Hey Nicola,

First, let me explain the problem, for those who may be curious about the main problem of the bug. In the second section, I will try to evaluate the suggestion that you have come up with.

Suppose that the m_vtA, ackSn and m_vtS values are equal to 1019, 1022 and 3, respectively (this is the case actually I was getting error, where m_vtA value is bigger than m_vtS. Since the sequence number of RLC layer of 10 bits (modulo 1024), that kind of case is much likely to being seen). At line 965 of lte-rlc-am.cc source file (ns-3.17), we see a while statement as follows.

```
while (m_vtA < ackSn && m_vtA < m_vtS)
```

Hence, the equivalent of that statement under the values supposed above is;

```
while (1019-0%1024 < 1022-0%1024 && 1019-0%1024 < 3-0%1024)
 = while (1019 < 1022 && 1019 < 3)
```

where, 0 is the default modulus base value of the class of SequenceNumber10;

Simply because the while statement returns false, it never terminates inside the loop although it was supposed to.

However, if we set the modulus base values of all the state variables, before while, to m_vtA then;

```
while (1019-1019%1024 < 1022-1019%1024 && 1019-1019%1024 < 3-1019%1024)
=while (0 < 3 && 0 < 8)
```

and the while statement is terminated 3 times in the following cases;

```
while (0 < 3 && 0 < 8)
while (1 < 3 && 1 < 8)
while (2 < 3 && 2 < 8)
```

Regarding with your doubt, basically it does not change anything but terminates in a same way.
  According to my suggestion, where modulus base value is only set once, the while statement is going to return true and terminated 3 times as above, whereas your suggestion generates the while statements in a slightly different way as follows;

```
while (0 < 3 && 0 < 8)
while (0 < 2 && 0 < 7)
while (0 < 1 && 0 < 6)
```

So yeah, as I said, this way is also welcome.
Hope, I could explain it clearly.
Best.

```
while (m_vtA < ackSn && m_vtA < m_vtS)
```