. |
| Apple | No | We prefer to have a unified “transaction ID” space to let the device only proceed with one ongoing transaction at a time From this perspective, the device does not need to differentiate those two cases mentioned in the question. |
| Xiaomi | No | Agree with other companies that it is unnecessary for the device to distinguish the serves from same or different Reader. |
| Tejas Networks | No | If the device is already involved in a service associated with some transaction/service ID, the device will not respond to a different service request.  However, if the device is not associated with any service and it receives two same service requests from two readers at the same time, the device cannot differentiate that the service request is from the same reader or from the different reader by using only transaction ID. The CN will allocate different transaction ID for different services. Hence, same service from two different reader is expected to have same transaction ID. Therefore, a reader ID is also required along with the transaction ID. |
| ZTE | No | We may need to firstly clarify whether the two scenarios for comparison (Scenario #1: another (different) service request is received from the same reader; Scenario #2: another (different) service request is received from a different reader) exist.  It seems companies may have different interpretation for the condition “while there is one ongoing procedure”, regarding whether this condition is from the perspective of the network or from the perspective of the device. Yes, from device perspective, we have agreed that the device is expected to only perform one procedure at a time (assuming it’s RA and data transmission procedure). So it may be common assumption that the device would not handle (or ignore) another Paging before it completes currently ongoing procedure (no matter it ends successfully or fails).  On the other hand, from network perspective, “the ongoing procedure” can refer to an AIoT service procedure that is currently being executed targeting one or a group of devices, e.g., via a Paging triggered by a reader (e.g., reader-A) and may last for a certain time period. Then it may be possible that, another different AIoT service is triggered during this time period and the targeting devices are the same as those of the previous AIoT service. Furthermore:   * We agree with above comments that according to previous agreement, it’s impossible that another different AIoT service is sent from the same reader (e.g., reader-A) when the previous AIoT service procedure triggered by this reader is still ongoing. Even the reader-A may receive another AIoT service request from CN/AF, it can hold this request and delay the processing on it. * From a purely procedural perspective or theoretically, it’s possible that at any time another reader, (e.g., reader-B) can receive another CN AIoT service which is different from the AIoT service received by reader-A but the targeting devices are the same as those in the AIoT service received by reader A. Then the reader-B may trigger another Paging procedure although the Paging procedure trigged by reader-A is still ongoing (reader-B may have no idea on what reader-A is doing).   + Since it’s hardly to assume reasonable motivation for the CN/AF to perform different AIoT services on the same group of devices simultaneously or within a very short interval, we think this sub-cases can be (need to be) avoided by the CN/AF implementation;   + Even if the CN/AF allows different AIoT service to be executed in parallel in order to increase AIoT operation efficiency, we still agree with the above CMCC and CATT views that, considering that interference between R2D messages may lead to a higher AIoT operation failure rate (for both AIoT services triggered by reader-A and reader-B), such concurrency might not actually improve AIoT operation efficiency. Therefore, from the RAN2 perspective, we also suggest not to support this Scenario#2 in R19.   In a summary, the two scenarios for comparison in Q1 are either already prevented due to existing RAN2 agreement, or deemed unnecessary to consider from the RAN2 perspective. |
| InterDigital | No | Regardless it is the same reader or different reader, the device should not handle multiple parallel service requests. |
| MediaTek | No to the question exactly as asked, but see comment | We assume “different” here means “different underlying service leading to a different transaction ID”. Considering agreement 2 above, a device already in random access for service X should never process a paging message from any reader for service Y. This can be achieved with only the transaction ID, \*assuming\* that different services from different readers always (or “acceptably close to always”) have different transaction IDs.  There could be a failure mode if the assumption above is not met and there is no reader ID in paging: Reader A pages for service request 1 with transaction ID X, and then reader B pages for service request 2 with (coincidentally) transaction ID X. The device might not be able to distinguish reader B’s page from a subsequent paging message for service 1 from reader A, leading to spurious re-access. We should avoid this case somehow.  We do not see agreement 5 above as related to this exact question. Per the discussion at RAN2#129 (under P3 of R2-2500129), it seems related to two readers paging for the same service. |
| Nokia | No | But, we do believe we have to be careful not to be worse than RFID by not considering multiple requests in parallel |
| ETRI | No | The A-IoT device performs only one service procedure at a time. However, the reader can distinguish different services through different types of identifiers. |
| Qualcomm | Yes but no spec impact in this release | Based on earlier RAN2 discussions and company comments, RAN2 can assume that the same reader will not initiate a different service request to the same device. In that sense, if the device ends up getting a new service request (although not expected, let’s say for whatever reason it happens), it can be treated as if it is from a different reader. |
| NEC | See comment | **Assume to have no reader id in paging:** If there’s no additional information in paging to identify a reader (e.g., reader id), we believe device cannot distinguish the same reader vs different reader.  However, as we agreed that “parallel service requests by the same reader is not supported”, device may not expect another (different) service request receiving from the same reader. So, upon receiving a different service, device may consider it is from a different reader.  **Assume to have reader id in paging:** In this case, device can distinguish the same reader vs different reader. |
| Samsung | Yes, but without necessity to do such differentiation. | Case 1: “another (different) service request is received from the same reader”   * According to agreement “Parallel service requests by the same reader is not supported”, the reader can ensure that another (different) service request is not triggered for the same device if there is ongoing procedure triggered by one service request, e.g., use TDM pattern to trigger the procedures for different service requests. Thus, there is no need for the same reader to trigger parallel service requests to the same device, which can be avoided by reader.   Case 2: “another (different) service request is received from a different reader”   * As agreed in RAN2, this case is possible, even though RAN2 is not intended to specify something.   Thus, technically, when the device is performing one procedure triggered by one service request, if the same device receives another different service request, it can consider such request is from a different reader.  However, **the device needn’t to have such differentiation**. The device only needs to determine whether to response after receiving a paging message, regardless of whether it is from the same or different reader. |
| HONOR | No | Generally, we agree with other companies that from perspective of the device, there is no need to distinguish the paging message from the same or a different reader. To realize that the device would only be in one procedure, the device could differentiate the service requests to which the received messages belong. So, we share similar concerns with MTK. Based on our understanding, the solution/implementation needs to be discussed and the final decision may be checked by other working groups. For now, we have some potential solutions on the table: the device needs to distinguish the two messages are from different service requests or the NW guarantees that there is no parallel procedure at the same time. |
| Sharp | No | A common “transaction ID” would be enough for an Aiot device to identify the same service and only one procedure is performed by an Aiot device. |
| Futurewei | No | The device can differentiate different service requests based on the Transaction ID. Without reader ID, the device may suspect a Transaction ID far away its current Transaction ID is from a different reader. But this requires the Transaction ID to be sufficiently long. |
| Ericsson | No, please see the comments. | It should not matter whether the request is received from the same or different reader from device standpoint, i.e., no need to introduce a reader ID, considering that it is the transaction ID of the service request that determines whether the device should respond based on the subsequent paging mechanism.  The device is expected to perform only one procedure at a time and the question we need to ask/discuss is **whether the device clearly knows if a procedure is completed/pending.** This is so that the device can determine whether to respond to this service request, i.e. if it has responded successfully to this service request or if it is a new service request.  For example, a device may know whether a procedure is completed if the reader transmits a NACK message indicating an unsuccessful transmission from the device or a QueryRep-like message to advance to the next accession occasion slot implicitly indicating ACK. |

**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** |
| OPPO |  | Regardless of another service request is coming from the same or different reader, the A-IOT device behavior shall be aligned, i.e., only performing one procedure at a time. This will reduce the complexity of A-IOT device. After the end of the procedure the A-IOT device shall respond to the related A-IOT paging message for a different service request |
| CMCC | No | **There is no difference in terms of device behavior.** The device only need to distinguish whether the received service request is for a new service or for a previous one that it has already responded successfully but do not need to distinguish whether the another (different) service request is from the same reader or from a different reader. |
| CATT | See our comment on Q1 | When there is one ongoing procedure, any message from other readers is interference for the device which may bring service failure. It is not necessary for device to solve the specific issue that the device just happens to decode one of messages in the interference. Actually this kind of interference should be controlled in other WGs, e.g. the coordination or the signal design. There won’t be such issue if the interference between readers is controlled well, for example, the coordination between readers. |
| Tejas Networks | See our comment on Q1 | If the device is already involved in a service associated with some transaction/service ID, the device will not respond to a different service request. |
| Qualcomm | See Q1 | As device is not expected to get another request from the same reader, the only possible case is for different reader. |
| Samsung | See comments | Regardless of same reader (if possible) or different reader, the device behavior is same, i.e., the device should ignore the service request with different transaction ID from the one of ongoing procedure. |
|  |  |  |
|  |  |  |

