3GPP TSG-RAN WG2 Meeting #113 electronic R2-200xxxx

Online, January 25th – February 5th 2021

Source: CATT

Title: Summary of email discussion 154 - eDRX cycles

Agenda Item: 8.12.3

Document for: Discussion and Decision

# Introduction

This contribution provides a summary of the following email discussion:

* [Post112-e][154][REDCAP] eDRX cycles (CATT)

 Scope: Progress on eDRX cycles for Idle and Inactive

 Intended outcome: email discussion report

 Deadline: Long

Rapporteur would like to have the following schedule for this email discussion to have enough time for preparing the summary report.

* Phase 1 (2021-01-06): Companies are invited to provide inputs and comments to questions.
* Phase 2 (2021-01-12): Rapporteur will provide draft summary with proposals, companies are invited to provide comments to the summary proposals.

# Contact information

|  |  |
| --- | --- |
| Company | Name and email address |
| CATT | Pierre Bertrand; pierrebertrand@catt.cn |
| Apple | Naveen palle ; naveen\_palle@apple.com |
| Ericsson | Tuomas Tirronen ; tuomas.tirronen@ericsson.com |
| Sharp | LIU Lei ; lei.liu@cn.sharp-world.com |
| Qualcomm | Linhai He ; linhaihe@qti.qualcomm.com |
| OPPO | Haitao Li ; lihaitao@oppo.com |
| Xiaomi | Yanhua Li ; liyanhua1@xiaomi.com |
| Huawei | Odile Rollinger : odile.rollinger@huawei.com |
| Futurewei | yyang1@futurewei.com |
| Samsung | Sangbum Kim; sb07.kim@samsung.com |
| Nokia | Samuli Turtinen ; samuli.turtinen@nokia-bell-labs.com |
| LGE | HyunJung Choe : stella.choe@lge.com |
| ZTE | liu.jing30@zte.com.cn |
| MediaTek | Pradeep Jose ; pradeep[dot]jose[at]mediatek[dot]com |
| Facebook | Yee Sin Chan ; yeesinchan@fb.com |
| CMCC | Xiaoman Liu；liuxiaoman@chinamobile.com |
| Sequans | Noam Cayron ; noam.cayron@sequans.com |

# Discussion

As a follow-up of the offline #111 [1][2], the following agreements on eDRX for REDCAP UEs were achieved in RAN2#111-e:

Agreements:

1. RAN2 study eDRX mechanism for both RRC\_IDLE and RRC\_INACTIVE in this SI. ‎
2. For RRC\_INACTIVE, the DRX cycle is extended to 10.24s as baseline.

Agreements via email - from offline 111:

1. For RRC\_IDLE, the DRX cycle is at least extended to 10.24s. FFS on further extension ‎beyond 10.24s.
2. For RRC\_IDLE and/or RRC\_INACTIVE, if the NR DRX cycle range is extended beyond 10.24s, the LTE ‎eDRX mechanism beyond 10.24s (e.g., PTW, PH, etc.) is used as baseline when NR eDRX cycle is configured beyond 10.24s.

FFS:

1. For RRC\_IDLE and/or RRC\_INACTIVE, FFS on baseline mechanism when the configured NR eDRX cycle is less or equal to 10.24s

Then, the above FFSs were further progressed in RAN2#112-e where the following agreements were achieved on eDRX for Idle and Inactive:

1. For UE in RRC IDLE/INACTIVE and eDRX cycle is less than 10.24s, paging monitoring does not use PTW and PH, if any.
2. RAN2 will study whether lower values than 5.12s for eDRX cycle for RRC\_IDLE and RRC\_INACTIVE REDCAP UEs, e.g. 2.56s, can also be considered.
3. eDRX cycle extension in RRC\_IDLE beyond 10.24s for REDCAP UEs will be studied in this SI/WI. For UE in RRC IDLE and eDRX cycle is equal to 10.24s, among the solution options, we start from the assumption that paging monitoring does not use PTW and PH.
4. the eDRX cycle in RRC\_IDLE is extended up to 2621.44s for REDCAP UEs, as a baseline (longer value e.g. 10485.76s can also be considered)

This discussion will focus on the leftover issues from RAN2#112-e on eDRX for Idle and Inactive:

1. eDRX in Idle: next steps on the solutions:
	1. solutions for 10.24s
	2. issues associated with upper and lower bounds
2. eDRX in Inactive
	1. Support > 10.24s?
	2. Inactive-specific issues (concurrent RAN/CN paging), difference with LTE, …

In general the trend should focus more on contents to be captured in the TR: listing pros and cons of the different options, and providing recommendations among the different solutions.

# eDRX in idle

### Solution for 10.24s

In RAN2#111-e, it was agreed that eDRX cycles up to (and including) 10.24s would be supported in RRC\_IDLE and then a long discussion took place in RAN2#112-e on the need to support eDRX cycles *beyond* 10.24s. Finally a compromise was reached to recommend supporting eDRX cycles beyond 10.24s where the solution for eDRX cycle = 10.24s would not use PTW and PH.

Given this compromise took place at the last stage of the meeting, it can be useful to elaborate pros/cons of this approach versus other approaches for capturing in the TR. Rapporteur’s understanding is that the benefit of such approach is that it would allow a UE requesting eDRX cycle always ≤ 10.24s during UE-CN negotiation to not support PTW/PH features, while the drawback is a departure from the legacy LTE solution for which the solution for eDRX cycle = 10.24s involves PTW and PH.

Companies are invited to further provide their views on the pros/cons of the recommended solution for eDRX cycle = 10.24s in RAN2#112-e versus other solutions, e.g. LTE solution.

