

RTCA Special Committee 186, Working Group 5

ADS-B UAT MOPS

Meeting #5

**UTC Time Mark Stability and
Other Range Validation Specification Issues**

**Prepared by Ian Levitt
Titan Corp., FAA Technical Center
In Response to Action Items 3-21 and 4-13**

SUMMARY

There is, as we well know, no way to completely eliminate the possibility of someone spoofing UAT avionics. We can only make it as difficult as possible for a spoofer to successfully spoof without being identified. Range validation is a promising technique to do this, but it requires a good amount of forethought in the UAT design.

There is, as we well know, no way to completely eliminate the possibility of someone spoofing UAT avionics. We can only make it as difficult as possible for a spoofer to successfully spoof without being identified. Range validation is a promising technique to do this, but it requires a good amount of forethought in the UAT design.

A critical element to the success of range validation is synchronization of all UAT avionics to a common clock. At the April Salem meeting, it was determined that we cannot specify the synchronization of the UAT clock to UTC to be any better than $-700/+2000 \mu\text{s}$. In terms of range validation, this is a very loose requirement, allowing a worst-case error between GPS-derived distance and Flight of Signal (FOS)-derived distance to be $(2700\mu\text{s} * 0.1617917 \text{ NM}/\mu\text{s}) \sim 437 \text{ NM}$. This is an enormous error, but may be acceptable if it remains constant. That is, if both UAT system clocks involved in the ranging are consistently offset from UTC (in the range $[-700, 2000] \mu\text{s}$) by the same amount, the error calculated for a particular target pair should be about the same from measurement to measurement, and could be averaged out. For example, if my UAT avionics is synchronized to UTC + 1000 μs , it must remain synchronized to UTC + 1000 μs within some tolerance for some sufficiently long duration. What the tolerance is must be determined, taking into account cost, feasibility, and desired effectiveness of range validation against spoofing. The following table shows the accuracy of the measurement (after bias) versus the allowable deviation from synchronization:

+/- μs	NM
0.5	0.161791689
1	0.323583378
1.5	0.485375067
2	0.647166757
2.5	0.808958446
3	0.970750135
3.5	1.132541824
4	1.294333513
4.5	1.456125202
5	1.617916892
5.5	1.779708581
6	1.94150027
6.5	2.103291959
7	2.265083648
7.5	2.426875337
8	2.588667026
8.5	2.750458716
9	2.912250405
9.5	3.074042094
10	3.235833783
10.5	3.397625472
11	3.559417161
11.5	3.721208851
12	3.88300054
12.5	4.044792229
13	4.206583918
13.5	4.368375607
14	4.530167296
14.5	4.691958985
15	4.853750675
15.5	5.015542364

I don't think that we want to introduce more than 5 NM of uncertainty. The left column is the magnitude of the deviation in microseconds that would result in a deviation in the right column. From an installation standpoint, I don't know what is possible to specify here, but I would suggest that we make it as tight a specification as is available. The larger the deviation, the easier it is to spoof. Depending on the physics of this problem, we can either specify a hard limit on how much the 1Hz UTC time mark can jitter around its synchronization, or if it should be specified as a standard deviation in the normal distribution.

Allowing the UAT synchronization to be in the very large range that has been specified opens up some pathological cases for successful spoofing. For instance, if one were to place a spoofing device in line with a runway (so that landing aircraft are flying directly toward it), any phantom target that he broadcasts is on the runway will look like a genuine target to an aircraft on approach or take-off, because any distance between the spoofing device and the phantom position will show up in range calculations as part of the allowable bias. The greater the allowable bias, the more distance the spoofer can put between himself and the scene of the crime. In the case of the current specifications, he can safely put a couple of hundred nautical miles between himself and the runway. Of course, we should not concern ourselves too much about particular situations. I mention this to illustrate that, without providing some mechanism for compensating for the latency in the UTC time mark and getting a better synchronization before any ranging calculations are made, there is a rather large window that could be taken advantage of by a studious spoofer. Perhaps, as suggested by Stan Jones, there could be some startup process where the avionics uses a nearby ground station (presumably closely coupled to UTC) to determine the offset of its own UTC time mark from actual UTC, and use that to calibrate its clock. Perhaps this bias is completely installation dependent and can be measured upon installation and entered into the avionics as a constant by the installers.

Yet another level of complexity, which we may not want to get into but I mention it for posterity, lies in the drift. It is conceivable that over the course of a flight, the mean synchronization time ($UTC + X \mu s$) may drift slowly, e.g. because of changes in temperature. This could present a sticky problem since a drift in the mean error is the most obvious characteristic of a spoofer. It is impossible to say what the lower limit should be to distinguish between spoofing and a natural drift because in the infinite geometries of spoofer/victim, there can be any amount of drift. Perhaps this is better left up to the application without our making any specifications on this third-order effect (so why do I mention it? good question.)

The final requirement needed for a good range validation capability is on how closely the transmission time is to the reported MSO. We want to make this as tight as is realistically possible.

It is important that we choose these requirements well so that the application has a chance to neutralize spoofers. There are plenty of areas for error to be introduced into the ranging calculation, such as position and velocity inaccuracies which are unavoidable.