**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. |
| OPPO | The A-IOT device should just discard the following paging message if it has already been in a access process and before the end of the process, i,e, either the device considers that the process has been successfully finished or the process has been failed ( and re-access is needed) . |
| Huawei, HiSilicon | Single reader can control the paging to send. So, the reader shall not send another new paging if there is on-going procedure. It is up to reader implantation to guarantee “Parallel service requests by the same reader is not supported”.  In any case, the device behavior is: If the transaction ID received in the paging message differs from the one currently maintained by the device, the device will update its currently maintained transaction ID to the received value. |
| CMCC | We strongly propose that **the device follows the latest service request received.** From our point of view, the performance of the solution that the device ignores any other paging message while it is in one ongoing procedure is closely related to how for a device to determine one procedure is terminated or not. For example, if an explicit R2D message is used to indicate that the ongoing procedure (procedure A) is terminated and one device (device X) fails to receive this due to poor channel conditions or device moving out of coverage range of previous reader, device X becomes inaccessible because it considers procedure A is still ongoing and ignore any new paging message for different service request. However, the solution that the device follows the latest service request received is a good and simple solution, since it will not affect or be affected by any other discussions in RAN2. |
| CATT | The same reader is not expected to send another service request while there is one ongoing procedure according to RAN2’s agreement: 1. Parallel service requests by the same reader is not supported. |
| Spreadtrum, UNISOC | According to the last meeting agreements, we think this case will not happen. Because the reader can avoid to trigger another service request while there is one ongoing procedure. If it happen, we assume that device will ignore another service request. |
| Apple | The reader should not do that. But it is not device’s responsibility to judge reader’s misbehavior. If this happens, the device simply follows the reader and respond to the latest transaction. |
| Xiaomi | Agree with Huawei. Based on agreements, the device is not expected to receive different service request if there is on-going procedure. We consider this as error case if it happens. Then the simple way is that the device just follows the request from the reader. If it is different transaction ID, the device shall perform the access based on the new ID and replace original ID. |
| Tejas Networks | In our understanding, this issue will not occur as the reader will not start another service request until the ongoing service is completed. However, if the reader triggers different service requests back-to-back, the device will respond to the first one only and discard the subsequent requests within that transaction period. |
| ZTE | Agree with some above comments that according to the previous agreements, we don’t need to discuss this case that another Paging triggered by (different) service request is received from the same reader while there is one ongoing (Paging) procedure. i.e. the reader should avoid this. Then also no need to specify device behavior for this case (similar to no UE behavior specified for network error cases). |
| InterDigital | We also understand that a paging triggered by a different service request should not happen based on reader implementation, and we don’t need to specify any device behavior for this case. |
| MediaTek | We are not sure this case can really be excluded considering group IDs. The reader can be expected not to page the same single device for two overlapping services, but can the reader know if the same device happens to belong to two groups that are inventoried together? We think it’s safer to specify some defensive behaviour by the device.  We understand there are basically two options, to ignore the second paging message (as suggested, e.g., by OPPO) and to abandon the ongoing transaction and handle the new service request (we understand this to be Huawei and CMCC’s suggestion above). Of these two, we tend to see the first one as simpler for the device (but OK to discuss): While a procedure is ongoing, the device simply does not process paging messages. It does assume that the device knows when a procedure ends. |
| Nokia | We understand that since it is another reader it will have another transaction ID, or at least we will then have a corner case where a paging occurs with the same transaction ID as an ongoing procedure.  Such exceptions should be easy to “catch” and handle by the device or in spec i.e.   1. if a reader is handling a service request with transaction ID x and a request with transaction ID y comes, it will only continue to reply to the one with x in header 2. if a reader is handling a service request with the transaction ID x and another request comes with the transaction ID x, we believe this would be visible as not a subsequent paging, and the device should not start the CBRA again. Thus, since the AS id is not sent, or is in another part of the state machine, no issues should occur. |
| ETRI | If the device is performing a service, the reader recognizes it, so this scenario does not occur. However, if the device receives a new service request while already in service, it discards the request. |
| Qualcomm | As commented by other companies, we can assume that this scenario (new service request from same reader while one procedure is ongoing) does not happen. But if it does, it seems reasonable for the device to respond to the newest/latest service request and (since there is already agreement that at max one procedure at a time) consider the earlier/older procedure as terminated. |
| NEC | As we agreed that “parallel service requests by the same reader is not supported”, device may not expect another (different) service request receiving from the same reader. |
| Samsung | This case cannot occur according to RAN2 agreement. Even if it occurs, the device simply ignores another (different) service request. |
| HONOR | This question is somehow relevant with the previous one. If there is only one procedure targeting the same device, the reader triggered the previous procedure could overwrite the previous one with a new one. But if there is another procedure triggered by other readers, the device would always drop the current one and start the counting for the new one which may cause a large latency to finish one service request.  Thus, we prefer the device would stick to the previous procedure (transaction A) and ignore the new procedure (transaction B) unless it is told that the procedure (transaction A) is ended (by a explicit indication or the default behavior) and stop monitoring the previous one. |
| Sharp | As agreed that “parallel service requests by the same reader is not supported”, Aiot device is expected to neglect the new service while one procedure is performed. |
| Futurewei | Since the device always follows the instruction from the reader, the device should follow the newly received paging and abandon the prior service request even if the device has not responded successfully yet. It is up to the reader to ensure that paging with the new Transaction ID is not sent until the prior service request is terminated. |
| Ericsson | In RAN2#129, it was agreed that parallel service requests by the same reader is not supported. But this particular discussion is not necessarily about whether a reader can break that principle and what happens if it is so. As we tried to explain in our response to Q1 above, this is a matter of whether the device and the reader having the same assumption regarding if the procedure is completed/pending, for example, considering that RAN2 has agreed to support NACK feedback. Note that the same reader can initiate a new service request if it assumes that there is no ongoing procedure as opposed to a device.  We should discuss this aspect first and conclude when a device/reader can consider that a procedure is completed. |