**Q1: Companies are invited to provide their views on the pros/cons of the solution recommended in RAN2#112-e for eDRX cycle = 10.24s for RRC\_IDLE versus other solutions, e.g. LTE solution.**

|  |  |
| --- | --- |
| Company | Comments |
| CATT | The benefit of not using PTW and PH for eDRX cycle = 10.24s is that a UE which, by implementation, would never express a preference for an eDRX cycle > 10.24s during the UE/CN negotiation phase would not need to implement the PTW/PH functionality without a need for an explicit capability signaling. On the other hand, for UEs supporting such functionality and aiming at requesting eDRX cycle > 10.24s, there should not be much implementation difference if PTW and PH are used or not for eDRX cycle = 10.24s. |
| Apple | We see that NR already has 10.24sec interval in C-DRX while is different from LTE, and NR UEs are already used to 10.24 sec DRX timing. So it shouldn’t be that different to use 10.24 eDRX without PTW/PH. On the other hand, implementing the LTE aspects for 10.24 with PTW/PH should also be straight-froward. We are ok to go with the majority. We do recognize that the CN part from LTE eDRX may need some changes for the RedCap anyway, |
| Ericsson | Agree with above views – additionally, for 10.24 s and RRC\_INACTIVE similar solution was adopted for LTE.  |
| Sharp | Agree with the observation on 10.24s in C-DRX from Apple. It is fine to confirm the assumption in last meeting, i.e. eDRX cycle = 10.24s would not use PTW and PH. |
| Qualcomm | We think not supporting PTW/PH for eDRX cycle of 10.24s is a good compromise, because it enables longer eDRX cycles needed by some RedCap UEs and yet allow other UEs that do not need long eDRX cycles (>10.24s) reuse NR R16 eDRX implementation without additional development work. Although the resulting behavior is slightly different from LTE, we do not expect it would have much impact. And this difference was made since early discussion in R16 power saving study, which was near unanimously agreed.  |
| OPPO | We agree that the main benefit of this solution is to allow a UE to request a eDRX cycle always ≤ 10.24s but not need to support PTW/PH, which could reduce UE implementation complexity. |
| Xiaomi | Theoretically speaking, PTW does not necessarily be introduced for with short eDRX, i.e., 5.12s or 10.24s as what we did for RRC\_INACTIVE state with short eDRX in R16 eMTC connected to 5GC. However, departuring from the legacy LTE solution would cause some trouble for 5GC, as the core network needs to identify the Redcap UE by subscription data. |
| Huawei, HiSilicon | Both are fine for us. Not using PTW/PH for DRX cycle of 10.24s would allow a single mechanism for UE that do not need longer DRX cycle. The behaviour will be slightly different from LTE but should not be a problem as this would align with RRC\_INACTIVE in eMTC. However, we will need to inform/ consult SA2/CT1. |
| Futurewei | We are fine with the compromise reached in RAN2#112-e. |
| Intel | Agree with others that for eMTC connected to 5GC and R16 NR C-DRX, DRX cycle 10.24s has been supported in R16 without using PTW/PH. The potential impact on core network should be limited. |
| Samsung | We prefer to follow the existing LTE solution. The main reason to introduce the PTW in LTE eDRX is to improve paging reception reliability, i.e. UE can have multiple opportunities to receive its paging during PTW.On the other hand, it seems a tiny optimization to use no PTW/PH in 10.24s. We have assumed that the additional complexity would be less with PTW/PH.  |
| Nokia | We are OK to confirm the agreement. |
| LGE | We don’t have strong view on this agreement.  |
| ZTE | We are fine with this solution, it allows UE not supporting eDRX >10.24s to not implement PTW/PH. As mentioned by other companies, since this is different from LTE eMTC, 5GC should be able to differentiate legacy eMTC and Redcap devices in order to take different actions. |
| MediaTek | We prefer to follow existing LTE solution for the same reasons as Samsung, i.e. to allow multiple opportunities to receive paging during a PTW.  |
| CMCC | We think it’s a good compromise which allows UEs do not need long eDRX cycle could reuse legacy approach. |
| Sequans | We agree with the compromise and agree with above comments |

### Issues associated with upper and lower bounds

Although it is a common understanding that final upper and lower bounds of eDRX cycles should be finalized during the normative phase, it is useful to capture in the TR some preliminary recommendations with associated justifications.

It was agreed in RAN2#112-e that 5.12s and 2621.44s should be considered as baseline lower and upper bounds respectively, while lower/larger values e.g. 2.56s and 10485.76s could also be considered. Although the reasoning for baseline values is to stick to LTE values, it is interesting to capture the pros/cons of supporting the lower/larger values.

For the lower bound, [9] claims that 2.56s is required to support the reception of emergency broadcast services (e.g. ETWS primary notification) within the required delay budget (of 4 seconds), which is not possible with 5.12s eDRX cycle lengths. And since REDCAP UEs include wearables, they should be able to receive emergency broadcast services. On the contrary, [6] thinks this is no different from LTE where it was decided that optimizations were not needed for these scenarios, because a UE supporting ETWS or CMAS is not expected to be configured with eDRX. And the similar argument is valid for REDCAP use-cases that require long eDRX cycles.

Therefore the issue of the eDRX lower bound seems primarily related to the need to receive emergency broadcast services.

