Maintain Your Time

Whitham D. Reeve (© 2012 W. Reeve)

Abbreviations in this article:
CPU: Central Processing Unit
GMT: Greenwich Mean Time
GPS: Global Positioning System
LAN: Local Area Network
NIST: National Institute of Standards and Technology
PC: Personal Computer
NTP: Network Time Protocol
PTP: Precision Time Protocol
RFI: Radio Frequency Interference
SID: Sudden Ionospheric Disturbance
SNTP: Simple Network Tine Protocol
TOD: Time Of Day
UT: Universal Time
UTC: Coordinated Universal Time
WAN: Wide Area Network

Introduction

The email blared "*M-class solar burst right now: 18h30 TU*" and it included a Radio-SkyPipe chart showing a burst peak at 1821 and a rising peak at 1828 on the chart's right edge. I received the email at 1731, about 1 hour *before* the stated time – was this a time traveling email? Probably not, but this event demonstrates why I wrote this article. One more thing, I am pretty sure the author of that email meant UT, Universal Time, and not "TU".

This article discusses two aspects of time – the basis for proper time keeping from an amateur radio astronomer's point of view and methods that help ensure computer-generated data are properly time stamped. In particular, since most amateur radio astronomers use a personal computer (PC) to record data, I will discuss methods used to maintain reasonable accuracy of a PC time-of-day, or TOD, clock (also called real-time clock).

When did that happen?

The event described above actually occurred near 1730 UT and not 1830 as claimed, a 1 hour error. It is clear that in our endeavors to receive celestial radio emissions we need to know not only *what* was detected but *when* it was detected. Knowing when is very important because sometimes the only way to verify reception of certain emissions is to compare charted times with other astronomers. For example, Jupiter emissions can be easily confused with ordinary radio frequency interference (RFI) but if other radio astronomers at different locations received similar emissions at the same time, then the time correlation helps confirmation. The same is true of many solar emissions and sudden ionospheric disturbances (SID) due to solar flares.

Most charted data and sampled data have a time/date stamp or time of day information associated with them (figure 1). For the purposes of time correlation, it is necessary that the TOD clocks used by all parties involved in an observation are synchronized to a common time reference. Synchronization of multiple PCs, whether in the same city or spread around the world, is no easy task, especially when humans are involved. I am always surprised by data published on the various amateur radio astronomer technical groups that have grossly incorrect time stamps. Adding to the problem is at least one software application used by many amateur radio astronomers that does not always correctly time-stamp the data (I will briefly discuss this later).

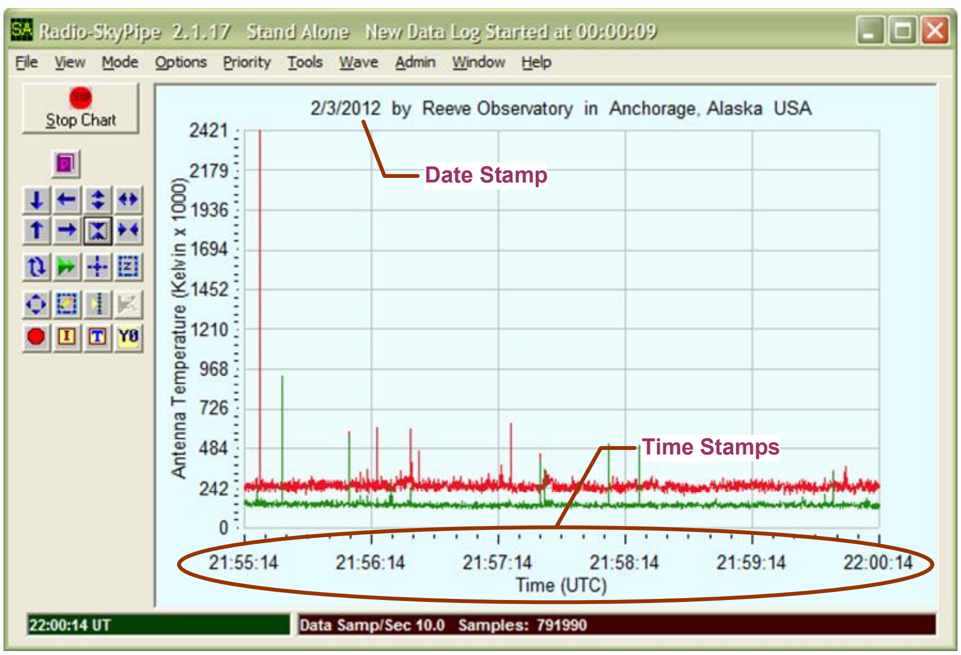

Figure 1 ~ Radio-SkyPipe chart with time information along the horizontal axis and date at the top.

