**3GPP TSG-RAN WG2 Meeting #119-bis electronic R2-22xxxxx**

**e-Meeting, 10th October – 19th October 2022**

Source: vivo

Title: Summary of [Offline-417][POS] Calculation of TIR and provision of AL to UE

Agenda Item: 6.11.2.3

Document for: Discussion and Decision

# Introduction

This document summarizes the following email discussion:

* [AT119bis-e][417][POS] Calculation of TIR and provision of AL to UE (vivo)

Scope: Discuss the proposal from R2-2210606 and conclude on a way forward.

Intended outcome: Report to CB session

Deadline: Friday 2022-10-14 1000 UTC

## 1.1 Contact Points

|  |  |  |
| --- | --- | --- |
| Name | Company | Email Address |
| Xiang Pan | vivo | panxiang@vivo.com |
| Grant Hausler | Swift Navigation | grant@swiftnav.com |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Background



**Figure 1: Relation of PL and TIR**

Figure 1 illustrates the relation of PL, TIR and error distribution. With the feared events, UE can generate the distribution of the position error based on implementation. One PL can be derived with a specific TIR for a certain error distribution, and vice versa.

In the LPP *ProvideLocationInformation* message, the achievableTargetIntegrityRisk-r17 is presented optionally along with PL.

|  |
| --- |
| IntegrityInfo-r17 ::= SEQUENCE {  horizontalProtectionLevel-r17 INTEGER (0..50000),  verticalProtectionLevel-r17 INTEGER (0..50000) OPTIONAL,  achievableTargetIntegrityRisk-r17 INTEGER (10..90) OPTIONAL,  ...  } |

|  |
| --- |
| ***integrityInfo***  This field provides the integrity result for the *locationEstimate.*  - ***horizontalProtectionLevel*** provides the HPL for the *locationEstimate* along the semi-major axis of the error ellipse. Scale factor 0.01 metre; range 0 – 500 metres.  - ***verticalProtectionLevel*** provides the VPL for the *locationEstimate*. Scale factor 0.01 metre; range 0 – 500 metres.  - ***achievableTargetIntegrityRisk*** indicates the achievable TIR for which the HPL and VPL are provided. The achievable TIR is given by *P*=10-0.1n [hour-1] where *n* is the value of *achievableTargetIntegrityRisk* and the range is 10-1 to 10-9 per hour. If this field is absent, the achievable TIR is the same as the *targetIntegrityRisk* in *CommonIEsRequestLocationInformation*. |

However, the definition/purpose of *achievableTargetIntegrityRisk* is not specified in the current spec. In the summary [2], it is assumed that the *achievableTargetIntegrityRisk* can be used for the following scenario:

- The UE may calculate the PL for the requested TIR (as normal). If the AL from the application is available at the UE, the UE could check whether the determined PL satisfies the AL requirement. If not, the UE may adjust the TIR in such a way that the determined PL satisfies the AL, and then reports the PL together with the 'new' TIR (achievableTargetIntegrityRisk). For example, the UE calculates a PL for the requested TIR of 10E-7. The UE then determines that this PL does not satisfy the required AL. The UE may then adjust the TIR until the computed PL satisfies the AL (e.g., a TIR of 10E-4 may satisfy the AL requirement). The UE then reports the PL with the achievableTargetIntegrityRisk of 10E-4.

The contribution [1] shares a similar view that *achievableTargetIntegrityRisk* is offered as a substitution, for the current situation cannot satisfy the client-required integrity risk, which is expected to be larger than TIR in terms of value. Knowledge of AL could offer UE a tuning boundary, with which the achievable TIR can be generated and then provided to the location server and LCS client for reference. Without AL, the UE may only feedback one random combination of PL and *achievableTargetIntegrityRisk*, which may be meaningless if both PL and TIR exceed the required bound.

The following Observations and Proposals are made based on the above understanding in [1]:

**Observation 1: The value of PL is obtained by the knowledge of TIR and the error probability distribution modeled by UE implementation.**

**Observation 2: UE would not intend to tune its implementation for other KPIs without the prior knowledge about the availability of positioning system (the relationship of AL and PL in terms of value).**

**Proposal: Alert Limit (AL) should be provided to UE in GNSS positioning integrity, in order to optionally obtain the achievable TIR.**

# Discussion

As described in the background, the purpose of *achievableTargetIntegrityRisk* is not clear in the current specification. To the understanding of [1], the *achievableTargetIntegrityRisk* is offered as a substitution when the available integrity risk cannot satisfy the client-required target integrity risk. The moderator thinks it may be due to limited UE capability and/or specific positioning error distribution.

**Question 1: Do companies agree with the above purpose of Achievable Target Integrity Risk?**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Comments |
| vivo | Yes |  |
| Swift Navigation | No with comments | The difficulty is that there is no general algorithm to compute a PL for an arbitrary TIR. Any specific implementation may have only one value of TIR for which it was designed and valid (or at best a small number of discrete values of TIR).  This is the primary purpose of the *achievableTIR*, to allow the UE to return a PL corresponding to its design value of TIR instead of the requested TIR, if they are not equal.  The proposed scenario in [2] is also another possible use case. If the UE had a very general implementation that could generate a PL for any TIR (or vice versa) then the *achievableTIR* could be used together with knowledge of the AL to improve system availability, however we are not aware of any implementations that allow TIR and PL to be freely traded off in this way. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

As to how the UE compute the achievable TIR, one company figured out that the AL is not needed and it is also possible that the UE reports PL>AL. In this case, the reported AL and *achievableTargetIntegrityRisk* seem to be a random combination. Note that TIR shall be sent to the UE to derive PL, the moderator thinks it’s straightforward to provide the AL to UE to derive the *achievableTargetIntegrityRisk*.

