

NTP Memo (*DRAFT*)

v1.1b by Holger Jakob 03/2003

Abstract

This document discusses the use of the **NTP** protocol on a Linux platform in astronomical observatory environments, in particular the timing accuracy achievable. The numerical values discussed are oriented at submm- and FIR-observations at ground-based and airborne observatories.

Notes

The **NTP** (Network Time Protocol) version referred in this memo is version 4 (NTPv4). Current Linux distributions (like SuSE, RedHat or Debian) come with this version. **NTP** installation and setup is very easy and well documented (see NTP FAQ and HOWTO or RFC 2030 for details). Also technical informations can be found in the Documentation provided with the package.

Requirements

Astronomical applications do normally require special time systems like the Local Sidereal Time (LST) and derivatives (e.g. LAST or JD. Further information can be found in Explanatory Supplement to the Astronomical Almanac.). However, astronomical time systems are usually derived from civil standard time Universal Time Coordinated (**UTC**). The basic unit of **UTC** is by definition one SI second and thus does not take into account deviations of the earth rotation. Compensation is done by adding leap seconds from time to time resulting in a discontinuation of **UTC** during a leap second. **UT1** defines a continuous time standard but it requires some prediction model for future times. Reference clocks do distribute **UTC** rather than **UT1** and tag leap seconds as such. In order to avoid an accumulation of pointing offsets due to long-term drifts Observatories need to fetch upcoming leap second events and include the predicted **UTC-UT1** deviations.

The quality of a time reference is important to obtain accurate pointing of the telescope (**a**) as well as time tagging or synchronization of measurements in data acquisition back ends (**b**).

(a) The pointing accuracy requirement depends on the angular resolution aimed for, which, at least at long wavelengths, is normally limited by the diffraction beam size of the telescope. Typically, a tenth of a beam is regarded as sufficient pointing accuracy. For submm- and FIR- applications, smallest beams are typically around 10 arcsec; hence, pointing accuracy of about 1 arcsec is sufficient.

With regard to timing accuracy, pointing accuracy is basically a matter of how fast the sky is moving relative to the focal plane coordinates. For normal tracking observing modes a timing error of 1 second will contribute a position offset of 15 arcsecs at maximum. Hence, timing accuracy of below 0.1 second is required in order to achieve a pointing of better than 1 arcsec. Timing might be more crucial for (faster) scanning observing modes such as the On-The-Fly mode. Here, the scanning velocity in combination with the data acquisition/pre-integration time may put an additional constraint on the timing accuracy. The Nyquist sampling theorem implies that individual integrations along the on-the-fly scan should cover about a quarter or a third of a diffraction beam size. A "worst case" scenario of a strong line in an extended source (a typical OTF observing scenario) shows, that for heterodyne instruments at typical spectral resolutions, typical per point integration times at present sensitivities are

still at least a few seconds or more. Thus typical slew speeds will be on the order of up to a few arcsecs/sec maximum, so that synchronization of the timing between the telescope and data acquisition within 0.1 sec or so is in fact fully sufficient. The only exception would be for broad-band continuum pointing scans, which might allow somewhat faster scanning.

(b) The basic data sampling interval of the Acousto-Optical-Spectrometers (AOS) is (slightly shorter than) 10 ms, with a typical integration time per subscan of the order of a few seconds up to several minutes. The actual integration time is chosen depending on the observing mode and overall system stability.

Note: the AOS data acquisition DSP generates its own time pulses and provide them to the external hardware such as the subreflector chopping unit. There is no need to synchronize the backend computer with the DSP itself to an absolute time, but it is necessary to calibrate the DSP cycle length because integration times requested by the user needs to be transformed to number of DSP cycles.

The NTP protocol

(The following section is based on the NTP FAQ from Ulrich Windl)

NTP stands for Network Time Protocol, and it is an Internet standard protocol used to synchronize the clocks of computers to some time reference. Such a reference clock provides **NTP** with a *true time* - that is usually **UTC**. All clocks under the control of **NTP** are set towards that true time. This determination is done by a series of measurements toward a number of reference **NTP** servers (including corrections for all kind of systematic errors) leading to an estimate of that true time. The protocol is optimized for network usage. Each **NTP** daemon can act as a client of a higher level reference clock and provide the time estimate to other lower level systems. From a set of available clocks it will select the best candidate to synchronize to. A local clock (which is not a build-in RTC, but an interrupt based kernel clock) will be disciplined by the daemon by stepping or slewing the time. Stepping is used if the time is found to be off more than 128 ms (but not more than 20 minutes) and adjusting the clock frequency is done by rates of +/-0.5 ms per second. Once an offset of the local clock to the reference is detected by the statistical analysis of the **NTP**-kernel PLL/FLL it can normally be compensated within some minutes. The main problem is to detect time offset of the local clock e.g. which do occasionally occurring due to missed timer interrupts during high loads on a system. 1024 s is the default polling interval, but times down to 64 s are also often found. **NTP** is permanently optimizing this time as suggested by an Allan intercept analysis of noise and drifts. The aimed theoretical average time accuracy is +/- 5 ms.