In addition to the obvious error mentioned at the beginning of this article, here are some other reasons I have seen for erroneous time stamps: "*My chart is off by 1 hour because I forgot to reset the clock for daylight savings time*" and "*I have not set the PC clock for several months and it is wrong by something* *between 30 minutes and 2 hours*" and "*my PC clock probably has not been set since the day I bought it*."

It takes time to set the time

Clock accuracy can be viewed as the ability of a hardware clock to prevent deviation and drift once it has been set to a particular time reference, and synchronization is the ability of the clock to get accurate time from a reference time source. The physical accuracy and stability of a given clock is directly related to its cost. PC clocks are inexpensive by design. A typical PC uses a quartz crystal oscillator that is mass produced and costs a tiny fraction of a US dollar. Crystal oscillators are sensitive to temperature and they drift with age. A TOD clock based on this technology must be regularly updated or resynchronized if useful accuracy is to be achieved.

Synchronization is not simple. It requires protocols, algorithms, estimations and interfaces to negotiate with a reference clock in another location. When a computer process is called to update the clock, it must know or at least estimate or assume when it sent the update request, the time it took the request to get to the reference, the time it took to be processed at the reference, the time it took for the response to get back and the time it took to process the response and set the PC clock. To some extent a PC clock must depend on its own accuracy to set itself accurately. Small errors in assumptions and estimates accumulate. A few milliseconds here and there quickly add up to a second, and a hundred milliseconds here and there quickly add up to 10 seconds. Before long the error can have a magnitude of minutes or hours.

Time protocols

The Network Time Protocol (NTP) is widely used for time distribution and to ensure time accuracy in local area networks (LAN) and wide area networks (WAN, for example the internet). NTP was developed as an internet protocol for synchronizing clocks in distributed time servers and clients to a "standard" time. NTP exchanges messages between a time server and client that are used to calculate time offsets and delays. NTP is capable of providing synchronization at the sub-tens of millisecond level in the WAN. Tens of thousands of NTP servers exist in the internet and many (but not all) of them are accessible by public users. A detailed description of NTP can be found at http://www.ntp.org/, and a list of National Institute of Standards and Technology (NIST) time servers in the USA can be found at http://tf.nist.gov/tf-cgi/servers.cgi.

A more recent timing protocol implementation is the Precision Time Protocol (PTP), which is defined in Institute of Electrical and Electronics Engineers (IEEE) Standard 1588 (*Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*). It has been deployed in industries that use packet networks and depend on synchronization at the sub-microsecond level, for example, telecommunications and manufacturing. More information on PTP is here: http://www.ieee1588.com/.

NTP and PTP are not the only time protocols that have been used over the years. For example, a simplified version of NTP, Simple NTP or SNTP, is used in some computer operating systems. The design of most time protocols are directly related to their end-use and the transmission media used to transport them.

What time is it?

Most of us live our daily lives based on the local time. Local time is an adjusted time scale consisting of two elements, a time reference and a time zone, where Local time = Time Reference ± Time Zone. The legal basis for time-keeping and the time reference used in most countries is Coordinated Universal Time (UTC), which is defined in International Telecommunications Union – Radio Communications Sector (ITU-R) Recommendation TF.460.[1] In most other countries UTC also is used even though it may not be legally adopted at the national level. One nice thing about UTC is that the time, in UTC, is the same around the world.

Time zones are a confusing mess based on geography, government jurisdictions and political whims and in rare cases on geographic longitude. For example, Alaska at one time had four time zones because of its size. Nowadays, because politicians are so smart, most of the state is in one time zone and the far reaches of the

Aleutian Islands are in another. The Aleutians are partly on the other side of the international dateline (whatever that is). The two present Alaska time zones have no correlation whatsoever with longitude or solar time.

In most places in the USA and many other parts of the world, there is a requirement, with demonstrably dubious benefits, to shift to a daylight savings time during certain times of year. This means the local time shifts by one hour twice a year, forward in the spring and backward in the fall.