**Q2: Do you agree that the lower bound of the eDRX cycle should be designed to allow REDCAP UEs to receive emergency broadcast services?**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Argument(s) |
| CATT | No | Current specification already supports configuring a UE with UE-specific DRX cycle = 2.56s, so if the network and UE want to support REDCAP UE to receive emergency broadcast services with short DRX cycle, the CN can configure a UE-specific DRX cycle with 2.56s instead of configuring an eDRX cycle with 2.56s. The CN can configure the UE without eDRX cycle, if shorter cycle less than 5.12s is preferred. So no matter whether redcap UE is allowed to receive emergency broadcast services, once shorter DRX cycle less than 5.12s is preferred, the current UE specific DRX can work. |
| Apple | Pls see our comments.  | UEs (including the RedCap ones) which need to receive emergency broadcast need to use 2.56sec DRX or lower, as anything above might not work. But whether eDRX cycle should allow this or not, is probably answered in a different way.There are two aspects to look at. The first one is the key difference between eDRX and DRX which is that the UEs with eDRX in LTE are not expected to follow the LTE RAN paging cycle if this RAN paging cycle is shorter. For RedCap UEs where power saving is very important for longer battery life and with bigger antenna impairments (especially with 1Rx/1Tx), since more effort (power) is needed to be able to receive the paging (by for eg., waking up early to read more than one burst of SSB to re-sync in time/freq), longer DRX cycle is an important way to save power. From this aspect, we would like to propose that **RedCap UE follow the DRX cycle that NAS has accepted, instead of following the RAN paging cycle which suits well for legacy NR devices.** This is the reason to propose to include 2.56sec in eDRX.If we agree in RAN2 that RedCap UEs follow their NAS DRX cycle independent of the RAN paging cycle, then we do not need to have a discussion on inclusion/exclusion of 2.56 as part of eDRX. The legacy DRX values are anyway part of RedCap option if certain RedCap UEs want to use these DRX values.The second aspect is the reception of emergency broadcast services, and as per SID, wearables are part RedCap and any devices that interact with the humans should not be precluded from receiving emergency broadcast. It can be argued that such wearables should not request eDRX cycles, and we tend to agree, and the above proposal (of including 2.56 DRX to be followed by RedCap UEs even if the RAN paging cycle is shorter) would be a good compromise. Meaning the wearable RedCap UEs which intend to receive emergency broadcast should not request for eDRX cycles > 2.56 sec. To summarize, we think the two below proposals would be worth pursing:**P1 :** **Atleast for the DRX 2.56 or larger, RedCap UE follow the DRX cycle that NAS has accepted, instead of following the RAN paging cycle which suits well for legacy NR devices.****P2: Atleast some of the RedCap UEs are not precluded from emergency broadcast reception.** |
| Ericsson | No | Question should not be about emergency broadcast services as in our understanding those would and should be supported by RedCap UEs regardless of eDRX or not. If the UE is configured with longer eDRX cycles, then it may not be always possible to receive such broadcast within a certain latency bound, but this is an aspect which should be understood when configuring a UE with eDRX, and is not a new issue. Thus, the question is about specific need for “2.56 s eDRX” cycle. We have similar view as CATT and additionally we don’t think monitoring for gNB configured default paging (and RAN paging) cycles should be circumvented in this way. The default paging and RAN paging should be in RAN control. The solutions would have impact at least on gNB configuration for SI acquisition and modification period and related procedures and so on.  |
| Sharp | No | Agree with the rapporteur. In addition, gNB can configure 2.56s default DRX cycle in NR. If it is configured then UE can wake up every 2.56s. If the shorter value is configured by gNB, the UE needs to wake up frequently. The gNB should have such kind of control. |
| Qualcomm | ~~No~~ Yes | ~~We have similar view as CATT.~~ We can support the lower bounder of 2.56s |
| OPPO | No | As stated above, a ETWS or CMAS capable UE is not expected to be configured with eDRX. So we see no need to introduce lower bound for eDRX cycle. |
| Xiaomi | No | Generally, if the UE wants to receive ETWS and CMAS, it shouldn’t be configured with eDRX. |
| Huawei, HiSilicon | Yes with comments | Wearables need to receive public warning system information in time. 2.56s eDRX period can meet this requirement and reduce power consumption for monitor paging when default DRX cycle is less than 2.56s at the same time. However, this will introduce a potential risk of UE missing SI change indicator. So, we should have a good solution for this issue before supporting 2.56s eDRX period.  |
| Intel | No | Agree with Rapporteur that the CN can configure UE specific DRX cycle 2.56s for RedCap UE instead of eDRX if the UE needs to receive PWS and also want to save power. . |
| Samsung | No | This topic was already discussed in LTE eDRX. UEs configured with eDRX need not satisfy the latency requirement on PWS reception.  |
| Nokia | No | If Redcap UE is to receive emergency broadcast, the DRX can be used instead of eDRX. |
| LGE | No | We think the network configuration can handle this. In our understanding, the delay budget requirement (4 sec) to receive emergency broadcast is not always mandated (it is ok for some UEs receiving emergency broadcast after 4sec). If the delay requirement should be supported by some REDCAP UEs, the network may configure the UEs with DRX cycle 2.56 sec or less. |
| ZTE | No | Similar view as CATT, eDRX should not be used if the UE wants to receive ETWS and CMAS.  |
| MediaTek | Yes – see comments | As pointed out by Apple, the key difference between eDRX and DRX which is that the UEs with eDRX are not expected to follow the RAN paging cycle if this RAN paging cycle is shorter. Given that an operator will have a mix of devices in their NW (smartphones and wearables), it is not reasonable to have the RAN paging cycle set to 2.56s. In this scenario, wearables can negotiate (by NAS signaling) an eDRX cycle that is more appropriate for its operation (e.g. 2.56s) while not impacting regular smartphone operation. This allows the operator to have a mix of smartphone and wearables in their network, with an appropriate paging cycle configured for each of them. |
| Facebook | Yes | We share the same point of view of MediaTek and Apple, the use cases/scenarios of eDRX and DRX are not exactly the same. Due to the compact formfactors, power saving/optimization is even more critical for wearables, while today wearables are expected to support lots of time sensitive applications. eDRX 2.56s provide a sweet spot for wearables to support the reception of emergency broadcast under the latency constraint while benefiting wearables power consumptions. |
| CMCC  | No | If a UE wants to receive ETWS or CMAS, DRX could be used instead of eDRX, and the NR DRX cycle supports 2.56s already. So, we don’t see the need to introduce lower bound for eDRX cycle. |
| Sequans | Yes | This could provide a sweet spot for REDCAP UEs, as e.g. MediaTek suggests, but we agree SI acquisition issues should be resolved first. |