**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. |
| OPPO | Regardless multi-reader or one-reader scenario, the A-IOT device behavior is the same: only performing one procedure at a time. |
| Huawei, HiSilicon | Device behavior is same as above Q3, i.e. update the stored transaction ID.  In our assumption, it is still up to the NW implementation to avoid the interleaved/parallel services in multiple reader case.  If multiple neighbor readers deployed belonging to the same BS, which should be one typical deployment, the coordination can be done by this BS implementation, to control the order of the services triggered by readers.  In case of multiple readers deployed belonging to the different BSs, which is not typical deployment for indoor scenario (i.e., one BS should be able to cover one factory), it is up to OAM to configure the isolate resources among BSs. As one example below, reader will not perform one service procedure across two discontinuous resources.    Please note that pure device solution, e.g., “device ignoring the new service”, does not work unless the transaction ID is carried in **all** R2D message. This is because, in case of interleaved services from different readers, the R2D messages (other than paging) from other reader-2 will impact the ongoing procedure of device under this reader-1. Hence, it requires transaction ID in all R2D message to let the device to filter out the R2D messages from neighbor reader.    Therefore, this issue has to be left to network. |
| CMCC | **The device follows the latest service request received.** First of all, we think that thedevice behavior should be same 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. Besides, same as our comment to Q3, the performance of the solution that the device ignores any other paging message while it is in one ongoing procedure is closely related to how for a device to determine one procedure is terminated or not. However, the solution that the device follows the latest service request received is a good and simple solution, since it will not affect or be affected by any other discussions in RAN2. |
| CATT | There is no expected device hebavior if another (different) service request is received from a different reader while there is one ongoing procedure. Please find our comment on Q1. |
| Spreadtrum, UNISOC | Device behavior is same as Q3, device will ignore another (different) service request. |
| Apple | The device need respond to the latest paging with the new “transaction ID” if it happens to be matched with this paging ID. This is consistent with the current design. |
| Xiaomi | Agree with Huawei and CMCC. The device shall just follows the latest service request since it is difficult for the device to distinguish whether it is error case or the reader wants to trigger the procedure for new service and cancel original service. |
| Tejas Networks | In our understanding, this issue will not occur as the reader will not start another service request until the ongoing service is completed. However, if the reader triggers different service requests back-to-back, the device will respond to the first one only and discard the subsequent requests within that transaction period. |
| ZTE | If Scenario#2 that another (different) service request is received from a different reader while there is one ongoing (Paging) procedure happens, the general UE behavior can be:   * For those devices which are still within ongoing RA/data transmission procedures, it cannot respond to the latter Paging message corresponding to anther different service request. * For those devices which have already finished their RA/data transmission procedure (successfully or failed) within the current Paging procedure, if the device can determine by some way that the latter Paging message was triggered by a CN AIoT service request that is different from the one that triggered the previous Paging message, the device can simply respond to the latter Paging.   However, according to our comments in Q1, we assume Scenario#2 itself should be avoided. |
| InterDigital | Same as answer in Q3. |
| MediaTek | As with Q3, we tend to think it’s simpler and cleaner for the device to finish the ongoing procedure and ignore any paging messages until it’s done. |
| Nokia | See Q3 |
| ETRI | Same as answer in Q3. |
| Qualcomm | Similar to Q3, it seems the device should respond to the new request and consider the previous procedure as terminated. Ignoring the new request and holding on to old procedure may be less preferred as that can create other issues – e.g. if the old procedure was not fully completed for whatever reason (device missed final ACK, or reader went through a failure and powered on again etc.) the device would be stuck with old procedure while the reader cannot legitimately start a new procedure towards the same device.  Regarding Huawei’s comment, it maybe not that difficult to do the TDM of readers in Topology 1, but when topology 2 is also in the picture (in future), that is not straightforward how the different readers guarantee the TDM. |
| NEC | Device may ignore the different service if the ongoing service is not completed. |
| Samsung | This depends on the transaction ID contained in the paging message from a different reader.  If the transaction ID is same as the ongoing procedure, the device needn’t response it again.  If the transaction ID is different from the ongoing procedure, the following options can be considered:   * Option 1: ignore the new paging message, and continue the ongoing procedure   In this option, after the device complete the ongoing procedure, it can also receive the paging message from another reader, and perform the related procedure.   * Option 2: stop the ongoing procedure, and perform the new procedure corresponding to the new transaction ID * Option 3: store the new transaction ID, and start the new procedure corresponding to the new transaction ID until the ongoing procedure is completed.   Among those options, we prefer to option 1, which is simple and reasonable. |
| HONOR | Same as answer in Q3. |
| Sharp | Device behavior is same as Q3 |
| Futurewei | If the transaction ID is sufficiently long and the device is able to differentiate the readers based on the transaction ID, then the device may be able to respond selectively. Otherwise, the device always follows the latest service request. |
| Ericsson | As mentioned in a response to an earlier question, the most important aspect here is to conclude how a device should determine whether a procedure is completed/pending. If the procedure is pending it should not matter whether the new request is transmitted with the same or different transaction ID from the same or different reader. If the procedure can be considered completed the device responds depending on whether the procedure is successfully completed and/or transaction ID is same/different, i.e., inline with the subsequent paging mechanism.  Regarding whether the device should respond to a new request or ignore the new request; both options have pros and cons. We have no strong preference in between these options as long as RAN2 concludes how a device should determine whether a procedure is completed/pending in all cases. |

**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. |
| OPPO | Maybe | If we achieve the agreement that in multi-reader scenario, we need to let the A-IOT device to respond to A-IOT paging messages corresponding to the same service request coming from different readers, then the device shall distinguish whether the same service request is received from the same reader or a different reader, since the device behaviors are different when receiving paging message from the reader vs from a different reader.  On the other hand, we don’t find any clue that in the last RAN2 meeting we agree to rely on **CN** implementation. Even if CN assign readers with different correlation ID, still it could be possible for the readers to use the same transaction ID to trigger the A-IOT paging message, since the transaction ID space is so limited. |
| Huawei, HiSilicon | No | If they are “same service”, i.e. using the same transaction ID, without differentiating the one from same vs. different readers, device always uses the transaction ID to determine whether to skip the subsequent paging or not. |
| CMCC | No | **The device do not need to distinguish whether the same service request is received from the same reader or from a different reader** as well. Same as our comment to Q1, enabling the device to distinguish between different readers would bring additional message overhead and additional complexity to the devices, so we prefer to avoid the occurrence of multi-reader scenarios or mitigating its impact as much as possible through implementation.   1. In R19, only D1T1-B where CW node is out of the Topology 1 is supported. So it is very likely that when one reader transmits R2D messages and receives D2R messages, adjacent readers transmit CW waves in actual deployment scenarios. Therefore, multi-reader scenario may also be naturally avoided as much as possible in practical scenarios. 2. For Topology 1, each TRP acts as one reader as discussed in RAN3. To this end, it can be achieved through gNB implementation to control the transmission power of each TRP, thereby avoiding overlapping coverage between adjacent readers. With this, the device(s) is less likely to receive the same service request from different reader. |
| CATT | No need | There is no need to distinguish.  Because we already achieved common understanding that device will rely on transaction ID and implementation to handle the multi-reader scenario, the device determines whether to respond the paging messages only relying on the transaction ID so that the device does not need to distinguish the paging messages of a certain service request from the same reader or from a different reader.  However, similar as our comment on Q1, we may investigate whether this issue “**same service request is received from a different reader**” makes sense. If multiple readers do not have coordination and initiate their own paging messages for the same service request in the overlap area at the same time, any messages from other readers are the interference for the device, whatever the reader ID or service ID is included in these R2D messages.  In a summary, interference issue won’t be solved just by adding the reader ID or service ID in the paging message. There are multiple R2D messages following paging message from other readers are still interference for the device. The coordination between readers is required to make sure the whole procedure works smoothly, not only just for paging message.  Furthermore, even if the multiple readers have resource coordination in advance and perform the service request in order (e.g., reader1: [t1,t2]; reader2: [t2,t3]), we wonder when does the device decides to release the stored transaction ID (should be [t1,t3]?) and how does the device determines the time point. This issue makes the device behavior in avoiding duplicate responses toward the same service request quite complex, so we prefer not to further enhance the subsequent paging mechanism facing the overlap case. |
| Spreadtrum, UNISOC | No | The Device do not need to distinguish whether the same service request is received from the same reader or from different reader. From device perspective, it only needs to respond once according to the transaction ID of a service.  For multi-reader case proximity service, if device is expected to respond to multiple readers for the same service, it can up to CN implementation to assign different correlation ID and reader can generate different transaction IDs for the same service. |
| Apple | Up to NW implementation | It is up to NW to decide to use the same transaction ID or use different transaction IDs for multiple-reader case. Either way, the device behaviour is consistent, the device will only proceed with the one (latest) transaction ID. |
| Xiaomi | No | Do not see the need to distinguish whether the message is from the same or different reader. It can be done by NW implementation, e.g. same transaction ID or trigger the procedure one by one. |
| Tejas Networks | No  [Rapp: based on comment and Q6 answer, it seems this is meant to be Yes? Because the question is ‘should it be possible?’, not ‘is it already possible?’] | Without reader ID, the device cannot differentiate the same service is from the same reader or from different reader. Although the device will prioritize the service request it receives first and discard the same repeated service requests received within that transaction period, it might respond to the same service from different reader once the current transaction period is over. Therefore, in our understanding reader ID also should included. |
| ZTE | See the comments | Similar as the discussion for Q1, we think for Q5, we also needs to firstly clarify whether the two scenarios for comparison (Scenario #3: same service request is received from the same reader; Scenario #4: same service request is received from different readers) exist.  As mentioned in some above comments, RAN2 has already agreed that one reader (e.g., reader-A) can trigger multiple Paging (rounds) for the same service request to increase the successful AIoT inventory rate, so we understand the Scenario #3 is possible, e.g., another Paging corresponding to the same AIoT service can be sent by the same reader even there is an on-going (Paging) procedure.  Meanwhile, it’s also possible that an adjacent reader, e.g., reader-B receives same CN AIoT service (targeting the same group of devices) as that received by reader-A. The possible reason for this could be that one CN AIoT service request is sent to multiple AIoT readers, e.g., reader-A and B (due to that CN has no priori/existing knowledge on the association between the reader and the target devices). So we understand from network perspective, the Scenario #4 may be possible:   * From device perspective, we haves some sympathy with the views from CMCC, or CATT that the Scenario#4 may not need to be supported due to the potential interference. We also think due to the very limited demodulation capability of the devices, even the different readers can send Paging messages corresponding to the same AIoT service request, it is unlikely the device could demodulate two paging messages at the same time. * However, we think it is better to separate the discussion of the scenario from the discussion of the solutions.   Back to Q5, rather than discussing whether the device needs to distinguish them, it’s more needed to discuss the actual device behavior for these different scenarios (if supported). There may be no need for explicit device behavior to be specified to distinguish between these two scenarios, but the device procedure for these two scenarios should be clear from the specification perspective, see our further comments in Q7 and Q8. |
| InterDigital | Yes | We have the same understanding as Lenovo and ZTE. Whether this can be handled by use of a different transaction ID can be discussed in Q7 and Q8. |
| MediaTek | Yes (if the device is processing paging messages at all) | We are a little surprised by all the “no” answers, as it seems to us that the device behaviour needs to be different while the UE is in random access (after random access, as discussed above, we think all paging should be ignored until the procedure finishes). We also think that, although the intention is to avoid reader overlap, it is reasonable to design the device defensively (deployment results might not always be perfect).  The same service request from the same reader is a “subsequent paging” case and should be processed in case it needs to trigger re-access for this device.  The same service request from a different reader should be ignored by a device already handling the service through the first reader (otherwise we have parallel procedures).  We could achieve this distinction by guaranteeing somehow that different readers always generate different transaction IDs or by including a reader ID in paging, and maybe there are other solutions, but we should address it. We are a bit concerned about agreement 5 above in this context; if the implication of trusting the “implementation” (we assume network/reader implementation is meant here) is that transaction IDs from different readers never collide, we should capture that assumption and discuss which group enforces it. |
| Nokia | Yes | Agree with MediaTek |
| ETRI | Yes or No, with comments | When the CN’s A-IoT AF delivers service-related information and the reader generates a transaction identifier based on this information and passes it to the device, the device can determine whether it is the same service request or a different one. However, if only the transaction ID is used without this information, the device cannot distinguish whether the service requests received from different readers are the same or not. |
| Qualcomm | Yes | Similar comment as explained by Lenovo and OPPO. E.g. for location use case, it is possible that *same* service request may be received from *different* readers and there could be need for the device to respond to all of them.  Further, as explained by MediaTek, the design should be defensive as deployments are not always perfect. |
| NEC | See comment | **Assume to have no reader id in paging:** If there’s no additional information in paging to identify a reader (e.g., reader id), we believe device cannot distinguish the same reader vs different reader.  **Assume to have reader id in paging:** In this case, device can distinguish the same reader vs different reader. |
| Samsung | Maybe | The device may be able to differentiate it, e.g.,   * Use the Msg1 resource   + One possible case is that the device receives the first paging message with CFRA resource, and it receives another paging message with the same transaction ID and a different CFRA resource when the device is performing the procedure triggered by the first paging message. * Use upper layer data in Paging message (if any)   + Another possible way is that the device can differentiate it based on the Upper layer data in Paging message.   Also, this question is related to how the transaction ID is generated, e.g., whether different readers can generate the same/different transaction ID for the same service. |
| HONOR | No | No need to distinguish. We prefer a consistent device behavior: device would not re-response the paging with the same transaction ID and would response to the paging with different transaction ID.  For the case that same transaction ID for the different service request, even the paging from reader B is for another service request (service request 2). The reader or CN could re-trigger the paging with a refreshed transaction ID. The NW could receive the device response with latency which is tolerable.  For the case that different transaction ID for the same service, redundant response from the device is received by the NW but the NW or the AF could remove duplicates by implementation. Considering the complexity brought by the solutions, the energy waste caused by this rare case is tolerable. |
| Sharp | No | As we response in Q1, a common transaction ID could avoid to distinguish this case. |
| Futurewei | No | If they are truly the same service request (i.e., the network only wish to receive one response from a device), the same transaction ID should be used across different readers. If the network wishes the device to respond to all the readers, then the network should ensure that different Transaction IDs are used. |
| Ericsson | No | We do not think there is a need for the device to know whether the same service request is received from the same/different reader. We agree that this can be beneficial as it may provide more accurate positioning, but existing accuracy, i.e., reader coverage area, is good enough for Rel-19. |