One quirk with UTC is that it has to be adjusted every so often with "leap seconds". The UTC time scale is the atomic standard of time. It is based on the emissions frequency of certain atoms (cesium) when their electrons jump from one energy level to another (see http://www.bipm.org/en/si/si_brochure/chapter2/2-1/second.html). However, commerce and almost all other human activities are based on daylight and darkness – the position of the Sun in the sky – so we have a generalized time scale called Universal Time (UT), which is based on Earth's rotation rate. There are three variations of UT: [1]

- ☼ UT0 is the mean solar time of the prime meridian obtained from direct astronomical observation;
- ☼ UT1 is UT0 corrected for the effects of small movements of Earth relative to the axis of rotation (polar variation). UT1 is the one most commonly seen and used and often is simply denoted UT;
- ☼ UT2 is UT1 corrected for the effects of a small seasonal fluctuation in the rate of rotation of Earth.

Because of the variations in Earth's rotation rate, UT1 and UTC would drift apart over time. To minimize the time difference, leap seconds were implemented in 1972 to compensate and to keep UT1 and UTC within about 1 second of each other. If leap seconds are not used, Earth rotation time, UT1, would differ from UTC, for example, by 2 or 3 minutes in 2100 and 30 minutes in 2700, and a lot of bus and train schedules would be disrupted. There have been many discussions among scientists around the world for many years about eliminating the leap second. I will discuss this topic in more detail in a future article.

For now, how do we correlate our observations with observers in other time zones or other places in the world? The answer is that we should stamp all of our observations with times and dates based on UTC and never use local time or any other time scale. Whenever we post an observation or discuss it, we should never use local times or dates except perhaps parenthetically to put certain observations in proper context (for example, when it is relevant that the observation was made during daytime or nighttime). To be sure, UT1 and UTC are not the only time scales used in radio and optical astronomy, but the others are beyond the scope of this article. The next question is how can you set the time on your PCs to UTC and keep it reasonably accurate? I will answer that in the following sections.

Microsoft Windows environments

Many amateur radio astronomers use PCs with various versions of the Windows operating system. In the following discussion, I will concentrate on Windows XP and Windows 7. PCs running Windows XP and 7 by default use the Windows Time service (figure 2). In Windows a "service" is a piece of software that performs a certain task. If the PC is in a local area network, the Windows Time service starts automatically. Windows Time service uses NTP and strives to ensure that all PCs in the network use the same time. Whether or not it is the correct time depends on the circumstances. The setting may not be accurate but at least the interconnected computers use the same nominal time.

If the PC is connected to the internet, and Windows Time service is properly configured, it will attempt to contact a time server to set the clock accurately. However, if the PC is not part of a network or is not connected to the internet, the Windows Time service may not be started, in which case the user must figure out how to use it or to use some other method. Additional information on the Windows Time service can be found here: http://technet.microsoft.com/en-us/library/bb490605.aspx.

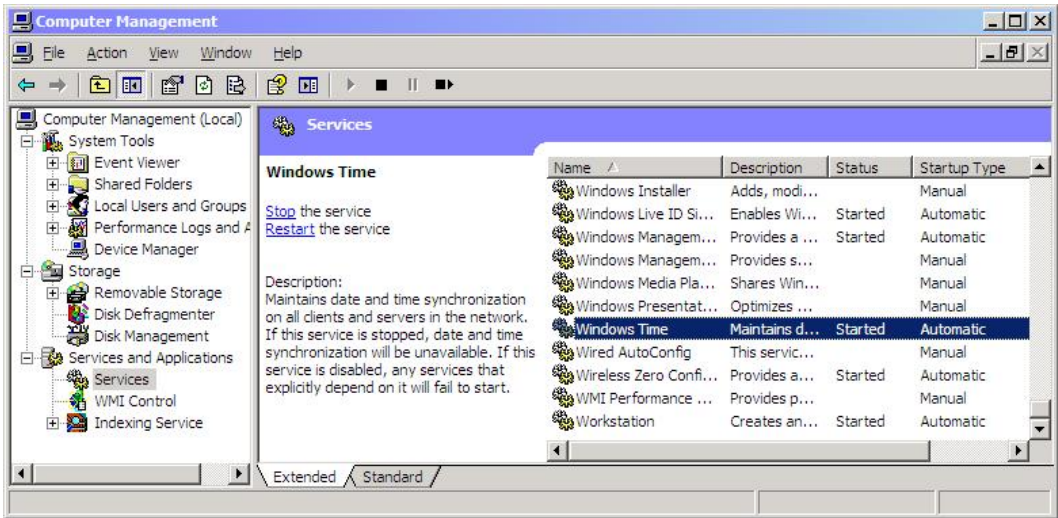

Figure 2 ~ Windows Time service is an underlying service on all modern Windows-based PCs.