We can also check if other reason/feature/usecase would require/justify supporting a lower value than 5.12s.

**Q3: Do you see any other reason/feature/usecase for supporting a lower value than 5.12s?**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Argument(s) |
| CATT | No |  |
| Apple | Pls see our comments in Q2. | Pls see our comments in Q2. |
| Ericsson | No | Assuming the question is about eDRX cycle lengths. For “normal” DRX there are use cases with shorter lengths. |
| Sharp | No | See the comments on Q2. |
| Qualcomm | No | - |
| OPPO | No | See our reply to Q2. |
| Xiaomi | No |  |
| Huawei, Hisilicon | No |  |
| Futurewei | No |  |
| Intel | No |  |
| Samsung | No |  |
| Nokia | No | for eDRX. |
| LGE | No |  |
| ZTE | No |  |
| MediaTek |  | See our comments in Q2 – i.e. this is needed to enable a mix of smartphones and wearables in the network, with an appropriate paging cycle configured for each of them. |
| Facebook | Yes  |  |
| CMCC | No |  |
| Sequans | Yes |  |

**Q4: Companies are invited to provide their views on the pros/cons of supporting larger values of eDRX cycles in RRC\_IDLE e.g. 10485.76s compared with the baseline value of 2621.44s.**

|  |  |
| --- | --- |
| Company | Comments |
| CATT | Pros: considering the upper limit of the HSFN (10bit) is 10485.76s, if such large value of eDRX cycle is supported in the spec, it can be flexible to support case for UE with more power saving requirement such as battery life up to several years in future. If there is no such case, the CN can configure the eDRX with other values and the advantage of supporting 10485.76s is future-proof and flexibility. |
| Apple | While we are not very strong on views, we think that longer RRC\_IDLE eDRX times help with the other class of RedCap devices that need very long battery life and so we are ok with 10485.76s. |
| Ericsson | If extended DRX cycles are specified, we do not see the reason why the configuration would be artificially limited especially as there does not seem to be any technical reason.The long RRC\_IDLE eDRX solution should be captured in the TR, and if the solution is recommended to be specified (this is our preference), the final details of configuration possibilities etc. can also be agreed in normative phase. Note that also for eDRX in RRC\_IDLE we should inform at least SA2/CT1 of RAN2 views and solution recommendations and we can further consult whether there would be any concerns from CN side when an LS is sent. |
| Sharp | It seems that there is no different spec impact from different upper bound value for eDRX cycle. It depends on the battery life requirement. |
| Qualcomm | We understand that there does not seem to be technical issues in supporting eDRX cycles beyond the baseline value of 2621.44s, but we don’t see strong RedCap use cases that require eDRX cycles beyond that either (RedCap is not LPWA). |
| OPPO | No strong view. Whether to support larger eDRX cycles depends on the requirement of battery lifetime as well as tolerable paging latency.  |
| Xiaomi | Our preference is 43.69mins (2621.44s). Based on simulation results we captured in the TP, the gain is saturated at around 40mins.Also, the choice of H-SFN range and I-eDRX cycle range should be future proof. Assuming a target max eDRX value of 44 minutes (8 bits H-SFN length), this will results in an H-SFN range of 4\*eDRX i.e. 10 bits H-SFN length. |
| Huawei, Hisilicon | Larger values may provide more flexibility to both UE and network and we do not see a technical justification for a restriction. However, same as for Q1, we need to inform/ consult SA2/CT1.  |
| Futurewei | 2621.44s seems to be sufficient for most cases today. On the other hand, it doesn’t hurt to extend the value to 10485.76s for future-proof. (However, we should consult with SA2/CT1 first.) |
| Intel | Similar view as Qualcomm. We do not see strong RedCap use cases considering RedCap is not LPWA.  |
| Samsung | We have a preference with at least 2621.44s. No strong opinion with 10485.76s. It would depend on use cases in real market. |
| Nokia | Agree with Ericsson. |
| LGE | We are fine to support eDRX cycle up to 10,285.76s because no technical impact is foreseen and it’s up to application characteristics. |
| ZTE | Similar view as Qualcomm. |
| MediaTek | Agree with Ericsson that we see no reason to artificially limit the eDRX range in RAN for no technical reason. The CN already supports eDRX values up to 10485.76 (see Negotiated extended DRX parameters in 24.501). |
| Facebook | Similar view as Ericsson. |
| CMCC | No strong view. It’s not a technical issue, but depends on whether there’s such RedCap cases having strong requirements on battery life and power saving. |
| Sequans | Don’t really see the scenario for the larger value, but agree there is no technical issue |

# eDRX in inactive

Whether to support eDRX cycle in RRC\_INACTIVE beyond 10.24s for REDCAP UEs was discussed in the email discussion #915 to RAN2#112-e [3], resulting in the following summary:

|  |
| --- |
| **Summary from email disc #915:**23 companies provided answers to this question.* Yes: 12/23
* No: 11/23

