Computer Time Synchronization

Michael Lombardi
Time and Frequency Division
NIST
email: lombardi@boulder.nist.gov

The personal computer revolution that began in the 1970's created a huge new group of time and frequency users, those people who need to keep computer clocks on time. As you probably know, computer clocks aren't particularly good at keeping time. Simple clocks like your wristwatch and most of the clocks in your home usually keep better time than a computer clock.

The poor performance of a computer clock can cause problems, since many computer applications require time kept to the nearest second or better. For example, the computers in a financial institution must keep very accurate records of when transactions were completed, for legal or other reasons. Computer systems that make physical measurements and acquire scientific data need to know precisely when the measurements were made. Software used on a manufacturing floor may need to turn a piece of equipment on or off at a specified time. Also, any system involved with synchronous communications must keep accurate time. For example, radio and TV stations may need computers that can switch feeds or link up with remotes at the right time.

This paper describes several methods to keep accurate time by computer. Before looking at these methods, let's look at how a PC-compatible computer keeps time.

Section 1.1 - How a Personal Computer keeps time

Since the introduction of the IBM-AT personal computer in 1984, all PC-compatible computers have kept time the same way. Each PC contains two clocks, regardless of whether it uses the 286, 386, 486, or Pentium microprocessor (or a derivative). These clocks go by several different names, but for simplicity, we'll call them the software and hardware clocks. The software clock runs when the computer is turned on and stops when the computer is turned off. The hardware clock uses a battery and runs even while the computer is turned off.

The software clock is generated by an Intel 8254 timer-counter (or a functionally equivalent device). This timer-counter generates an interrupt every 54.936 milliseconds, or about 18.2 times per second. The computer's BIOS (Basic Input Output System) contains a software routine that counts the interrupt requests and generates a time-of-day clock that can be read or set by other software programs. For example, DOS uses time-of-day information from the software clock to date and time stamp files.

The software clock is a poor timekeeper. Its timing uncertainty is limited by the stability of the interrupt requests. Any change in the interrupt request rate causes the clock to gain or lose time. If you leave your computer turned on for long periods, the software clock might be off by large amounts, perhaps a minute or more for every day that the computer was left turned on. It's also possible for an ill-behaved software program to use the timer-counter for another purpose and change its interrupt rate. This could cause the clock to rapidly gain or lose time.

The software clock also has limited resolution. It can only display values that are even multiples of the time interval between interrupts (55 milliseconds). For example, 00:00:01.00 could never be displayed by the software clock. The closest possible values it can display are 00:00:00.98 and 00:00:01.04.

The single biggest drawback of the software clock, however, is that when the computer is turned off, the clock stops running and loses all of its time-of-day information. For this reason, a hardware clock is also necessary. The hardware clock is based on the Motorola 146818 Real Time Clock Chip, or an equivalent device. When the computer is turned off, the hardware clock runs off batteries. When the computer is turned back on, the software clock starts running again and sets itself (within 1 second) to the hardware clock. Although the hardware and software clocks are synchronized at power-up, they run at different rates and will gain or lose time relative to each other while the computer is running.

The hardware clock is updated once per second and cannot display fractions of a second. Its timing uncertainty is determined by the quality of the crystal oscillator it uses as its time base. These crystals cost less than $1 in single quantities and offer only marginal timekeeping performance. They are sensitive to temperature and other factors and their frequency uncertainty is not likely to be better than 1×10^{-5} (about 1 second per day). In actual operation, most hardware clocks gain or lose about 5 to 15 seconds per day, with 10 seconds per day being typical. Although the hardware clock usually outperforms the software clock, its performance pales in comparison to even a low-cost wristwatch.

As you can see, neither the software or hardware clock is suitable for accurate timekeeping. Fortunately, there are ways to solve the PC timekeeping problem. Let's start by looking at how to synchronize a computer clock using a dial-up phone service.

Section 1.2 - Dial-Up Time Setting Services

There are a number of dial-up telephone services that allow a computer with a modem to synchronize its clock. To illustrate how these services work, let's look at NIST's Automated Computer Time Service (ACTS), which went on-line in 1988.