Time setting alternatives

I have found that Windows Time service can be unreliable, possibly because of my network configurations – I really have no idea why. Rather than monkey with it, I use a free alternative, which I have found over the last several years to be non-intrusive and to always work very well. The time application program I use is Symmetricom's free SymmTime program (http://www.symmetricom.com/resources/downloads/symmtime/). Other application programs are available that reliably maintain and synchronize PC clocks, but I have not used anything but SymmTime for around the last 10 years. However, I discuss another worthwhile PC time-keeping alternative called NISTime and provide a link to a list of time-keeping software in a sidebar. Next, I will describe the Windows setup and then the SymmTime setup.

Windows Time service setup

The following provides specific setup for Windows XP but the Windows 7 setup is very similar. First, the PC clock should be set to Coordinated Universal Time (UTC) and not local time. This greatly reduces the possibility of incorrect time stamps. To set the PC clock to UTC click on Start – Control Panel – Date and Time. Select the Time Zone tab. From the drop-down list select *(GMT) Coordinated Universal Time* (Figure 3) and then OK. After setting the PC to UTC, it may be necessary to adjust the time zone or offset settings in your various software applications that do the actual time stamping on your observations.

Figure 3 ~ Windows Date and Time Properties

It is very important that the PC clock is accurate at all times, especially during data collection. As mentioned above Windows has built-in *Internet Time* synchronization capability but it may not be turned on. First, make sure your PC is connected to the internet, and then determine if your system has Internet Time synchronization capability, then click Start – Control Panel – Date and Time. Select the Internet Time tab, if it exists, and then check the box A*utomatically synchronize with an Internet time server*. If available, select *time.nist.gov* server and then *Update Now*. Wait a moment. If the update is successful you will see a screen similar to figure 4-left. If not, you will see an error similar to figure 4-right. The update could fail for any number of reasons and you could either troubleshoot or install an alternative software tool as described in the next section.

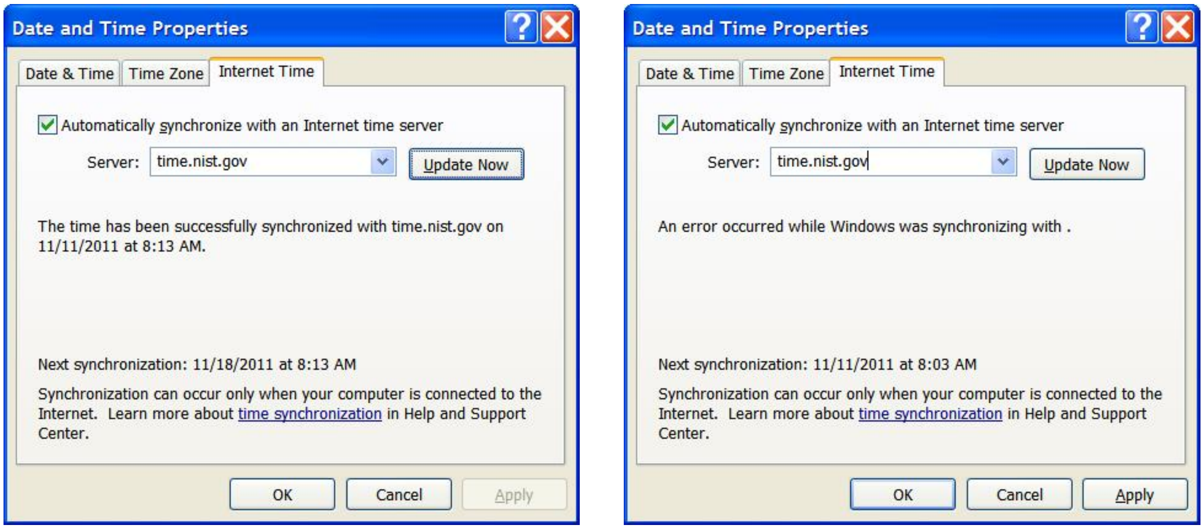

Figure 4 ~ Additional Windows Date and Time Properties settings

If you decide to use the Windows Time service to update your PC clock, you should beware of its default settings. The default update interval is 604800 seconds, or 7 days. This is not nearly frequent enough for most PCs and useless for data collection. The default update interval can be changed through a Registry edit as follows:

- Start – Run...
- Type Regedit – OK
- Select: *HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\W32Time\TimeProviders\NtpClient*
- Right-click *SpecialPollInterval* and select *Modify*
- Select the *Decimal* radio button in Base
- Change the default decimal value from *604800* seconds to some other value. The value to be used depends on the drift rate of the PC clock. Typically, you would set it to update the clock at 10 to 60 minute intervals (example, for 15 minutes update rate, set the value to 900)
- Click OK

SymmTime setup

Your version of Windows may not have the *Internet Time* tab described above or it may not be reliable or you may not want to use it. In those situations, I suggest SymmTime by Symmetricom. If you choose not to use the Windows Time service, you should uncheck *Automatically synchronize with an internet time server* in Windows Date and Time properties. You still should set your PC's clock to GMT/UTC as previously described regardless of the application that handles time synchronization. You can configure SymmTime to display both UTC and local time.

After downloading and installing the SymmTime application, it should start and run. You should see a small round clock icon on the Windows taskbar. Right-click the SymmTime window and select *Sync Options*.... Under the *Synchronize with Server* setting select the radio button *Periodically, every xx minutes*, where xx is your desired update interval, typically 10 to 60 minutes (Figure 5-left). Click OK when done.

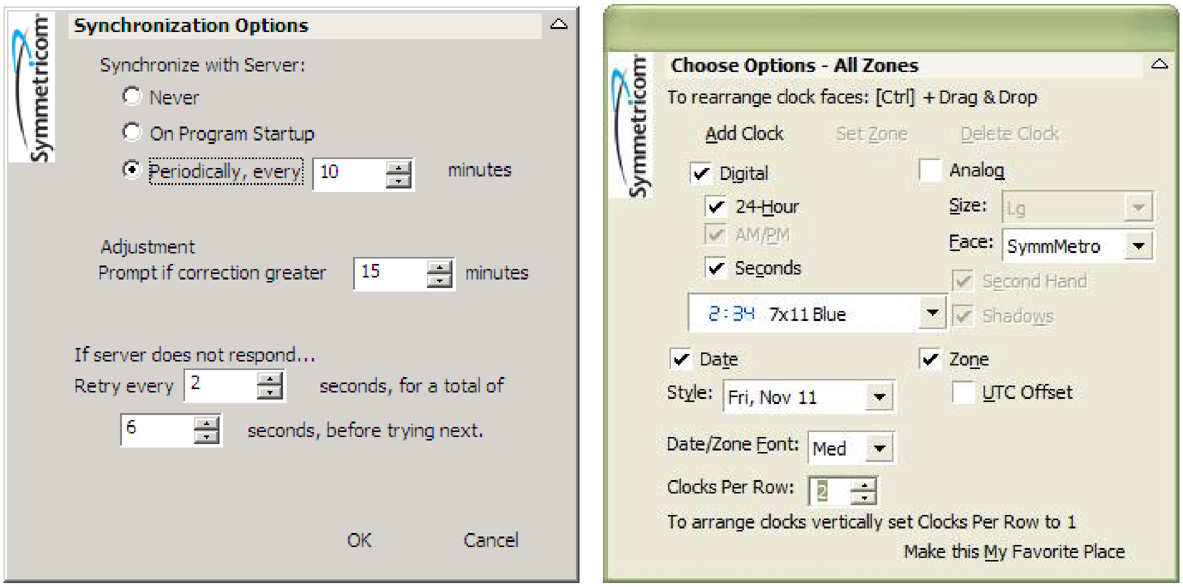

Figure 5 ~ SymmTime Option settings

Right-click the SymmTime window again and select *View Options...* (figure 5-right). Setting up various time zone displays is self-explanatory (or you can use the *Help...* function), but you should setup two time displays, one to show UTC/GMT and another to show your local time. Note that the time zone displays are simply that – displays. The PC TOD clock itself is governed according to the settings described at the beginning of the previous section. You can setup the two time displays in a horizontal or vertical format (figure 6).