Proponents of extending the eDRX cycle in RRC\_INACTIVE beyond 10.24s use similar arguments as for RRC\_IDLE, i.e. the years-long battery life requirement with, in addition, the benefit of leveraging RRC\_INACTIVE to take advantage of the Small Data Transmission feature to be specified in Rel-17.Opponents also use similar arguments as for RRC\_IDLE while, in addition, pointing that:* REDCAP UEs needing long battery life can transition to RRC\_IDLE
* It has impact on NAS retransmission
* If the UE has two eDRX cycles larger than 10.24s (one for IDLE, the other for INACTIVE), we need to first discuss how UE behaves because there are two PTWs and PHs

Based on the above, it is clear that there are split views on this issue and no conclusion can be drawn from this outcome. It is suggested to further discuss this issue online. |

Then, during the GTW online discussion of RAN2#112-e, it was decided to focus on the eDRX in RRC\_IDLE and postpone the discussion for RRC\_INACTIVE. Thus although no progress was done on this issue, some arguments have been exposed during email discussion #915 and it can now be re-discussed considering:

* The progress made in RRC\_IDLE during RAN2#112-e e.g. could similar compromise for 10.24s be recommended?
* The issues brought up during email discussion #915 and in contributions, which were mainly:
	+ Impact on NAS retransmission
	+ Handling of two PTWs and PHs (one for IDLE, the other for INACTIVE) when the UE has two eDRX cycles larger than 10.24s
	+ Which node decides the eDRX cycle for RRC\_INACTIVE [7][8]?

However, it is Rapporteur’s view that we are still in the study phase so we should focus on what is beneficial and possible from RAN side. For possible SA/CT impact we can coordinate with the relevant groups, provide our input and ask for solutions and feasibility. Thus it should not be precluded that the TR recommends a solution conditional to SA/CT response on feasibility during the normalization phase.

Therefore we propose to progress this issue as follows:

1. Re-assess RAN2’s view on the need and motivation to support eDRX cycle in RRC\_INACTIVE beyond 10.24s for REDCAP UEs (independently of the above issues)
2. If the answer is “Yes”, analyse how to address the issues.

### On the need for eDRX cycle > 10.24s in inactive

From companies’ inputs to the email discussion#915 one argument in favour of supporting eDRX cycle > 10.24s also in RRC\_INACTIVE is that REDCAP UEs may actually spend significant time in RRC\_INACTIVE, considering the Small Data Transmission feature to be specified in Rel-17. On the contrary, opponents consider that REDCAP UEs needing long battery life can switch to RRC\_IDLE.

Companies are invited to express their views on the need, from RAN2’s perspective, to support eDRX cycle > 10.24s also in RRC\_INACTIVE, from performance perspective, and highlight the differences, if any, with LTE. Note again that NAS retransmission issue and further RAN issues are addressed in Section 2.2.2.