**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. |
| OPPO | yes | For the same-reader case, the device shall not respond to the message if it has already successfully finished the procedure before. For the multi-reader case, the device shall respond to the message |
| CMCC | No | **There is no difference in terms of device behavior.** |
| Apple | No | The device behaviour is consistent. |
| Tejas Networks | Yes | The device behavior can be different if it can differentiate the service is from different reader. The device may choose to respond to the service from the different reader if it is not involved with any service request from the previous reader. |
| ZTE | Maybe | See the detailed comments in Q7 and Q8. |
| InterDigital | Yes | See responses in Q7 and Q8. |
| MedisTek | Yes | See our answer to Q5. Here we interpret that “has previously responded” means that Msg1 has been sent, not that the device has completed the whole random access procedure. |
| Nokia | Yes | Agree with Mediatek |
| Qualcomm | Yes | Similar view as Lenovo and OPPO. |
| Samsung | See comments | We can leave this question open for now. After we have clear answer to Q5, we can discuss this. |
| Sharp | No | There is no difference considering the same service. |
|  |  |  |

**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. |
| OPPO | Agree with vivo. If successfully finished, then the device shall skip the following same service request; otherwise, the device shall respond to the same service request. |
| Huawei, HiSilicon | The device behavior for such case has been agreed already: if the device has successful performed the same procedure, device will not response the subsequent same service request. Otherwise, device will response. |
| CMCC | **If one device has previously responded and completed the corresponding procedure successfully, it ignores the same request, otherwise, it attempts to respond/re-access.** |
| CATT | If the device has successfully responded to the service request, it shall not respond to the subsequent paging messages. (If the device previously responded to the service request but with failure, e.g., received NACK for msg3, the device responds the subsequent paging message for re-access) |
| Spreadtrum, UNISOC | As discussed in SI phase, if the same service request is received after device has successfully responded to the service request, device will skip to respond to the same service request to avoid duplicate responses. |
| Apple | Based on the prior agreement, the subsequent paging with same transaction ID will be ignored by the device. |
| Xiaomi | Agree with other companies that it has been agreed before, i.e. the same device shall ignore the paging for the same service if the device has performed it successfully. |
| Tejas Networks | Device will check the transaction ID. Since the transaction ID is same, the device will not respond to the same request from the same reader once it successfully completes the same transaction. |
| ZTE | Agree with most of the above understanding on the device behaviors, e.g., one device has previously responded a Paging message and completed the corresponding RA/data transmission procedure successfully, it can ignore the latter Paging corresponding to the same request, otherwise, if failed in previous RA/data transmission procedure, it attempts to respond to the latter Paging for re-access.  Also based on the previous agreement, we assume for this Scenario #3, the transaction ID in the multiple Paging messages corresponding to the same AIoT service request need to be **same**. By this way the device can determine whether the Paging messages received previously and subsequently belong to the same AIoT service. |
| InterDigital | Agree with ZTE. |
| MediaTek | As noted above (Q5) and by other respondents, the device needs to handle this case as “subsequent paging” with the potential to trigger re-access. If the device considers that it responded successfully the first time, it will drop the page on the floor. |
| Nokia | Agree with MediaTek |
| ETRI | Based on the agreement in the SI phase, any subsequent paging with the same transaction ID will be disregarded by the device. |
| Qualcomm | Similar to other companies’ views: if device has successfully responded and receives another request for same transaction ID, it can ignore. However, it should be clear that transaction ID can wrap around, so the **‘same’ here should mean ‘immediately preceding’**. |
| Samsung | The question is a bit unclear i.e. especially what “after device has previously responded to the same service request” really means:   * If it means after device performs inventory procedure successfully, the device does not need to respond the same service request from the same reader further. * If it also covers the case that some devices performed inventory procedure but failed or are still under on-going inventory procedure, then depending on the content of subsequent paging, the device may need to (re-)access. |
| HONOR | The device would skip the subsequent one with the same transaction ID. |
| Sharp | The device would ignore the same service request after device has previously responded to the same service request. |
| Futurewei | This has been covered by earlier agreement, i.e., the device should skip the service request. |
| Ericsson | The device should ignore the service request if it has responded to a service request with the same transaction ID successfully. |