An even faster response can be obtained with an external periodic Pulse-Per-Second signal (PPS). Linux and **NTP** can be configured to process PPS signals. Detection of pulses can easily be done with standard serial port drivers or some signal edge detection hardware. It should be obvious that a 1 s interval will be more sensitive to sporadic time offsets than the default 1024 s polling interval but tests on how PPS behaves in real live have not been done yet.

Note:GPS or atomic reference clock usually provide PPS. DCF77 is a terrestrial PPS signal with time and date encoded into the pulse length.

Tests

Surveys of Internet **NTP** servers showed that offsets range from 5 ms to 100 ms. To verify this we have run tests at the **KOSMA** observatory. The setup used at **KOSMA** consists of a AMD 166 MHz class LinuxPC connected to a HAMEG GPS receiver over a HOPF DCF77 ISA card. Additionally, two backup reference clocks at PTB/Germany are polled over a ISDN Internet connection. Typical delays of the remote clocks are 70 ms with a jitter of less than 0.5 ms. It normally takes up to 5 minutes to validate a reference clock and reach a precision of the local clock of less than 10 ms, but it is possible to reduce this initialization time to about 30 seconds (*iburst* mode). As long as GPS is running for some hours (GPS itself needs some hours to get a good time estimate) **NTP** prefers the GPS (due to the lower jitter). The offsets between the clocks are usually less than 5 ms (see figure below). However, the tests also showed daily variations and casual offsets up to 100 ms over some 10 minutes. The limiting factors here are clearly originating from the PC hardware and not from **NTP** itself. Daily and other occasionally deviations do usually arise from changes of the room temperature causing the quartz oscillator to drift in frequency (expressed in PPM). The PC in the test setup had a frequency error of 4 PPM. More expensive hardware may perform better. Also faster CPUs generally give better results by reducing the overall latency.

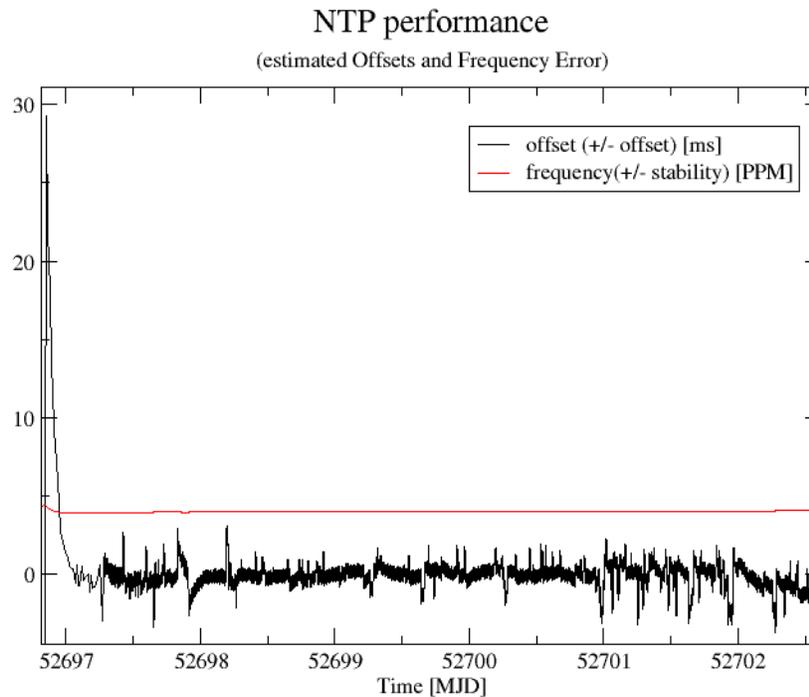


Figure 1 NTP accuracy and stability for several days after starting up

Summary

NTP is a very flexible and powerful package to control the local clock within accuracies of generally better than 100 ms with average deviations of less than 5 ms. Clock disciplining is very transparent for user level applications which do not have to use special libraries or depend on additional hardware

components. Even under worst conditions offsets do not grow larger than 100 ms and therefore allow astronomical applications to work within the required accuracy, but it cannot overcome intrinsic hardware limitations.