**Q5: Do you agree it is beneficial, from RAN perspective, to extend the eDRX cycle in RRC\_INACTIVE beyond 10.24s for REDCAP UEs?**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Argument(s) |
| CATT | Yes | We think a difference with LTE is the support of small data transmission in Inactive from R17 onwards, which can then change the network strategy to keep the UE in inactive more often than in LTE. For example a device sending short payload (e.g. measurements) every 30s could be kept in inactive rather than switching back and forth between idle and connected. As a result it is expected that supporting eDRX cycle > 10.24s in inactive will bring power saving benefits.  |
| Apple | Yes | While the support from the NW is needed, we see it as very beneficial to have >10.24 sec in RRC\_INACTIVE to effectively support the usage of SDT (small data transfer) where the RedCap UE can avoid going into connected mode where avoidable and hence save some power during the longer DRX cycle times -> the small data transfer might not be latency critical as well. While it requires the NW to store the UE context, the other aspects of RRC\_IDLE and RRC\_INACTIVE are pretty similar if we also align on the DRX aspects. We also think the NAS timer part can be solved with additional signaling. |
| Ericsson | Yes | We agree with CATT and Apple views on use cases. It is clear there would be a power consumption benefit, and UEs in RRC\_INACTIVE would additionally be able to benefit from SDT for e.g. use cases with periodic uplink data with periodicity > 10.24 s. Based on the results in R2-2009620 and in the Appendix of the TR, there is a clear power saving gain vs eDRX in RRC\_IDLE at least for eDRX cycles of 10.24 s – couple of minutes, where the UE in eDRX in RRC\_INACTIVE additionally benefits from less signaling. Based on these results, lifetime of several years would not be achievable in some cases (e.g. 1 minute IAT) if only RRC\_IDLE can be used, because of the signaling overhead.Signaling reduction is an additional benefit from network point of view – there is need for less RRC signaling.  |
| Sharp | Neutral | The use case mentioned by other companies is fine to us, however whether the NAS retransmission issue can be solved is uncertain. If the majority of companies support this, we are also fine. |
| Qualcomm | Neutral | From our perspectives, we do not see strong use cases for >10.24s. But we are fine with the extension if it is preferred by majority of companies AND similar compromise for RRC Idle (no PTW/PH for 10.24s) is adopted. |
| OPPO | Yes | We understand that the main obstacle to support eDRX cycle beyond 10.24s for RRC\_INACTIVE UEs comes from NAS layer. From RAN’s perspective, extending eDRX cycle can always benefit the UE’s power consumption. And the tradeoff between delay performance and UE power saving can be in the control of RAN, who should be responsible for configuring RAN eDRX on a per-UE basis. |
| Xiaomi | Yes | We see some benefit to support eDRX for RRC\_INACTIVE beyond 10.24s due to the reduced signalling load between gNB and UE. Also, the simulation results we captured in the TP shows eDRX for RRC\_INACTIVE can extend the UE battery lifetime.  |
| Huawei, HiSilicon | Yes | According to the power saving gain analysis for small data transmission (SDT), SDT can reduce the power consumption significantly compared to RRC\_CONNECTED data transmission. According to the simulation results from different companies, eDRX with cycle longer than 10.24s can provide significant power saving for downlink data monitoring. Combining both will obviously benefit UEs with uplink data periodicity higher than 10.24s. According to TS 22.104, some industrial wireless sensors need to transfer small packets while they are not very sensitive to DL traffic delay, but they have strict battery lifetime requirement (>5 years). Both SDT and eDRX with cycle longer than 10.24s are necessary for them to meet this requirement. Due to the fact SDT is supported only in RRC\_INACTIVE in Rel-17, it is necessary to support eDRX with cycle longer than 10.24s for RRC\_INACTIVE. |
| Futurewei | Yes | Agree with CATT and Ericsson. |
| Intel | Neutral | We do not see the strong need to support >10.24s for INACTIVE since the IDLE can be used if power consumption is more sensitive for the UE. For delay sensitive UE, if delay requirement is for both DL and UL, eDRX cannot work well even if the SDT is supported. On the other hand, continuing to keep the UE in INACTIVE may have other benefits as mentioned by other companies.  |
| Samsung | Yes | If not supported, INACTIVE UE cannot achieve extreme power saving gain with eDRX. We would like to prefer further flexibility in operation. |
| Nokia | Yes | We agree with Ericsson and would be fine with Qualcomm’s suggestion to adopt the same compromise with RRC\_IDLE. |
| LGE | Yes | Agree with other comments above.  |
| ZTE | Yes | We understand the only motivation of supporting eDRX >10.24s, is to save UE’s power when the UE requires periodical/frequent UL transmission, but DL data is less and insensitive to the delay. We are fine to support it, but regarding the maximum value of eDRX cycle for RRC\_INACTIVE, whether it can be the same as RRC\_IDLE needs further discussion.  |
| MediaTek | No | When compared to a UE with eDRX > 10.24s in Idle mode, the delta power savings that Inactive mode operation will introduce are negligible. On the other hand, the impact this will have on the NAS procedures are quite significant as the UE is in connected mode in NAS. We are therefore reluctant to extend eDRX cycle in Inactive mode. |
| Facebook | Yes | Agree with Apple and Ericsson |
| CMCC | Yes | Share similar view with CATT, with small data transmission studied in R17, UE could perform data transmission in RRC\_INACTIVE without transfer to RRC\_CONNECTED. With extend the eDRX cycles beyond 10.24s could further realize power saving for UE in RRC\_INACTIVE. |
| Sequans | Neutral | We do not see a strong need for >10.24, but would be OK if majority is interested and solution for =10.24 does not include PTW/PH |

### Addressing the impacts of eDRC cycle >10.24s in inactive

### Handling of two PTWs and PHs

[7] raises this issue and one possible solution is proposed in [8][10] and consists in configuring the same PTW and eDRX cycle for both RRC\_IDLE (CN paging) and RRC\_INACTIVE (RAN paging), as shown in below figure from [10].



Figure 1: Shared PTW for RAN and CN paging.

**Q6: If the Q5 answer is “yes”, do you agree considering a common PTW and eDRX cycle configuration for RRC\_IDLE and RRC\_INACTIVE as one of the possible solutions to consider during the WI phase?**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Comments |
| CATT | Yes | We think such simple solution should be the starting point to minimize the complexity, and the details can then be finalized during the WI phase. We do not see any justification of supporting separate PTW and eDRX cycles for idle and inactive.  |
| Apple  | Yes | Yes common framework is better. |
| Ericsson | Yes | Agree with CATT. |
| Sharp | See comments | The common PTW and eDRX cycle is the simplest way for the UE. If the RRC\_INACTIVE needs shorter eDRX cycle than RRC\_IDLE, the UE can be configured different eDRX cycles and common PTW for RRC\_IDLE and RRC\_INACTIVE, and the UE monitors the shortest of eDRX cycle for RRC\_IDLE and eDRX cycle for RRC\_INACTIVE.  |
| Qualcomm | Yes | If >10.24s for RRC Inactive is adopted, we agree a common framework between RRC Idle and RRC Inactive is desirable. |
| OPPO | No | Regarding eDRX cycle, we prefer to support separate eDRX cycle configuration for CN paging and RAN paging, which could provide more flexible configuration compared with a common eDRX cycle configuration.Regarding PTW, from UE power saving point of view, the PTWs for CN paging and RAN paging should be overlapped as much as possible, e.g. a common PTW length and offset can be used for both CN paging and RAN paging.  |
| Xiaomi | Yes | We think the details can be studied further. And we do agree that some common parameters can be shared for monitoring CN paging and RAN paging for a simple solution. |
| Huawei, HiSilicon | Yes with comments | This solution is very simple and efficient but it will limit the flexibility in the configuration of eDRX period in RRC\_INACTIVE. We can discuss other potential solutions in the WI phase.  |
| Futurewei | Yes | Agree that common PTW and eDRX cycle is simple and desirable. Details can be studied further. |
| Intel | Yes | Agree the common PTW and eDRX configuration is the simple solution.  |
| Samsung | Yes | It is beneficial to have common configuration in power saving aspect. |
| Nokia | Yes | Common PTW seems beneficial to consider. |
| LGE | Yes | A common solution for RRC\_IDLE and RRC\_INACTIVE can be baseline for further discussion. |
| ZTE | See comments | Different from RRC\_IDLE, for RRC\_INACTIVE, when DL data arrives, CN will deliver the DL data to RAN node directly, and let RAN node to page UE. So far it is unclear whether CN can buffer the DL data until PTW arrives when UE is in RRC\_INACTIVE state, or RAN node is responsible for buffering those data until PTW arrives. In our understanding, it is too early to decide whether to use common PTW and eDRX cycle for both IDLE and INACTIVE. We agree it seems beneficial to have overlapped/common PTW windows, but we also prefer to have the possibility of configuring different eDRX cycles for IDLE and INACTIVE (the one for INACTIVE can be shorter). So we suggest not to make decision right now, and to discuss such details in WI phase.  |
| CMCC | Yes | Agree with CATT |
| Sequans | Yes | Agree this should be baseline, but enhancements can be considered, e.g. periodicities that are multiples of one another instead of equal. |