**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. |
| OPPO | If we achieve the agreement that in multi-reader scenario, we need to let the A-IOT device to respond to A-IOT paging messages corresponding to the same service request coming from different readers, the A-IOT device shall respond to the A-IOT paging message corresponding to the same service request coming from another reader. |
| Huawei, HiSilicon | 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.  Based on the RAN2 agreement, “same service request” to the device means the same transaction ID. So, it is up to the CN to use the suitable correlation ID for each service.  In the case in Q8, it is the CN choice to use same correlation ID for different readers for this “same service”. Then, CN expects the device to not redundantly response from different readers. |
| CMCC | The device behavior is same as our comment to Q7, that is, **if one device has previously responded and completed the corresponding procedure successfully, it ignores the same request, otherwise, it attempts to respond/re-access.** |
| CATT | The same answer as Q7 |
| Spreadtrum, UNISOC | As answer in Q5, for multi-reader scenario, it dependents on CN implementation to assign different or same correlation ID for different reader of the same service. If CN assign different correlation ID for different reader corresponding to the same service request, and reader generate the different transaction ID, device will response the service request. If CN assign the same correlation ID for different reader. The device will ignore the service request. |
| Apple | Based on the prior agreement, the subsequent paging with same transaction ID will be ignored by the device. |
| Xiaomi | Same as the comments in Q7, the same device shall ignore the paging for the same service if the device has performed it successfully. |
| Tejas Networks | The device behavior can be different if it can differentiate the service is from different reader. The device may choose to respond to the service from the different reader. |
| ZTE | In Q5, we have analyzed the possibility of this Scenario #4 and the reasons that lead to Scenario #4. So we think the UE behavior needs to be discussed in details:   * (sub-case#4-1) A general thinking is that for a certain device, e.g., devic-1, if it can determine a received Paging is from another reader and corresponds to the same AIoT service request, the UE behavior can be same as that for Scenario #3, e.g., after the device has previously responded a Paging message and completed the corresponding RA/data transmission procedure successfully, it can ignore the latter Paging corresponding to the same request from reader-B as the inventory result of device-1 can be delivered to CN/AF by the reader-A.   + For this sub-case, to make Paging messages from different readers but triggered by the same AIoT service contain the same transaction ID can facilitate the above mentioned device behavior (e.g., to ignore the latter Paging) and also simplify device operations, meaning that the device does not need to distinguish whether Paging messages with the same transaction ID come from the same or different readers. * (sub-case#4-2) But in some special case, e.g., as mentioned in some previous meeting contributions, for the purpose of coarse positioning of the device, the network may intentionally have multiple readers inventory the same batch of devices and expect these devices to respond to the requests from each reader. So for those devices which have already finished their RA/data transmission procedure within the current Paging procedure, if the device can determine by some way that, the latter Paging message comes from a different reader and triggered by a same AIoT service request, but redundant response is still required, the device needs to respond to the latter Paging:   + To facilitate such device behavior, one option is that, similar as that in sub-case#4-1, to make Paging messages from different readers but triggered by the same AIoT service still contain the same transaction ID, while giving additional information about reader or other indication that redundant response is needed.   + Another option is that (maybe simpler), to make Paging messages from different readers just contain different transaction ID (although they are triggered by a same AIoT service). Since the latter Paging message contains different transaction ID from that in previous Paging, the device can simply respond to the latter Paging. It’s easy to be seen that if going this option, it may cause unnecessary redundant responses in the sub-case#1.   In a summary, RAN2 needs to firstly confirm whether all the above sub-cases in Scenario#4 need to be addressed (e.g., whether sub-case#4-2 needs to be addressed?)  Furthermore, RAN2 can discuss which alternative can be a baseline assumption:   * Alt1: to make Paging messages from different readers but triggered by the same AIoT service still contain the **same** transaction ID (may need other indication to make this Alt1 applicable to all the sub-cases). This alternative seems be assumed by more companies above? * Alt2: to make Paging messages from different readers but triggered by the same AIoT service contain **different** transaction ID (unnecessary redundant responses may need to be further addressed in some sub-cases)   After selecting a baseline assumption, we can further discuss what kind of mapping from correlation ID to transaction ID can be workable? Also we can discuss whether or not inter-reader coordination is needed and if yes, what’s the requirement from the RAN2 perspective (may need to inform RAN3)? |
| InterDigital | Same view as ZTE. As for the alternative, we prefer Alt1 because it means there is no need for coordination of the transaction IDs between different readers. |
| MediaTek | As noted above (Q5), the device needs to ignore this paging message if it is still in the first response procedure (after completing random access). If the device is still in random access with reader A, we should avoid having it interpret a page from reader B as a re-page from reader A.  If reader A pages for transaction ID X, the device responds successfully and completes the procedure, and then reader B pages for transaction ID X, it raises the question of how/when the transaction IDs can be reused—does the device know if this is the same service, vs. a new service that uses the same transaction ID after some time? We would prefer to keep the device stateless and consider the second instance of transaction ID X as a new procedure in this case. |
| Nokia | Agree with MediaTek |
| ETRI | Same as the comments in Q7, any subsequent paging with the same transaction ID will be disregarded by the device. |
| Qualcomm | Unless some additional information such as reader ID is supported, the device cannot figure out whether the new request is from the same or different reader. |
| Samsung | We may need some clarification on “after device has previously responded to the same service request”, whether this is a successful response or a failed response.  If it just means the successful inventory procedure, the device behavior depends on the transaction ID in the paging message from a different reader:   * If it is different from the one that the device previously responded, the device should response it * If it is same as the one the device previously responded to, the device needn’t response it.   Whether the same service from the different readers is assigned the same transaction ID or not is up to NW implementation or RAN3 stuff. |
| HONOR | Same as Q7. |
| Sharp | Same as Q7 |
| Futurewei | If same service request means the same Transaction ID, then the device will ignore the seemingly redundant service request. |
| Ericsson | The device should ignore the service request if it has responded to a service request with the same transaction ID successfully, i.e., same principle as in Q7. This is based on the assumption that it is not possible for the device to know whether the service request is from a different reader, i.e. no reader ID.  Regarding the discussion on location accuracy, we understand the motivation for providing an indication, e.g., reader ID, so that the device responds to a service request with the same transaction ID to which it has responded successfully. We think reader coverage level accuracy can be considered sufficient in Rel-19, but even if further accuracy is required, this can be achieved by network implementation, e.g., neighboring readers are provided service requests with different transaction IDs by the CN. |

