# DISCUSSION CONCLUDING AAS 13-519

ROB SEAMAN asked if HARLAN STENN wanted "these timestamps to be signed." STENN replied that it is certainly easy to sign them and that is a mechanism that must be present. Based on discussions with a lot of other people, STENN expected there to be three sources of time forever. If one wants an authenticated timestamp then one should be prepared to pay for that service, even if the fee is as small as something like a tenth of cent. This is because there is legal liability and a warranty implied with a signed timestamp.

SEAMAN responded that digital signatures for NTP did not seem as easy as STENN made it sound, based on SEAMAN's reading of internet discussions. STENN replied there are several issues with NTP and authentication, one of which has to do with nomenclature. People think that 'authenticated NTP' means that either "you know you are talking to the server you want," or, the server gets to decide "whether or not it is going to tell you the time." The problem with the latter situation is that it is much faster to simply provide the time regardless, rather than decide if the time should be given or not. This avoids a denial-of-service attack, where people send "please authenticate me—and then I don't know if I want to do that." The other issue is digitally-signing time. How easy is to get a timestamp with a digital signature of that timescale? Issues of traceability and auditing come into that, and eventually the issue of billing payment for that timestamp, and finally being able to stand up in a court of law saying "We certify here is how we can prove to you that we delivered that timestamp when we said we did."

Noting that the timescale is included in the proposed timestamp, ARNOLD ROTS asked if there would be any merit to a timestamp providing *multiple* timescales by including instantaneous offsets to other timescales in the timestamp? STENN said that is possible but he thought that would be far more work than necessary, because users could have a conversion utility. A user thus picks a timestamp and sets up his system how he wants. This is "one of the questions on our checklist: How often are you going to be looking at a timescale where, if you cannot objectively convert from one to the other, then what is the point?"

ROTS replied that, for instance, the broadcast distribution of UTC includes the DUT1 offset[*] which is variable in principle; ROTS wondered if the inclusion of something like that would have benefit. STENN said he would love to discuss that and possibly do it; the timestamp format could possibly include some kind of an extension field to include their idea of what is broadcast, if that was desired. STENN expects weekly updates to support UT1 from the IERS: "If today is the last day of that thing [weekly bulletin] and I am not going to have an update that is good for a week for anything timestamped with this one, I am going to know that in a week I will have the ability to go back and correct all those timestamps to go from 'projected time' to 'actual time'". CHRIS TUASON thought an extension field was not needed; with two 64-bit words there would be plenty of room to do it. STENN countered that "*timestamp* is not *timescale*," understanding that TUASON

[*] DUT1 is UT1−UTC precise to 0.1 s, per ITU-R Recommendation TF.460, and announced via IERS *Bulletin D*.

was talking about the fields for error bounds in the measurement. TUASON clarified that he was suggesting limiting the error bounds to pack other information into there. STENN thought somebody at the colloquium may have an idea as to how far to limit an error bound.

JIM KIESSLING said that computing an error after the fact is not the same as going back and performing retrospective corrections to a database; KIESSLING wondered why STENN was endorsing retrospective corrections. STENN said he was not necessarily saying one should correct the database. "If I have IERS *Bulletin A*, here are the projected values for this week. Once we get the new values we have the ability to convert from last week's IERS values to this week's IERS values to get a better fix on when it went." This goes to a mechanism; one may say that is a stupid policy and one may not want to do it, but STENN wants to make sure that the mechanism is there because it has to be done for other things. So as long as the mechanism allows this, the use case is of less concern. "If we say something is not supported and you ask a library to do it, and someone says we cannot do this because you asked us to do a stupid thing. Take a case where I went ahead and did something because I knew what time it was supposed to be in a small town that changes its time zone whenever it feels like it. If you do not have the update because it did not make it into the other A.D.O.[*] database in time, you have the ability to bump that version number when the Eastern time zone is updated. You can still go back and convert it out of the old one, but if you have to go talk to a local, you do not want to be telling them the time to look at when you had it wrong and they had it right. You still need to be able to convert the three time scales."

STEVE ALLEN said that the timescale library then needs to include things like tz data version numbers. STENN said "absolutely, but the point is that if you do not have that, you have no idea what is in your timescale. If I have a timestamp that says this was done on Eastern Standard Time as of this date, that does not necessarily mean that was the date of your time zone database. So if you do not have the right numbers to support it, you cannot do useful things with your timestamps." SEAMAN asked if the timescale was more like more like a MIME type in this database. STENN said he was thinking 32 bits worth because "his gut was telling him that is probably enough for the moment." STENN was fine if bigger values were needed. Google Summer of Code was to start in a week or two and the NTF's intern working the project had been really excited about reading everything STENN gave him for the last month.

SEAMAN said that is the process by which one constructs a new timescale, which could be quite a complex thing, including the location of the observer. STENN agreed, saying "We never said it was going to be easy." ALLEN said that there is the added complexity of the polysemy of the older time scales; there is the current UT2 expression which is not really current anyway, and then the original UT2 expression, and so on. STENN said that as long one of them could be done and given a version number, the others could be done if people have the desire and motivation (which means that they either do it themselves or they pay somebody to do it), because then we have ability to expand it and give people what they want. SEAMAN said that it could be extended to 64-bits because of the versioning. STENN said "absolutely"; he liked "to make mistakes."

RUSSELL REDMAN said this was a "nifty-sounding idea" and asked if there were any kind of forward plans to integrate this with the rest of NTP; in some of those fields it would be very useful to propagate from one machine to the next. STENN said that the Network Time Foundation releases open source. REDMAN clarified that he was thinking at the design level for the next version of NTP. STENN said that, for now, it was mostly conceptual. Network Time Foundation was

[*] The initials of Arthur David Olson, originator of the time-zone database now managed by Internet Assigned Numbers Authority (IANA).

starting to code more of it and as the Google Summer of Code happened, NTF would be releasing more and inviting folks to look at it. The good news is that STENN expected there to be a workable implementation that people could test and see the behavior. And the fact that NTF was providing it in C would make it easy to convert to other languages. The intent is that it should be portable and pretty quick. If people want to do something difficult, it may take more time but NTF wants to make sure people have a mechanism that will let them do it.