### Which node is responsible for configuring the eDRX cycle in inactive?

This issue is discussed in [7][8]. In LTE the eDRX parameters are configured by MME to UE via NAS which is transparent to RAN, but in NR, the CN and RAN both can trigger the paging according to the eDRX parameters, so both CN and RAN should be aware of the eDRX configuration for inactive. Irrespective of the details on how RAN and CN exchange this information, both [7][8] conclude that two options should be considered for the deciding node for the eDRX configuration for inactive:

* Option 1: CN decides the eDRX parameters for RRC\_INACTIVE
* Option 2: RAN decides the eDRX parameters for RRC\_INACTIVE

Rapporteur suggests to agree on these two options and to perform the down-selection during the WI phase.

**Q7: If the Q5 answer is “yes”, do you agree considering the above two options regarding which node decides the eDRX parameters for RRC\_INACTIVE. Companies are also welcome to express their preference, if any.**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Comments |
| CATT | Yes | Clearly both options are obvious candidates and could be captured in the TR. We would prefer that CN decides the eDRX parameters for inactive when > 10.24s, especially if they are common for idle and inactive, as proposed in Q6. Thus CN, which has better insight on UE traffic profile, would decide on the common PTW and eDRX cycle for both idle and connected.  |
| Apple | Yes | Opt-1 can help address the NAS re-tranmission timer issue. |
| Ericsson | Yes | Both options should be captured in TR with pros/cons. We currently think Option 1 should be the way and in any case CN is responsible for eDRX in RRC\_IDLE (and UE needs to monitor for CN paging also in RRC\_INACTIVE).We can work out the details in WI phase also after sending LS to SA2/CT1 as this requires coordination with other groups. |
| Sharp | Yes | Based on our comments on Q6, Opt-2 is slightly preferred. |
| Qualcomm | Yes | We prefer Opt-1 |
| OPPO | Yes  |  |
| Xiaomi | Yes | Both options can be considered. It worth to notice that in in R16 eMTC connected to 5GC , during SA2 study phase, both options has been captured as solution 38, alternative 2/3 specified in TR 23.724“Alternative 2: The UE requests different extended DRX parameters in the registration request to be used when in UE is in CM-IDLE and CM-CONNECTED with RRC\_INACTIVE. The AMF may reject or accept the UE request for enabling extended DRX for RRC inactive. If the AMF accepts the use of extended DRX for RRC inactive, the AMF indicates the enabling of extended DRX for inactive to the UE and passes the requested extended RRC inactive DRX parameters to RAN. Then the RAN decides and configures DRX parameters for RRC inactive as specified in TS 38.300 [19]. Alternative 3: The UE requests idle mode eDRX parameters from the AMF. The AMF passes the UE's Accepted idle mode eDRX parameters to the RAN. If the UE supports eDRX in RRC inactive, the RAN configures the UE with an eDRX cycle in RRC-INACTIVE up to the value for the UE's idle mode eDRX cycle as provided by the AMF or up to the maximum value allowed based on the NAS (and SMS) retransmission timers (whichever is lower). The RAN buffers DL packets up to the duration of the eDRX cycle chosen by RAN.” |
| Huawei, HiSilicon | Yes  | We have a preference for Option 2 which provides more flexibility to the RAN node in the configuration of the eDRX parameters.We can discuss further in the WI phase and consult with SA2/CT1.  |
| Futurewei | Yes | Both options can be considered.  |
| Intel | yes | Both options can work. But it is related to the coordination between RAN and CN, and therefore should be decided by SA2 and RAN3.  |
| Samsung | Yes | Both options are valid.  |
| Nokia | Yes |  |
| LGE | Yes | Slightly prefer option-1 but both options can be considered.  |
| ZTE | Yes | Both options can be captured in TR, down selection can be made during WI phase.  |
| CMCC | Yes | Both options can be considered. |
| Sequans | Yes |  |

### Case of eDRX cycle = 10.24s

As discussed in Section 2.1.1, a compromise was reached to recommend supporting eDRX cycles beyond 10.24s in RRC\_IDLE where the solution for eDRX cycle = 10.24s would not use PTW and PH. For the case where eDRX > 10.24s would also be supported in RRC\_INACTIVE (Q5 answer is “yes”), Rapporteur believes it is worth considering such compromise as well.