**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. |
| OPPO | No | For multi-reader scenario, only if transaction ID bit space is quite large, then even without reader ID or coordination between the different readers on usage of the transaction ID, the probability of collision of the transaction IDs to be used for different service requests from different reader could be low. However, that will require lots of signaling overhead, maybe more than 10 bit is required.  For same-reader case, we should rely on the reader implementation to not issue a recently used transaction ID to a new service request paging message to avoid the wrap around problem, if the transaction ID bit space is reasonable large. |
| Huawei, HiSilicon | Yes | It is already agreed. |
| CMCC | Yes | **Transaction ID is sufficient for device to confirm whether the service request is the one that the device has already successfully responded**. To be specific, the device will save the transaction ID, of which it has responded successfully, in its volatile memory as long as it has enough energy. When another paging message is received, the device compare the transaction ID in the paging message and the one it has saved. If same, the device confirm the service request is the one that the device has already successfully responded and ignore this paging message, otherwise, it confirms that the received paging message is for a new service request. |
| CATT | Yes | Based on our comment on Q5, apart from the transaction ID, there is no need for the device to determine whether the same service request is received from the same or different reader. The readers behavior e.g. coordination is expected to solve the interference issue between readers. |
| Spreadtrum, UNISOC | Yes | We have agreed to rely on transaction ID and implementation to handle multi-reader scenario. |
| Apple | Yes | As agreed in RAN2#129 meeting, one transcation ID will cover all cases. |
| Xiaomi | Yes | Transaction ID is sufficient as agreed in last meeting, We can rely on transaction ID and implementation to handle it. |
| Tejas Networks | No | The transaction ID is not sufficient to different same service from same/different reader. |
| ZTE | Maybe No | According to our comments in Q7 and Q8, we think pure transaction ID may be not sufficient to achieve the expected UE behavior, if all the sub-cases in Scenario#3 and Scenario#4 need to be addressed. |
| InterDigital | No | As mentioned by ZTE, if there is a need to support coarse positioning, it is not clear that transaction ID alone is sufficient. |
| MediaTek | Depends on assumptions about the transaction ID | As described in our previous answers, the device needs to handle the paging message differently depending on same/different readers. The transaction ID is sufficient if different readers will “always” (i.e., with an acceptable failure frequency) allocate different transaction IDs \*even for the same service\*, but if different readers may allocate the same transaction ID for the same service, the device cannot distinguish these cases from the transaction ID alone. |
| Nokia | Commend | We agree with MediaTek, this is also why we would like to at least specify a method for the transaction ID to be dependent on the reader. We don’t need a dedicated fixed reader ID, but an assumption that readers will be configured to not use the same (can be up to NW implementation.  The most important thing to note is that the ID size needs to be slightly larger than if only transaction ID was used. |
| ETRI | No | The transaction ID is sufficient if it includes the service information(e.g. CN correlation ID) received from the CN. Otherwise, an additional identifier, such as a reader ID, is required to differentiate between readers. |
| Qualcomm | No | It is a tradeoff between the size of transaction ID and need for reader ID. If transaction ID space is small, it would be impossible to make sure transaction ID is not reused across readers. Furthermore, wrap around issue is also there. So, either transaction ID needs to be large enough or another way is to add reader ID. |
| Samsung | See comments | The usage of transaction ID is to determine whether the device needs response to the received paging message or not. It has nothing to do with the differentiation on whether the same service request is from the same or different readers. |
| HONOR | See comments | The transaction ID could not provide enough information about whether paging with the same transaction ID is from the same or different reader or whether it is for the same service request. For Release 19, we think there is no motivation for the device to differentiate the request is from the same or different reader. |
| Sharp | Yes |  |
| Futurewei | Yes |  |
| Ericsson | Yes | Transaction ID would be sufficient, and this has already been agreed. However, the question is how this transaction ID is generated, e.g., whether it is the CN correlation ID or a function of correlation ID, or a function of correlation ID with some reader specific aspects.  We do not think there is a need for an additional reader ID or a need for using reader specific aspects when generating the transaction ID since this would require coordination in the network which would increase complexity. |

**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** |
| OPPO | For multi-reader scenario, to confirm that if the received service request comes from a different reader at 100% confidence level, a reader ID is required to be carried in the A-IOT paging message. |
| Tejas Networks | A reader ID is required to be carried along with transaction ID |
| ZTE | Similar view as OPPO.  If sub-case#4-2 needs to be addressed, it may further need reader ID information (can integrated to transaction ID) or other indication about requiring redundant responses. |
| InterDigital | Same view as OPPO and ZTE. |
| MediaTek | We see two solutions:   1. Guarantee separate transaction IDs at different readers even for the same service (this probably goes beyond RAN2 scope to ensure); or 2. Put a reader ID in the paging message (RAN2 can do this).   We think it looks easier to have the reader ID and avoid complications to coordinate between different readers. |
| Nokia | Agree with MediaTek |
| ETRI | Reader ID or Transaction ID that includes the service-identifying information(e.g. CN correlation ID) received from CN(. |
| Qualcomm | We also think reader ID is needed. This reader ID doesn’t have to be same as the ‘global’ reader ID being discussed in RAN3 but can be a shorter version for the air interface. And it is preferable to have a reader ID which can be optional, and keep the transaction ID space smaller, instead of increasing the transaction ID size to cover all the cases. |
|  |  |

**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. | |
| OPPO | No | | Bearing in mind that topology 2 needs to be supported in future. For future-proof, we don’t know think that coordination between the readers shall be pursued. For topology 2, this will require the SL communication capability of readers. For instance, the UE reader, in the first step, shall find a neighbor UE reader via SL discovery procedure. | |
| Huawei, HiSilicon | Seems no | | Reader can use the rightmost 3 bits of the correlation ID from CN as the transaction ID. Or, other reader implementation example can map one CN correlation ID to its transaction ID, to somehow make the transaction ID different if the correlation ID is different. | |
| CMCC | No | | **No coordination between the readers when generating transaction ID is needed**. As we introduced in comments to Q1, there are solutions to avoid the multi-reader scenario from occurrence or to mitigate its impact as much as possible. In case that the scenario where multiple readers overlap really happens, we prefer not to specify something for optimizing this in R19. | |
| CATT | No | | It was agreed that the transaction ID can be generated by reader based on CN correlation ID. We understand that the readers should generate the transaction ID based on the correlation ID with a certain method, i.e., if the CN sends the same correlation ID to multiple readers, these readers should generate a same transaction ID. Otherwise, if the CN send different correlation IDs, the readers generate different transaction IDs. Then, coordination between readers for generating transaction ID is not needed. | |
| Spreadtrum, UNISOC | No | | It seems that there is no need for readers to cooperate to generate transaction IDs. Reader can generate transaction ID based on predefined rules or by taking the lowest few bits of the correlation ID. | |
| Apple | Possible | | It is up to NW implementation and out of RAN2 scope:   1. If multiple readers use the same transaction ID for the same service request, then the reader can simply use the correlation ID 2. If multiple readers need use different transaction ID to distinguish different readers, then SA2/RAN3 need come out with a scheme to assign a prefix to each individual reader to use to generate transaction ID. | |
| Xiaomi | No | | If the transaction ID is provided by the CN, the coordination is not needed;  If the transaction ID is generated based on the ID from CN, the fixed rule should be sufficient, e.g. x bits of MSB or LSB. There coordination is also not needed for this alternative. | |
| Tejas Networks | No | | If CN is generating transaction IDs, coordination between the readers is not needed.  If reader is generating transaction ID based on correlation ID from CN, all the readers can follow a fixed rule to generate the transaction ID. Hence, coordination between the readers is not needed. | |
| ZTE | Hope No but may depend on solution | | We think for one reader case, the setting of transaction ID should at least ensure that Paging messages triggered by two consecutive different AIoT service requests (with different correlation IDs) have distinct transaction IDs. Therefore:   * One simple way is to sequentially encode the transaction ID for the triggered Paging messages, e.g., the first Paging message triggered by the firstly received AIoT service request would have transaction ID #0, the second Paging message triggered by the secondly received AIoT service request (containing different correlation ID) would have transaction ID #1, and so forth. When the maximum value of transaction ID range is reached, the encoding restarts from #0.   + With the “transaction” concept of the transaction ID, we assume for one or multiple Paging rounds triggered by same service request, the transaction ID may need to be included at least in all the Paging messages, or maybe in other broadcast R2D messages. Whether transaction ID can be skipped in some broadcast R2D messages for reducing signaling overhead can be further discussed. * Using truncated correlation ID, e.g., taking rightmost X bits of the correlation ID as transaction ID might result in the same transaction ID for Paging messages triggered by different AIoT service requests. That may be undesirable. For example, the transaction ID length is 2 bits (X=2), and the first AIoT service request has correlation ID = 5 (“101”) while the second one has correlation ID = 9 (“1001”), this could lead to conflicts that both first triggered and second triggered Paging messages have same transaction ID of #1.   The transaction ID setting for sub-case#4-2 needs further discussion, based on the output of the scenario and baseline assumption discussion.  BTW, for correlation ID, we may have kind of different view from some above. We think we cannot assume the CN’s setting for correlation ID can handle all the cases. In other word, the CN should not apply too complex/flexible logic on setting it (also depending on the definition of correlation ID). The most suitable way may be to directly associate it with different service requests and/or different readers, as more complex logic (e.g., setting same correlation ID to different service requests/readers or different correlation IDs to the same service request/reader) may hide real information. | |
| InterDigital | No | | We should avoid this, especially for forward compatibility to topology 2. | |
| MediaTek | Yes, if there is no reader ID in the paging message | | See our answers to Q9/Q10. | |
| Nokia | Depends | | Should be possible for network to do this correctly as long as the transaction ID accounts for the AIoTF part of the ID and reader part. | |
| ETRI | Maybe No | It depends on the network implementation. | |
| Qualcomm | Depends how correlation ID is defined | | If there is a fixed rule defined between the correlation ID and transaction ID, then reader coordination does not seem to be needed. However, if transaction ID is to be used to differentiate the request is from different readers as commented by companies in previous questions, then it depends on how correlation ID is defined by SA2. If correlation ID is same for same service even for different readers, then those readers need to make sure not to use the same tx ID (and coordination is needed). On the other hand, if correlation ID is always different between readers even for the same service, then the readers may not need to coordinate and can simply derive tx ID based on correlation ID. So, it is dependent on how SA2 defines the correlation ID. | |
| Samsung | No. But is it RAN3 decision? | | How to generate the transaction ID depends on how this ID is used at the device side.  In our opinion, the transaction ID is used by the device to determine whether a response to the reader is needed or not. If the device receives a new transaction ID different from the one that the device has already responded to, the device should response it; otherwise, the device needn’t response.  Based on the above handling, the reader should generate the transaction ID to realize the following three cases:   * Case 1: for the same service, if the same device is required to response towards different readers, different readers should generate different transaction IDs * Case 2: for the same service, if the same device needs response toward one reader only, different readers can generate the same transaction ID * Case 3: for different services, different transaction IDs should be generated   One possible method is that the CN can assign the correlation ID based on the above three cases, i.e.,   * CN assigns different correlation IDs for case 1 and 3 * CN assigns same correlation ID for case 2.   The reader can generate the same transaction ID for the same correlation ID, while generate different transaction IDs for different correlation IDs. With this method, the coordination among readers is not needed. However, this issue is more likely to be RAN3 issue. | |
| HONOR | No | |  | |
| Sharp | No | | A common transaction ID is preferred. | |
| Futurewei | No | |  | |
| Ericsson | No | | We prefer that no coordination is required between the readers, i.e., transaction ID is generated with no reader specific aspects. It may be equal to the CN correlation ID or generated from the CN correlation ID, e.g., by truncation or using a specified function that would apply to all readers. | |