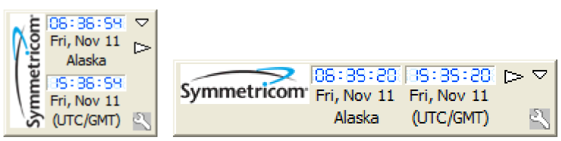

Figure 6 ~ SymmTime vertical and horizontal display configurations

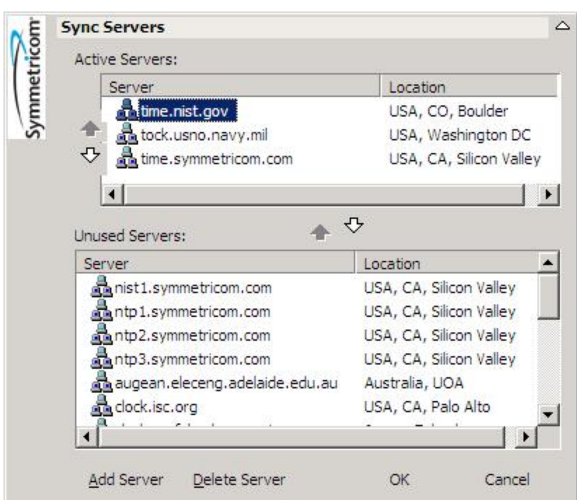

Figure 7 ~ Time Server selection in SymmTime

SymmTime can access any public time server. The default time server settings should be fine but you can change them. Right-click the SymmTime window and select *Sync Servers....* You will see a scroll-list of *Unused Servers* in the bottom frame (figure 7). You can customize the list by using the *Add Server* and *Delete Server* selections at the bottom. You also can add or delete servers to the *Active Servers* in the top frame by selecting a server and then clicking on the arrows between the two frames. The individual servers are quite reliable but you can place several in the *Active Server* list. This way, if SymmTime is unable to contact the first one on the list, it will attempt to contact the next one, and so on. It has been my experience that, if SymmTime cannot contact one of the listed public servers, the problem lies in the PC or its access to the internet and all other times servers are similarly inaccessible. When SymmTime is unable to contact the servers the clock icon in the Windows taskbar will flash red.

The last step is to place SymmTime in the Windows Startup folder. This will ensure that SymmTime starts automatically when the PC is rebooted. One way of doing this is to go to the Program Files folder where SymmTime is installed (typically c:\Program Files\Symmetricom\Symmtime\). Right-click and drag the file GeTTime.exe to Start – All Programs – Startup. That places a shortcut in the Startup folder, and it will automatically run whenever the PC reboots or is started.

One minor inconvenience with SymmTime is that, when you have setup a local time display, it does not automatically compensate for daylight savings time. For data collection purposes this does not matter since your PC TOD clock is set to UTC anyway. However, if your location uses daylight savings time and you have setup SymmTime to display the local time, the display will be off by one hour during spring, summer and fall. For example, Alaska Standard Time (AST) is nine hours earlier than coordinated universal time, or UTC–9. On the first Sunday in March, we switch to daylight savings time, UTC–8, and on the second Sunday in November we switch back to standard time. In order for SymmTime to display the correct time, it is necessary to Choose New Time Zone – New Zone and then setup the 8 hour difference. This new Zone is applied to the local display until standard time goes into effect. It is simple to do. If you encounter any problems with SymmTime or have questions on it, Symmetricom provides a support contact on their website (and, yes, they do respond!).

PC time statistics

After SymmTime setup is completed, you can check the amount of correction made when SymmTime queries the time server. You should let SymmTime run at least 24 hours before making this check. Right-click on the SymmTime window and select *Sync Status...*. The window that opens shows some statistics including the average clock offset and recent history (figure 8). Just below the *Avg Offset* value is a *Recent History* frame. It shows the correction made during each time query. It also will indicate if the time server is unreachable for any reason. If you find that the update rate is too frequent or too infrequent, adjust the *Synchronize with Server* setting in *Sync Options...*. as previously described.

