

Measurement of Time Servers

Vladimír Smotlacha

December 10, 2001

1 Abstract

This report describes two methods of objective measurement of computer clock offset. The first high accurate method requires external reference clock and a two channel counter, while the second method uses only external reference clock. Both methods are demonstrated at measurement of the offset of NTP server.

2 Keywords

time offset measurement, computer clock, NTP

3 Introduction

Clock in computer is virtual object. It does not exist until a program asks "What's the time?". From user viewpoint, it is represented by a function (e.g. `gettimeofday()` or `ntp_gettime()`) which returns current time. To measure offset of this clock, we need either to retrieve the clock signal outside of the box or to carry external clock into the box. We implemented both these methods. Although both methods are demonstrated at offset measurement of NTP servers, they are universal and do not depend on NTP system.

The second goal of this report is to describe several measurements of NTP servers which were done in order to estimate the accuracy reachable in practice. We directed our attention at the offset of stratum-1 and stratum-2 servers. PC with Linux (RedHat distribution, kernels 2.2.18/2.2.19 with nanokernel patch), NTP v. 4.0.99m-rc2 or CVS interleaved versions from April 2001 were used for testing. Standard quartz oscillator (14.31818 MHz) without any temperature compensation was used in all PCs. All measurements were performed on a production network under normal load - local network of Cesnet and the Prague Academic Network (PASNET).

The report is organized as follows: Sections 2. and 3. describe both methods. Section 4. discusses interrupt latency in PC. Section 5. contains results of absolute offset measurement of the primary NTP server and in Section 6. is offset measurement of secondary NTP server.

In the report is used terminology (e.g. offset, skew, drift, accurate, precision) according to RFC-1305 [1]. In the following text, 'us' stays for a 'microsecond'.

4 Method I

Internal clock from computer is retrieved by a specially designed software pulse generator. Heaving these pulses (PPS - Pulse Per Second), we can compare them to another accurate PPS signal (e.g. cesium clock, GPS receiver). For this measurement, two channel counter is needed. We observe the difference between both signals. As time of each generated pulse is known and logged, it is possible to compensate the drift involved by pulse generator.

The algorithm of pulse generator is as follows:

```
long d0 = 5e8;
/* 5*10^8 ns = 0.5 s pulse offset
   from beginning of each second */
long d1 = 1e8;
/* 10^8 bs = 100 ms length of pulse */
long t1,t0;

do {
    waituntil(d0);
    t0 = gettime();
    edge(1);
    t1 = gettime();
    delay(d1);
    edge(0);
    print(t0, t1-t0)
}
```

In practical realization, the output is logged as a triple (sec, off, lng), where

- sec is the current second (modulo day),
- off is the pulse offset from beginning of second, and
- lng is the duration of pulse edge retrieving (i.e. the upper limit of uncertainty of pulse edge).

The base drift of the pulse generator could not be better than clock precision and resolution. The resolution is clock increment step and it is done by hardware (e.g. PC has resolution about 1 us as the clock is derived from 1.193 MHz). Precision is the unit in which is time reported. In Unix like systems it used to be 1 us, while nanokernel (Linux, FreeBSD) it changed down to 1 ns. In practical realization, the drift is done by processing time of `gettime()`, which depends on CPU clock rate and is about 1.6 us for Pentium III / 500 MHz.

The item off gives us the opportunity to compensate the drift of the generator drift.

5 Method II

This method assumes that computer is provided by PPS signal. We created a simple program based on PPS API [2] for reading and saving timestamps of this PPS. Assuming that PPS signal is accurate and that offset does not exceeds 0.5 s, the saved fraction part of timestamp is just the offset of internal clock. Accuracy of this method depends on interrupt latency delay and variation as is discussed in next section. The average latency could be subtracted from saved timestamps. As interrupt latency in particular case can exceed the average latency by several us or even several tenth of us, such sample has to be omitted. Therefore a simple filtering algorithm (sample exceeding the average of previous 3 samples by 5 us or more is omitted) is applied. The accuracy of this method can not be better than one or several us. This method is quite good and valuable when clock is synchronized by a network (e.g. by NTP server).

6 PPS processing latency

The standard way to discipline a primary NTP server is to use an external device providing one pulse each second, which is called PPS (pulse per second). This pulse is connected to DCD pin of a serial line interface. Change of the DCD signal initiates an interrupt and in the interrupt routine timestamp to the pulse is assigned.

In this chapter we are not interested in PPS delay outside of the computer (cable delay, voltage level converter delay, ...) as it is constant and can be compensated. More important is delay in the computer, i.e. the time from the pulse edge until a timestamp is assigned.

Linux implementation of NTP (PPSkit) contains code for a PPS software echo. The idea is quite simple: a change of DCD signal (PPS edge) can be echoed in interrupt routine to RTS signal output after getting the PPS timestamp. Using

some external device like a two channel oscilloscope or a counter, we can measure the difference between the original signal and its echo. In this way we can find out how much time the PPS processing takes and we can estimate the delay and jitter between the real PPS signal and its timestamps.