**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. |
| OPPO | By now the details of the pattern of the CN correlation ID is still being discussed in SA2, so we don’t know whether or not it is possible to ensure the uniqueness of the transaction ID in a certain time duration by making some processing of the CN correlation ID to generate the transaction. But one general principle should be follows:  A newly received CN correlation ID shall be mapped to a transaction ID that has not been used recently, i.e., leaving enough time gap for reusing the same transaction ID in order to let A-IOT devices recognize that latter A-IOT paging message shall correspond a new service request.  Regarding Lenovo’s opinion, we don’t know yet whether there will be a unique part of the correlation ID. Also, the transaction ID bitstream length is limited, we can not make sure that for different service, the transaction ID could be different. |
| Huawei, HiSilicon | See above. But, we don’t have to specify the reader behavior. |
| CMCC | This is closely related to the detailed design of CN correlation ID which has not been decided by SA2. From our perspective, **one simple way is to truncate the CN correlation ID to tansaction ID, if CN correlation ID is too long**. For example, the reader can use the last X bits as its transaction ID, where X denotes the size of transaction ID. |
| CATT | There is no RAN2 protocol impact on how the transaction ID is generated by the Reader. We could take partial of the correlation ID as the transaction ID, e.g., last 4 bits. Considering SA2 has not determined the definition of correlation ID, we may postpone this issue for a while. |
| Spreadtrum, UNISOC | See above. One simple way is to truncate correlation ID as transaction ID, e.g., take the lowest few bits of the correlation ID as transaction ID. |
| Apple | We think this is out of RAN2 scope. Please see our answer in Q11 |
| Xiaomi | This can be decided by RAN3. |
| ZTE | See our comments for Q11. Truncated correlation ID may be not feasible. |
| InterDigital | This is outside of RAN2 scope for now. |
| MediaTek | We don’t think this is exactly in RAN2 scope, but RAN2 may have requirements on the transaction ID as described in our previous answers, mainly that different readers should generate different transaction IDs even for the same service. If we need particular characteristics from the transaction ID, we should notify RAN3 and probably SA2. |
| Nokia | This should be an RAN3 decision |
| ETRI | This is outside the scope of RAN2, but this issue may affect device behavior, such as service differentiation, duplicated paging, and other related aspects. |
| Qualcomm | Depends on how SA2 defines correlation ID. Agree with MediaTek we may need to notify SA2/RAN3 about what RAN2 is looking for. |
| Samsung | This is RAN3 issue. One possible way is up to implementation. |
| HONOR | Out of the scope of RAN2. |
| Sharp | It could be a RAN3 issue. |
| Futurewei | Should be up to reader implementation. |
| Ericsson | We prefer that transaction ID is generated with no reader specific aspects. It may be equal to the CN correlation ID or generated from the CN correlation ID, e.g., by truncation or using a specified function that would apply to all readers.  This is not entirely up to RAN2 to decide, SA2 and RAN3 should be involved, however it would be beneficial if RAN2 can provide feedback regarding the design criteria considering that RAN2 has agreed to specify the subsequent paging mechanism. |

**Summary:** TBD

## Size of Transaction ID

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

1. 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. |
| OPPO | 3-4 bit is enough for the same-reader case. For multi-reader scenario, if reader ID is not supported, then more than 10 bits are required since for one overlapped coverage area, may be 3 to 4 reader’s A-IOT paging message could be heard. |
| Huawei, HiSilicon | 2 or 3 bits should be sufficient for indoor case. |
| CMCC | **No strong view, maybe 2 bits or 3 bits.** |
| CATT | See our comment on Q12. |
| Spreadtrum, UNISOC | No strong view, 2 or 3 bits may be enough. |
| Apple | Anything less than 4 bit is not reasonable and risky.  The device may miss a large chunk of messages from the same reader due to mobility and bad channel conditions. So, the device may misinterpret a wrap-up new transaction ID as an old transaction once the channel conditions improves. As type.1devic does not have its own timer-based expiry of transaction ID, so we need to prevent the device misunderstanding with some enhanced signaling design.  Also, transaction ID size depends on correlation ID size and how SA2/RAN3 handles multiple-reader case. We think this needs to be postponed the exact size issue. |
| Xiaomi | To our understanding, to avoid the case that a device misses paging due to charging, the repetition times should be longer enough. Longer ID size needs more resources for the transmission from the reader side and more power consumption for the reception and decoding from device side.  Therefore 4 bits should be sufficient. |
| Tejas Networks | As the number of services are limited (e,g. 8-32), 3-5 bits should be enough. |
| ZTE | Given that we have transaction ID that should at least convey service ID and also there is no explicit reader ID, also will according to the output on the scenario and baseline assumption discussion, we think it may need a sufficiently long transaction ID, maybe at least 4 bits?, to distinguish different services and/or different readers. |
| InterDigital | Should be large enough to support different services as well as different readers in an area, if we go with a transaction ID to support the multi-reader scenario. |
| MediaTek | We don’t think a poll of RAN2 is the right way to determine the size. We need information from RAN1 and maybe SA2.  As expressed at RAN2#129, we think the main issue is wraparound due to device unavailability, which should be a fairly short-lived condition. RAN2 can’t quantify “fairly short-lived” without RAN1 input; we tend to think 3 bits is safe and 2 may be adequate, but we should get RAN1 input on how long device unavailability may be expected to persist. SA2 could also be consulted about the supported level of service parallelism. |
| Nokia | We don’t think that RAN2 is the right group to determine this, but would expect not less than 6 bits (4 to prevent wraparound for pagings and 2 for readers in proximity) |
| ETRI | We share the same view as MediaTek. |
| Qualcomm | Similar view as OPPO, it depends on whether reader ID is separately included or not. In absence of reader ID, the transaction ID cannot be very small. Also agree with Apple’s comment. Further, as commented by MediaTek, it also depends on RAN1 and maybe SA2. |
| Samsung | This depends on the number services among neighboring readers of a specific reader. The service served by one reader should be uniquely identified among neighboring readers. To achieve such purpose, 2 or 3 bits could be enough for Rel19. |
| HONOR | Considering the signlling overhead, 2 or 3 bits is enough. The collision issue could be solved by the implement of reader or CN. |
| Sharp | 2 or 3 bits is preferred. |
| Futurewei | For single-reader deployment, 4 bits may be enough.  For multi-reader deployment, could consider Transaction ID much longer than 4 bits, e.g., 8 to 16 bits. |
| Ericsson | This is not entirely up to RAN2 to decide, SA2 and RAN3 should be involved, however it would be beneficial if RAN2 can provide feedback regarding the design criteria considering that RAN2 has agreed to specify the subsequent paging mechanism. The main aspects to consider here is to avoid wraparound and the maximum number of neighboring readers that one requires to cover an overlapping area. No strong preference, but it should at least be 3 bits. |