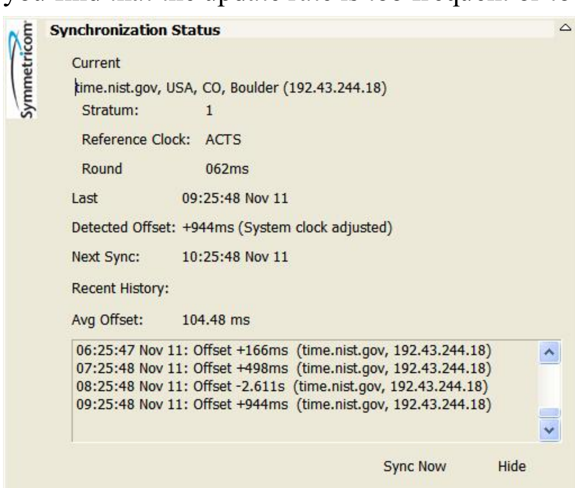

Figure 8 ~ Time statistics available in SymmTime. The update rate in *Sync Options...* for the PC shown here is 60 minutes. The maximum offset during the previous 24 hours was 2.6 seconds and the average offset was a little more than 100 ms.

If the PC is runs only during observations or during certain hours, it is necessary to resynchronize the PC TOD clock each time the PC is started. If you use SymmTime and set it up as described above, it will automatically resynchronize the clock within the interval specified in *Sync Options... – Periodically, every xx minutes*. Therefore, if your setting (xx) is 10 minutes, the program will resynchronize the PC clock within 10 minutes after SymmTime is loaded.

What do you expect?

A typical commercial-grade crystal has a frequency tolerance of, say, ±50 parts per million (ppm) at 25° C and ±100 ppm frequency stability over a temperature range of 0 to 70° C. For example, a new 1.0 MHz crystal may operate between 999 975 and 1.000 025 MHz at 25° C and may drift between 999 900 and 1.000 100 MHz over the given temperature range. To see what this means, let us assume the TOD clock in a PC drifts no more than 25 ppm when averaged over 1 day. At the end of a 24 hour time period, the clock would be fast (or slow) by about 2.2 seconds (25×10^{-6} x 24 h/d x 3600 s/h). If that 25 ppm average offset is sustained for 30 days, the clock will drift by slightly more than 1 minute in a month. This is not especially bad, but it is my experience that most PC clocks drift much more rapidly than 1 minute per month. For one thing, tolerances of crystals used in PCs could be much worse than those used in this example. For another, the inside of some PCs can easily exceed 70° C and cause the crystal oscillator to operate outside of its specified range. I cannot even say with any confidence that the crystals used in high-quality PCs are better than in low-quality PCs (presumably they are, but I would not go to the bank on it).

I operate a number of PCs at my receiver stations, lab and office. I run SymmTime on all of them and use various update intervals. I took a snapshot of the Sync Status windows on each PC for comparison and summarized the information in table 1. The Lightning and Receiver PCs are used for data collection so their update intervals are relatively short. It is seen that I can expect my PC clocks to be accurate within a few tens of milliseconds. This assumes the clock accuracy read by SymmTime is not degraded by application software that actually collects and stores the data or by some other applications running on the PC; if there is no degradation then the stated accuracy is more than adequate for my types of radio astronomy.

PC name	Mfr and Model	Update interval	Average offset	Typical correction	Round-trip delay	Purpose
Lightning	Shuttle K45	5 min	+19.08 ms	6 ~ 54 ms	109 ms	Data collection
Lightning2	Shuttle K48	5 min	+14.92 ms	8 ~ 21 ms	109 ms	Data collection
ReceiverA61e	Lenovo A61e	10 min	–1.95 ms	0 ~ 8 ms	109 ms	Data collection
LenovoT61	Lenovo T61	10 min	–8.95 ms	3 ~ 10 ms	140 ms	Laptop
LabA61eNo1	Lenovo A61e	15 min	–3.15 ms	0 ~ 133 ms	109 ms	Lab
LabA61eNo2	Lenovo A61e	15 min	+33.62 ms	32 ~ 35 ms	78 ms	Lab
LabA61eW7	Lenovo A61e	60 min	+242 ms	74 ~ 84 ms	124 ms	Lab
AV8-MX	Custom	60 min	+156.62 ms	152 ~ `169 ms	109 ms	Office

Table 1 – PC time of day clock statistics at my observatory in Anchorage. All PCs had been running at least 24 hours prior to readings, all use SymmTime time update software, and all are in an ordinary environment in terms of temperature and humidity. The column "Round-trip delay" is SymmTime's measure of the delay determined by NTP time for queries to the time server.