ACTS requires only a computer, a modem, and some simple software. When a computer connects to ACTS by telephone, it receives an ASCII time code. The information in this time code is then used to set the computer clock to the correct time. ACTS is usable at modem speeds up to 9600 baud with 8 data bits, 1 stop bit, and no parity. To receive the full time code, you must connect at a speed of at least 1200 baud. The full time code is transmitted every second and contains more information than the 300 baud time code, which is transmitted once every 2 seconds. Table 1.20 describes the full ACTS time code.

JJJJJ YR-MO-DA HH:MM:SS TT L UT1 msADV UTC(NIST) <OTM>
JJJJJ is the Modified Julian Date (MJD). The MJD is the last five digits of the Julian Date, which is simply a count of the number of days since January 1, 4713 B.C. To get the Julian Date, add 2.4 million to the MJD.
YR-MO-DA is the date. It shows the last two digits of the year, the month, and the current day of month.
HH:MM:SS is the time in hours, minutes, and seconds. The time is always sent as Coordinated Universal Time (UTC). An offset needs to be applied to UTC to obtain local time. For example, Mountain Time in the U. S. is 7 hours behind UTC during Standard Time, and 6 hours behind UTC during Daylight Saving Time.
TT is a two-digit code (00 to 99) that indicates whether the United States is on Standard Time (ST) or Daylight Saving Time (DST). It also indicates when ST or DST is approaching. This code is set to 00 when ST is in effect, or to 50 when DST is in effect. During the month in which the time change actually occurs, this number will deincrement every day until the change occurs. For example, during the month of October, the U.S. changes from DST to ST. On October 1, the number will change from 50 to the actual number of days until the time change. It will deincrement by 1 every day, and reach 0 the day the change occurs.
L is a one-digit code that indicates whether a leap second will be added or subtracted at midnight on the last day of the current month. If the code is 0, no leap second will occur this month. If the code is 1, a positive leap second will be added at the end of the month. This means that the last minute of the month will contain 61 seconds instead of 60. If the code is 2, a second will be deleted on the last day of the month. Leap seconds occur at a rate of about one per year. They are used to correct for irregularity in the earth's rotation. UT1 is a correction factor for converting UTC to an older form of universal time that is still used in navigation. It is always a number ranging from -0.8 to +0.8 seconds. This number is added to UTC to obtain UT1.
msADV is a five-digit code that displays the number of milliseconds that NIST advances the time code. It is originally set to 45.0 milliseconds. If you return the on-time marker (OTM) three consecutive times, it will change to reflect the actual one way line delay.
The label UTC(NIST) is contained in every time code. It indicates that you are receiving Coordinated Universal Time (UTC) from the National Institute of Standards and Technology (NIST).
The on-time marker (OTM) is a single character sent at the end of each time code. The OTM is originally an asterisk (*) and changes to a pound sign (#) if ACTS has successfully calibrated the path.

Table 1.20 - The ACTS time code

The last character in the time code is an asterisk (*). The asterisk is called the on-time marker (OTM). The time values sent by the time code refer to the arrival time of the OTM. In other words, if the time code says it is 12:45:45, this means it is 12:45:45 when the OTM arrives. Of course, there is some delay between the time the OTM leaves NIST and the time it arrives at your computer. Some of this delay is the actual data transmission time. However, most of the delay occurs when the modem processes the incoming data and sends it to the computer.

Since the OTM is delayed by the time it takes to travel from NIST to your computer, ACTS sends the OTM out 45 milliseconds early. This 45 milliseconds includes the 8 milliseconds that it takes to send the OTM at 1200 baud, 7 milliseconds transmission time to allow for travel from NIST to an average user in the United States, and 30 milliseconds to allow for the modem processing delay. The 45 millisecond advance was chosen based on experiments conducted at NIST using 1200 baud modems.