**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. | |
| OPPO | No | Our initial though was to make the ID visible to the MAC layer, which leaves more time for the MAC layer to determine the random-access resource index and the resource time-frequency location for Msg1 transmission. But such purpose could be also fulfilled by the implementation, e.g., the NAS layer sends a confirmation flag to the MAC layer to trigger it to do the msg1 transmission preparation work, as soon as the NAS layer confirms this device has been paged.  In addition, consider the possible security problem brought by making the ID visible to the MAC layer, and more signaling overhead may be incurred, we agree that such ID shall not be visible to the MAC layer. | |
| Huawei, HiSilicon | No | We don’t support any MAC layer sub-group paging solution. | |
| CMCC | Open | **We are open to Q14.** In SA3, temporary ID is agreed to protect A-IoT device ID. Thus, the paging identifier discussed in RAN2 is very likely to be a temporary ID. With the consideration that the detailed mechanism of the freshness, synchronization and re-synchronization of temporary IDs still have not been determined by SA3, RAN2 is asked to wait for further input from SA3. | |
| CATT | Yes with comment | We understand the temporary ID introduced by SA3 is mainly used to page UE over the AioT interface, similar with the 5G-S-TMSI in NR. We think the temporary ID is AS information and visible to MAC layer with the following reasons.  1) According to the TR 23700-13, from SA2’s perspective, the procedures/messages supported by AioT Device NAS layer do not include the inventory request, i.e., the paging message does not contain AioT Device NAS layer data.   |  | | --- | | 4. AioT Device NAS protocol is supported between the AioT Device and the AIOTF. The AioT Device NAS layer supports Inventory Response and Command (e.g. Read and Write) Request and Response. |   2) After the device receives a paging message containing the paging identifier, it is simpler for MAC layer in device to decode and determine whether the paging identifier matches with its own. | |
| Spreadtrum, UNISOC | No | In our opinion, Paging identifier is allocated by AioTF and should be handled in AioT NAS layer. There is no need to make it visible to MAC layer. | |
| Apple | Yes | I think the paging ID or (temp) device ID needs to be exposed to the reader’s MAC layer. This is because we have agreed to use AS ID in AS layer. If the reader does not know the exact (temp)device ID of the device, the reader cannot associate a follow-up unicast command procedure with the correct AS ID.  The paging ID does not need visible to AS layer of device. The device NAS layer can determine whether it is paged or not.  If RAN2 reverse the agreement and no longer supports AS ID, then we are fine to not expose paging ID in MAC layer. | |
| Xiaomi | See comments | The FFS was for “MAC layer sub-group paging solution.”, we do not think it is valid use case.  We are open if there are other valid use cases. | |
| ZTE | Yes | Similar views as CATT.  We see no any existing agreement to assume Paging identifier needs to be handled in AIoT NAS layer. Previously, there may have assumption that permanent ID have to be used as Paging identifier and it needs to be encrypted at the AIoT NAS layer. But according to the latest SA3 LS, such assumption no longer holds.  Companies cannot disagree on something without indicating reasons/justifications. For the motivation/justification to make paging identifier needs to be visible to the MAC layer, besides the ones mentioned by CATT, we add the following:   * In the case where a very large number of devices to be inventoried (this can be a typical/special service in AIoT which is not in general IoT system), the reader may need to perform sub-grouping (mask only a subset) according to its real resources situation. This is also the way to make this special service in AIoT truly feasible and to achieve the expected AIoT air interface efficiency and inventory latency. Shortly to say, we think specification should allow the possibility to do sub-grouping by reader and so the paging identifier needs to be visible to the MAC layer.   + If to pursue the minimal specification work in the MAC layer, at least the mask rules provided by the CN needs to be visible so that the reader can append RAN mask rules on top of them. For the reader, this operation can only be performed at the MAC layer. | |
| InterDigital | Yes, with comments | The online discussion distinguished two different understandings of “transparent”. Firstly, we think there is no need to allow the MAC layer to page based on subgroups, or modify the paging ID to select a subgroup so paging by IDs received by the reader from the CN only should be supported only.  That being said, there should be no reason why the MAC layer cannot see the paging ID (similar to Uu paging where the NAS ID is present in the paging message). We are also fine to wait for further SA2/SA3 inputs on temporary ID to resolve this. | |
| MediaTek | Question is ambiguous | What does “visible” mean here? We assume it will work something like the ID in Uu paging: The reader will be informed of an upper-layer ID for the device it is paging, but the ID means nothing to the reader, which blindly stuffs it into the paging message. Does this count as “visible”?  To design the “paging record” part of the paging message, RAN2 do need to know the size(s) of all possible paging identifier formats, and specifically whether we need to specify a variable-size ID field to accommodate different formats. (E.g., is there ever paging by permanent ID, such as when a re-sync of the temporary ID is needed or the device has never been paged before?) We assume SA2 will illuminate this issue. | |
| Nokia | Agree with CATT |  | |
| ETRI |  | We share the same view as MediaTek. |
| Qualcomm | Yes | First of all, there seems to be disconnect between what SA2 has concluded and what some companies in RAN2 are assuming. According to SA2 conclusions from TR 23.700-13 version 2.0.0 clause 8.1.1 (as also indicated by CATT and commented by ZTE above):  “4. AIoT Device NAS protocol is supported between the AIoT Device and the AIOTF. The AIoT Device NAS layer supports Inventory Response and Command (e.g. Read and Write) Request and Response.”  Furthermore, the draft SA2 specification captures the following (see <https://www.3gpp.org/ftp/tsg_sa/WG2_Arch/Latest_SA2_Specs/Latest_draft_S2_Specs/23xyz-020_AIoT.zip>)  “The AIoT NAS protocol supports the inventory response and command related signalling.”  **That is, inventory *request* is not included in the list of items supported by or carried using AIoT NAS.**  For the Reader MAC, the paging identifier/temp ID is beneficial to be visible to the reader for multiple reasons. Firstly, as Apple explained, when AS ID is to be used over the air, the reader needs to know the mapping of paging ID to AS ID at least for the inventory+command case where we assume the follow up command by CN will probably not use AS ID but rather use temp ID/device ID.  Secondly, the reader needs to at least know whether it is a dedicated or group paging. In absence of the information whether it is unique ID or filtering criteria, so that it can properly do resource allocation for D2R.  So, regardless of how it is done (e.g. providing this as separate assistance info or letting reader look into the inventory request message), the reader must be able to associate the paging/temp ID to the AS ID.  On the device side, the device’s MAC layer (or even the lower layer depending on how RAN1 designs the L1 control) needs to check the paging ID to determine that the paging is addressed for itself. Again, recall the SA2 conclusion, inventory request is not necessarily supported by AIoT NAS protocol. So, there is no point in always passing the whole packet to NAS layer. Furthermore, such solution comes with the limitation that every device must fully receive and decode the whole packet and send to NAS without doing any processing at the lower layers. | |
| NEC | No | If the purpose is to expose the number of devices to the reader, we think additional information from CN can achieve this, so paging identifier do not need to be visible to the MAC layer. | |
| Samsung | No | The “AS ID” is used to identify the device over A-IoT air interface. So, the visibility of the paging identifier is not needed at the MAC layer. | |
| HONOR | No strong view | The visible assumption is also related to the design of SA2. RAN2 could further discuss if there is some motivations to make it visible or not visible. | |
| Sharp | No | It is not necessary for the paging ID to be visible at the MAC layer, upper layer would be enough. | |
| Futurewei | Postpone | We also feel that the word “visible” does not accurately reflect what is being discussed here. The reader anyway needs to use a field to carry the paging ID. So, what can prevent the paging ID from being visible to the reader? Based on some suggestions in the discussion above, we felt the real question is whether the paging ID can be manipulated by the reader to form sub-groups in subsequent paging or R2D triggers. We think this question can be postponed until the D2R resource selection scheme for CBRA becomes clearer. | |
| Ericsson | No, please see the comments | It is not clear to us what is intended by “visible” here. If we assume that the question is whether it should be possible for the reader to, for example, mask/reformulate the identifier provided by the CN, we prefer that this should not be possible.  In that sense, we do not support MAC layer sub-group paging solutions. We assume that temporary ID is intended for paging a single device, hence there is no need for sub-grouping and it is not clear yet whether group paging via masking permanent ID would be possible since paging identifier may be encrypted. | |

**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