**Question 2: As to how the UE compute the *achievableTargetIntegrityRisk*, which option do you prefer?**

* **Option 1: based on AL, e.g., PL is set equal to AL.**
* **Option 2: without AL, the combination of PL and achievable TIR is set up to UE implementation.**
* **Option 3: Others, please specify.**

|  |  |  |
| --- | --- | --- |
| Company | Option | Comments |
| vivo | Option 1 |  |
| Swift Navigation | Option 2, with comments | There are two options depending on the UE implementation:   1. The UE is able to calculate the PL corresponding to any requested TIR. In this case the *achievableTIR* should not be used and the PL should be computed according to the requested TIR. **Swift is not aware of any algorithm or implementation that can achieve this in practice.** 2. The UE is only capable of calculating the PL corresponding to one (or a small number of) discrete values of TIR. Then if the requested TIR is not one of the supported values, the UE should respond with a PL and *achievableTIR* corresponding to the only/best/closest supported TIR. **This was the intended use case for *achievableTIR* and all integrity algorithms that we are aware of fall into this category.**   We think this is in line with the current field description for *achievableTargetIntegrityRisk* highlighted above.  In response to the comment from QC in the summary [2], we do agree that this is counter to the intention for the LMF to set the TIR, however we do not see any other mechanism for the UE to report to the LMF in advance what TIR(s) are supported and therefore at least the UE can respond on a “best effort” basis what it is able to achieve. It is then at the discretion of the LMF how to use the returned information.  To respond to Vivo’s proposal, we agree that if a UE implementation has sufficient flexibility to use a provided AL to set the PL and hence the *achievableTIR*, then it is free to do so and this may be a reasonable choice, but it should not be a constraint of the system and should be left up to implementation as most or all UE implementations are anticipated to be significantly more constrained in what TIR value or values they can support. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

For the definition of Achievable Target Integrity Risk, one initial version is formulated as follows:

**Achievable Target Integrity Risk**: A integrity risk that can be achieved with a specific protection level [(e.g., equal to AL)]. This parameter should be explicitly indicated when the required target integrity risk cannot be satisfied.

Note: whether the phrase [(e.g., equal to AL)] is needed relies on the conclusion of Q2.

**Question 3: Do companies agree to take the above definition of Achievable TIR as a baseline and capture it in stage 2 specification when available?**

|  |  |  |
| --- | --- | --- |
| Company | Yes/ No/  Yes with changes | Comments |
| vivo | Yes |  |
| Swift Navigation | Alternative suggestion | Remove “e.g., equal to AL”  Suggestion:  “If the requested Target Integrity Risk cannot be satisfied then the Achievable Target Integrity Risk should be explicitly indicated and should contain the integrity risk to which the computed HPL and VPL correspond. If the requested Target Integrity Risk is not supported, then the choice of what Achievable Target Integrity Risk to return is up the implementation.” |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

During the online session, some companies thought that providing AL to UE to compute the achievable TIR was a new functionality and could be discussed in Rel-18. However, the *achievableTargetIntegrityRisk* was introduced in Rel-17 and the definition or how it works is not clear. So the moderator tends to clarify it as Rel-17 CR.

**Question 4: if the AL shall be provided to UE to compute the achievable TIR, which option do you prefer:**

* **Option 1: essential CR in Rel-17 to make how it works clear.**
* **Option 2: new functionality to be discussed in Rel-18.**
* **Option 3: Others, please specify.**

|  |  |  |
| --- | --- | --- |
| Company | Option | Comments |
| vivo | Option 1 |  |
| Swift Navigation | Option 2 | We do think there are several useful reasons to provide the AL to the UE in the request (e.g. for allowing the UE flexibility to distinguish Misleading Information that is not Hazardous Misleading Information, which can improve system availability).  Swift has previously advocated for including the AL in the request, and we would advocate for revisiting this topic.  However, we do not see the AL as necessary to compute the achievableTIR nor to correct/complete the Rel-17 functionality. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Besides, [2] also indicated that a new UE capability is required. After UE indicates the capability, the presence of AL in the CommonIEsRequestLocationInformation could be interpreted as the UE being requested to provide an achievableTargetIntegrityRisk if the requested TIR cannot be satisfied.

**Question 5: Do companies agree that a new UE capability is needed to enable the UE to feedback an *achievableTargetIntegrityRisk* based on the provided AL?**

|  |  |  |
| --- | --- | --- |
| Company | Yes/No | Comments |
| vivo | Yes |  |
| Swift Navigation | No | The achievableTIR is an optional field with clearly defined criteria for when it is present, not a capability of the UE. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Conclusion

To be populated.

# Reference

1. R2-2210606 Discussion on the provision of AL for achievable TIR calculation
2. R2-2210784 Summary of AI 6.11.2.3: LPP corrections