Application software

The question of TOD accuracy degradation by application software is of interest. I mentioned earlier that at least one popular radio astronomy software application sometimes applies incorrect time stamps to data. According to Deborah Scherrer, Director of the Stanford Solar Center, the SuperSID software obtains its initial time each day from the PC TOD clock. However, as the software collects data over the next 24 hours it assumes that each data point occurs 5 seconds after the one before it. It does not refer to the TOD clock and, as has been observed by many users, may drift away.

If you do not observe sudden ionospheric disturbances (SID) with the SuperSID software, this is not a problem for you. However, there are other problems waiting to strike. Everyone who uses a PC based on the Windows operating system has observed freezes and stalls, in which the PC seems to stop running momentarily, maybe for a few seconds, tens of seconds or several minutes. What happens to time stamps associated with data collection during these periods? I have not investigated this question but can guess there is loss in accuracy. Probably the most notorious applications that lead to these kinds of problems are anti-virus (AV) and software firewall programs.

The amateur radio astronomer is well advised to try different brands of AV software – there are several high-quality free programs available, and paid programs seldom are the best. It also is obvious that what works well on one PC may not work well on another, making experimentation necessary. I use the free Microsoft Security Essentials (MSE) along with the built-in Windows Firewall (my LAN router also has a firmware firewall). Even though I set MSE's CPU usage during scans to 10% and to run only at certain times, I still see occasional stalls of a few seconds, but these are not pervasive enough to be a problem for me. I do not know their source, anyway, except that MSE seems to be involved. For serious data collection it is important that only the bare minimum essential applications be running, and it is unfortunate that anti-virus software is essential for any PC connected to the internet.

Conclusions

Amateur radio astronomers wishing to compare data and observations collected by their PCs must have reasonably accurate time-of-day clocks. With very little effort and no extra cost it is possible to achieve 100 ms or better time-of-day clock accuracy by using simple software applications that regularly update the PC clock using publicly available time servers.

References

[1] ITU-R Recommendation TF.460-6, Standard-frequency and time-signal emissions, International Telecommunications Union – Radio Communications Sector, 2002 (http://www.itu.int/rec/R-REC-TF.460/en).

Whitham Reeve was born in Anchorage, Alaska and has lived there his entire life. He became interested in electronics in 1958 and worked in the airline industry in the 1960s and 1970s as an avionics technician, engineer and manager responsible for the design, installation and maintenance of aircraft electronic equipment and systems. For the next 38 years he worked as an engineer in the telecommunications and electric utility industries with the last 34 years as owner and operator of Reeve Engineers, an Anchorage-based consulting engineering firm. Mr. Reeve has been registered as a professional electrical engineer and has BSEE and MEE degrees. He has written a number of books for practicing engineers and enjoys writing about technical subjects. Since 2008 he has been observing electromagnetic phenomena associated with the Sun, Earth and other planets and presently is a SARA Director at Large and a Contributing Editor of the SARA journal, *Radio Astronomy*.

**********SIDEBAR ON NEXT PAGE**********

Sidebar ~ PC Time-Keeping Alternatives

The National Institute of Standards and Technology (NIST) in the USA provides a list of publishers of time and frequency software: http://www.nist.gov/pml/div688/grp40/softwarelist.cfm. NIST also has available for free their own PC time-keeping software called NISTime: http://www.nist.gov/pml/div688/grp40/its.cfm. NISTime is very simple to setup and use and I recommend it as an alternative to SymmTime described in this article. However, NISTime does not have the same features, for example, it does not provide statistics and its minimum update interval is 1 hour, but it is very reliable and non-intrusive (below). NIST also has a browser "widget" that may be used to indicate the correct time on a webpage: http://www.nist.gov/pml/div688/time_010511.cfm

Another time-keeping alternative is Radio-SkyPipe Pro (http://www.radiosky.com/), a very popular charting program with a built-in feature that uses SNTP to the set the PC TOD clock (below). This feature is found in the Tools menu – Atomic Time. However, Atomic Time updates the PC clock only when RSP is running. Note that the free version of Radio-SkyPipe does not have the Atomic Time feature.