**Proposal: For UE in RRC INACTIVE and eDRX cycle is equal to 10.24s, among the solution options, we start from the assumption that paging monitoring does not use PTW and PH.**

**Q8: If the Q5 answer is “yes”, do you agree considering the above proposal, similar to RRC\_IDLE?**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Comments |
| CATT | Yes | We would indeed prefer to have a common design for both idle and connected regarding eDRX cycle = 10.24s, which is obviously simpler. |
| Apple | Neutral | We agree to CATT view and are ok to go with majority. |
| Ericsson | Yes | Agree with CATT, also this is the solution which was adopted for LTE-M connected to 5GC.  |
| Sharp | Yes | Agree with rapporteur. |
| Qualcomm | Yes | We prefer to have a common design for RRC Idle and RRC Inactive |
| OPPO | Yes | We prefer a unified solution for RRC IDLE and RRC INACTIVE. |
| Xiaomi | Yes | We can follow the LTE way. |
| Huawei, HiSilicon | Yes | Similar to RRC\_IDLE |
| Futurewei | Yes | A common design for RRC\_IDLE and RRC\_INACTIVE is desirable. |
| Intel | Yes | Agree with others.  |
| Samsung | No | We prefer to use PTW/PH as in LTE. |
| Nokia | Yes |  |
| LGE | Yes |  |
| ZTE | Yes |  |
| MediaTek | Neutral |  |
| CMCC | Yes | We prefer to have a common design for RRC\_IDLE and RRC\_INACTIVE. |
| Sequans | Yes |  |

### NAS retransmission issue

For eMTC UEs connected to 5GC, eDRX cycles in RRC\_INACTIVE are already supported up to 10.24 sec. One reason for not extending this value further is given in [4] (same argument also used in [5]):

“*For UE in CM-CONNECTED mode with RRC\_INACTIVE, the impacts of eDRX on CN should be considered. The value of eDRX period has impact on NAS signalling transmission in CM-CONNECTED. As specified in 5GS for Rel-15, the smallest NAS retransmission timer is 6s and the maximum retransmission times is 4. To avoid the failure of the procedure, the response from UE in eDRX should be given within 30s after initial transmission. Considering all potential factors, the longest eDRX period without impacting 5GC is set to 10.24s for RRC\_INACTIVE eMTC UE in Rel-16*”.

If the answer to Q5 is “Yes” the likely most secure way to assess the impact on the NAS timers is to ask CT1/SA2 (+ RAN3?) on the feasibility of this solution.

**Q9: If the Q5 answer is “yes”, would you agree sending an LS to (at least) CT1/SA2 informing them about RAN2’s preference and asking about feasibility?**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Comments |
| CATT | Yes | We should inform CT1/SA2 about RAN2’s preference soon enough to let them study the impact and potential workaround, if needed. |
| Apple | Yes | We like the line of the LS as indicating RAN2 preference and asking for any in-feasibility. |
| Ericsson | Yes | Yes, eDRX is a feature with CN impacts thus we need to send LS in any case to CT1 and SA2. We don’t think this should be referred to as an issue in RAN as this has been discussed in SA2 where solutions are already available (see e.g. TR 23.724 soln 24). The detailed discussion is not in RAN2 scope, but if CN is aware the UE is not reachable (because the UE has been configured with longer eDRX) it can manage the NAS procedures properly. (Also note the mentioned timers have actually been extended for LTE-M/NB-IoT case already, but this is not referred to above or in all discussion). |
| Sharp | Yes | The eDRX no matter IDLE or INACTIVE impacts CT1/SA2. |
| Qualcomm | Yes |  |
| OPPO | Yes | Maybe it’s beneficial to have some RAN2 agreements before we decide to send LS to CT1/SA2 |
| Xiaomi | Yes |  |
| Huawei, HiSilicon | Yes | We agree with the Rapporteur’s view that we are still in the study phase so we should focus on what is beneficial and possible from RAN side. For possible SA/CT impact we can coordinate with the relevant groups, provide our input and ask for solutions and feasibility. So, we agree to send LS to CT1/SA2. |
| Futurewei | Yes | We should inform CT1/SA2. |
| Intel | Yes | This should be decided by SA2 and CT1 especially considering their concern in previous LS.  |
| Samsung | Yes | It’s needed. |
| Nokia | Yes |  |
| LGE | Yes |  |
| ZTE | Yes | Besides NAS retransmission timer, maybe we can also consult them about other aspects in the LS, e.g. maximum value of eDRX for INACTIVE, the preferred solution for Q6/7? |
| MediaTek | Yes – see comments | We should not form a RAN2 preference before understanding if the option to extend Inactive mode eDRX is reasonably feasible from SA2/CT1 |
| CMCC | Yes |  |
| Sequans | Yes |  |

# Other

**Q10: Any other relevant issue to discuss?**

|  |  |
| --- | --- |
| Company | Issue description  |
|  |  |
|  |  |
|  |  |

# Conclusion

# Reference

1. R2-2008193 Summary of offline 111 - DRX aspects; CATT
2. R2-2008216 Summary of offline 111 - DRX aspects - second round; CATT
3. R2-2009364 Summary of email discussion 915 - UE power saving features; CATT
4. R2-2007346 Discussion on eDRX for RRC\_INACTIVE and RRC\_IDLE; Huawei, HiSilicon
5. R2-2006748 Use cases target to extend paging DRX cycle and relax measurements for stationary devices; Intel Corporation
6. R2-2009116 Further considerations for eDRX; MediaTek Inc.
7. R2-2009247 Discussion on eDRX for Redcap UE; ZTE Corporation, Sanechips
8. R2-2009363 On eDRX for NR RRC Inactive and Idle; CATT
9. R2-2009532 Support of 2.56 eDRX cycle and emergency broadcast reception for RedCap UEs; Apple, Facebook
10. R2-2009620 RedCap power saving enhancements; Ericsson