Advancing the OTM by 45 milliseconds always removes some of the delay. However, to get the least amount of timing uncertainty, we need to advance the OTM by the amount of the actual path delay. ACTS can do this by using a *loop-back* technique to calibrate the path. The loop-back technique works if the user's computer software returns the OTM to NIST after it is received. Each time the OTM is returned, ACTS measures the amount of time it took for the OTM to go from NIST to the user and back to NIST. This quantity is the round-trip path delay so it is divided by 2 to get the one-way path delay. After 3 consecutive measurements have been made, ACTS advances the time by the amount of the one-way path delay. For example, if the one-way path delay is 50.4 milliseconds, ACTS sends the OTM out 50.4 (instead of 45) milliseconds early. At this point, the path is calibrated, and OTM changes from an asterisk to a pound sign (#).

ACTS is a very popular time transfer system. At this writing (1998) the service receives about 11,000 telephone calls per day. If you calibrate the path, ACTS can set a computer clock with an uncertainty of less than 10 milliseconds.

There are a number of software packages available both commercially and through shareware that call ACTS. All of them are far more sophisticated than the simple program listed here. For a current listing, see:

http://gpsmonitor.timefreq.bldrdoc.gov/pdf/timesoftware.pdf

Some software has the ability to automatically adjust the computer clock. For example, a computer clock might gain 4 seconds per day. Using this information, the software can gradually move the time back 4 seconds per day (1/6 second every hour, for instance). You'll have to set the clock a few times so the software can "learn" about the clock's performance. After that, the software can "steer" the clock to within about 1 second per week. This reduces the number of times that you'll have to call.

There are other time setting services around the world that work in much the same fashion as ACTS. Table 1.21 lists some of these services.

Organization	**Location**	**Telephone Number**
British Broadcasting Corporation (BBC)	United Kingdom	0891-51680 (only usable within U. K.)
Federal Institute of Physics and Metrology	Germany	011-49-531-512038
National Research Council	Canada	(613) 745-3900, Ottawa (416) 445-9408, Toronto
National Electrotechnical Institute	Italy	011-391-13487892
National Institute of Standards and Technology (NIST)	Boulder, Colorado	(303) 494-4774
Swedish National Time Service	Sweden	011-468-7410909
Technical University of Graz	Austria	011-433-16472366
Telecom Australia	Australia	Various numbers that distribute local time throughout Australia
Telecommunications Laboratory (TL)	Taiwan	011-886-3-424-5490
United States Naval Observatory (USNO)	Washington, DC	(202) 762-1594

Table 1.21 - Dial-Up Time Setting Services

Section 1.3 - Network Time Setting Services

If your computer is connected to the Internet, you can synchronize its clock to an Internet server and save the expense of using a dial-up service. A complete listing of public Internet time servers is maintained at the University of Delaware. It can be obtained from:

http://www.eecis.udel.edu/~mills/ntp/servers.htm

The dial-up services listed in Table 1.21 all transmit time codes in ASCII format. For the most part, the different dial-up time servers all use a different time code format. Software programs that access these services must be designed so they can extract the information they need from the ASCII time code. Due to the lack of standardization, software that accesses several services must be able to read and interpret several different time code formats.

The Internet time servers provide a higher level of standardization than the dial-up services. Several standard timing protocols were defined in a series of RFC (Request for Comments) documents. You may obtain these documents via the Internet from a number of sites, including:

http://www.alternic.net/info/rfcs/

The three major timing protocols are the Time Protocol, the Daytime Protocol, and the Network Time Protocol (NTP). A fourth protocol, the Simple Network Time Protocol (SNTP) has recently been defined. Table 1.30 summarizes the various protocols and their port assignments, or the port on which the time server "listens" for a request from the client.

