The Global Leader in Resilient **PNT**

Technical Note:
Differences Between NTP and SNTP

NTP (Network Time Protocol) and SNTP (Simple Network Time Protocol) are very similar TCP/IP protocols in that they use the same time packet from a Time Server message to compute accurate time. The procedure used by the Time Server to assemble and send out a time stamp is exactly the same whether NTP (i.e. full implementation NTP) is being used, or if SNTP is being used.

The difference between NTP and SNTP is the time synchronization program running on each individual PC (Server or Workstation). The time program, whether it is a Windows built-in program like W32Time (which uses the SNTP protocol) or a third-party add-on, determines which protocol is being used not the Time Server. The Time Server does not care which protocol is being used.

The difference between NTP and SNTP is in the error checking and the actual correction to the time itself. The NTP algorithm is much more complicated than the SNTP algorithm. NTP normally uses multiple time servers to verify the time and then controls the slew rate of the PC. The algorithm determines if the values are accurate using several methods, including fudge factors and identifying time servers that don't agree with the other time servers. It then speeds up or slows down the PC clock's drift rate so that (1) the PC's time is always correct and (2) there won't be any subsequent time jumps after the initial correction.

Unlike NTP, SNTP usually uses just one Time Server to calculate the time, then "jumps" the system time to the calculated time. It can, however, have back-up Time Servers in case one is not available. During each interval, it determines whether the time is off enough to make a correction and if it is, it applies the correction.

If not completely clear, perhaps an analogy of comparing and adjusting a wristwatch to a clock on the wall would help. The wristwatch is analogous to the "client" device (like a PC) and the clock on the wall is the time server.

With SNTP, you always look at the clock at pre-determined intervals. Let's say one-per-hour. (As an aside, the act at comparing time for computer synchronization is known as a "poll".) When you think it is 12:00:00 you look at (poll) the clock to see that it is 11:59:57. You are three seconds fast, so you set your watch back three seconds. You do not do anything else until 1:00:00. You look again at the clock to see that it is 12:59:57, again three seconds fast, and again you set your watch back three seconds. Every hour, you reset your watch 3 seconds to be in sync with the clock on the wall. From an error perspective, you are most accurate immediately after the poll, and you progressively get worse with the maximum error immediately before the poll, when a sudden adjustment occurs, such as is the case when time goes from 12:59:57 to 12:59:58 to 12:59.59 to 1:00:00 to 12:59:57.

If a maximum error of three seconds and the discontinuity of the time scale bothers you, then consider the NTP case. Here, you want to do something with the knowledge that your watch is gaining three seconds every hour so you don't have to change it so often. Simply compensate for the drift by using your error versus time measurements. You do not need to use the same measurement period all the time. All you need to know is rate and direction of the change. After you have a pretty good feel for the drift, you can program your watch to adjust in real-time. You want to make very little adjustments so at any given time you are in sync with the clock on the wall without even looking at it. Of course, the drift rate may change over time, so you do want to continually poll the clock, and apply the best correction you can come up with. And with that you get a wristwatch that is seemingly never out of synchronization!

13 February, 2019 - TN06-101 (D)