NIST operates a Network Time Service from Boulder, Colorado using multiple servers distributed around the country. The NIST servers distribute time using the Time, Daytime, and NTP formats. For a current list of IP addresses for the NIST servers, see:

http://www.bldrdoc.gov/timefreq/service/nts.htm

The format of the Daytime Protocol is very similar to ACTS. Software to access the NIST service (including source code) using the Daytime Protocol can be obtained from:

ftp://time.nist.gov

For a current listing of network client software, see:

http://gpsmonitor.timefreq.bldrdoc.gov/pdf/timesoftware.pdf

Name	Document	Format	Port Assignments
Time Protocol	RFC-868	Unformatted 32-bit binary number contains time in UTC seconds since January 1, 1900.	Port 37 tcp/ip, udp/ip
Daytime Protocol	RFC-867	Exact format not specified in standard. Only requirement is that time code is sent as standard ASCII characters. Often is very similar to time codes sent by dial-up services like ACTS.	Port 13 tcp/ip, udp/ip
Network Time Protocol (NTP)	RFC-1305	Server responds to each query with a data packet in NTP format. The data packet includes a 64-bit timestamp containing the time in UTC seconds since January 1, 1900 with a resolution of 200 picoseconds. NTP provides accuracy of 1 to 50 milliseconds. NTP software normally runs continuously on the client machine as a background task that periodically gets updates from the server.	Port 123 udp/ip
Simple Network Time Protocol (SNTP)	RFC-1769	A version of NTP that does not change the specification, but simplifies some design features. It is intended for machines where the full performance of NTP is "not needed or justified."	Port 123 udp/ip

Table 1.30 - Internet Time Protocols

Section 1.4 - Radio Clocks

One shortcoming of dial-up services like ACTS is that they require making a phone call (often long distance) each time you need to set your clock. If you need to keep your PC's clock on the right second all the time, this could involve a substantial number of phone calls at substantial cost. The network services are less expensive to use, but not every computer is connected to a network. You may find that your application demands continuous access to an accurate time code, without making phone calls or without having a network connection.

If this is the case, you can get accurate time, all the time, by using a radio clock. There are a number of different types of radio clocks that receive time codes transmitted by radio. The costs vary widely, from less than $200 to $20,000 or more. Radio clocks come in several different forms. Some are standalone devices with a digital time display. These are often interfaced to the PC (or to other types of computers) using an interface like the RS-232, RS-422, or IEEE-488. Others are available on plug-in expansion cards that work on the PC or AT bus. Your application can use a radio clock to constantly set the PC clock, or it can use a software driver to get all of its timing information from the radio clock and bypass the PC clock entirely.

Before purchasing a radio clock, make sure that the signal you choose is usable in your area. Also, remember that you'll need to be able to mount an outdoor antenna so you can receive the radio signal. Radio clocks are available that receive WWV/WWVH, WWVB, and GPS.

Radio clocks often provide the reference for Internet time servers, and many of the radio clocks listed in this table support one or more of the protocols listed in Table 1.30. They also often support protocols supported by other operating systems or local area networks. Be sure and check that the clock that you choose produces time in a format compatible with your application.

For a current list of radio clock manufacturers, see:

http://gpsmonitor.timefreq.bldrdoc.gov/pdf/receiverlist.pdf

Section 1.5 - Replacing the Hardware Clock

You can also improve the timekeeping performance of your PC by replacing its clock with a better clock. We mentioned that the hardware clock on a PC uses a very low cost (less than $1) time base oscillator. If you use a clock with a better time base, you can obviously keep better time.

Precision clock boards are available that plug into the PC bus and effectively replace the hardware clock. These boards include a better time base oscillator than the one inside the PC and allow you to use an external oscillator to get even better results. This means that if you have access to a frequency standard (like a quartz, rubidium, or cesium oscillator) you can use it as the time base. With a good time base oscillator, these boards can keep the correct time for many years. Of course, you will still need to synchronize the clock (using ACTS or another service) and check it occasionally.

Another advantage of using a precision clock board is resolution. As we mentioned earlier, the software clock in a PC-compatible has a resolution of 55 milliseconds, and the hardware clock only provides 1-second resolution. There are many applications in science and metrology where more resolution is necessary. With the best precision clock boards, sub-microsecond resolution is attainable.

As we have seen, there are a number of ways to keep accurate computer time. A wide variety of computer timekeeping products and services exist. Services are available to synchronize computer clocks through dial-up, network, and radio links. Precision clock boards are available to replace the built-in hardware clock. In short, if you have a computer timekeeping problem, a solution is readily available.

The mention of company or product names in this paper does not constitute any endorsement by the National Insitute of Standards and Technology.