6.1 PPS check

In order to exclude influence of possible PPS irregularities (manufacturer specifies the PPS accuracy as 1 us), the PPS interval was measured as well (in principle only each second interval could be measured). Root of Allan variance of PPS signal from the GPS receiver used (Garmin GPS 35) was calculated as 43 nanoseconds. The distribution is shown on the following graph.

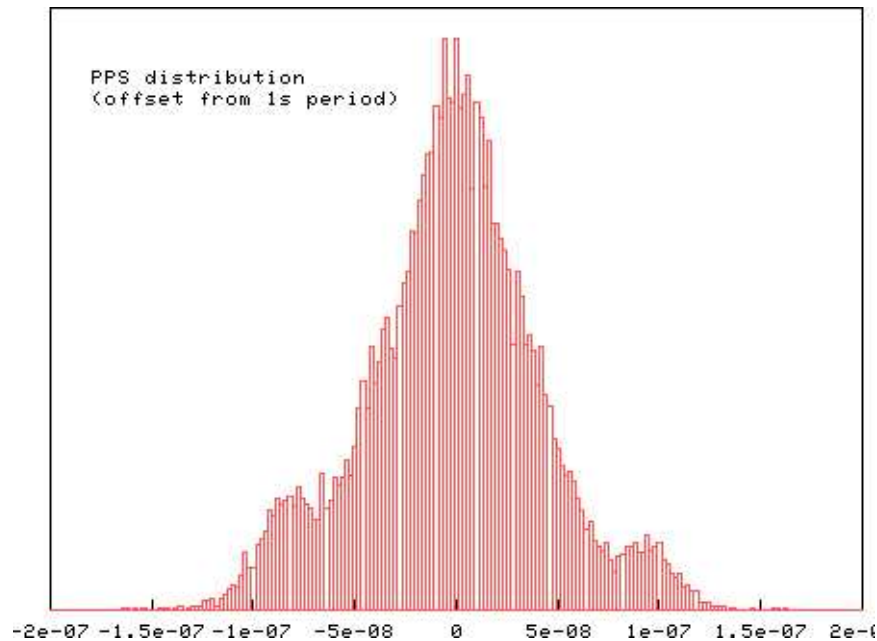


Figure 1: Distribution of PPS signal

6.2 Delay in normal operation mode

Following graphs are results of a PPS echo delay measurement in normal operation of a primary NTP server (with some 120 clients). The first PC used is a standard Pentium 150MHz (UDMA disk, kernel 2.2.17, NTP 4.0.99k), the second one is DELL 1400 (PentiumIII 860MHz, SCSI disk, kernel 2.2.18, NTP 4.0.99k36). The source of PPS signal was Garmin GPS35 receiver. Universal counter SR 620 (Stanford Research Systems) was used to measure the delay.

Calculated statistics (in microseconds):

	mean delay	root Allan variance	standard deviation
P150	11.02	0.83	0.95
P860	8.31	0.34	0.36

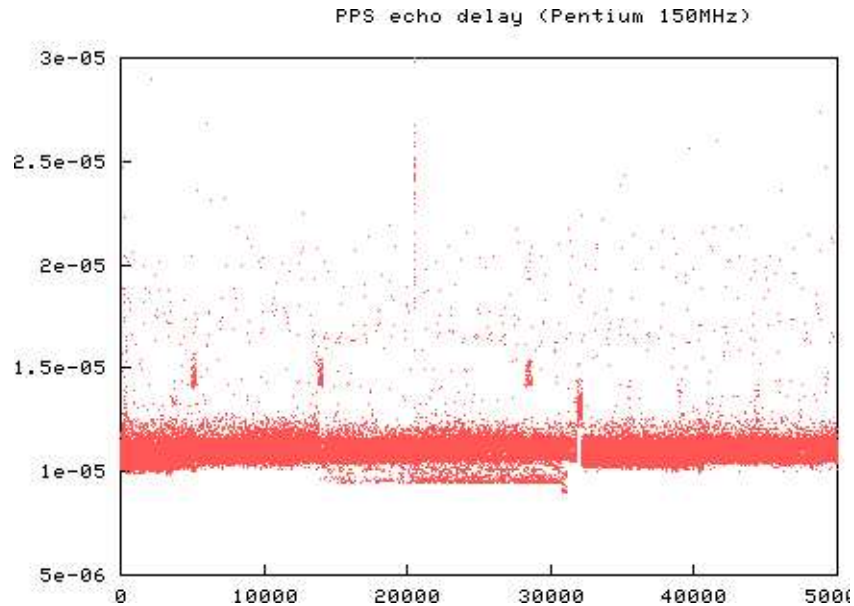


Figure 2: PPS echo delay (P 150MHz)

6.3 Influence of the load

Next graph shows the influence of computer load on PPS echo delay (P150) events:

- 0 - 1000s: kernel compilation - delay increases by 15 us
- 1200 - 1350s: extra load by a CPU intensive large process (very long endless loop - 500000 NOPs) - delay increases by 7 us
- 1600 - 1700s: extra load by disk reading (dd reading of 10^9 bytes) - delay increases by 8 to 30 us
- short CPU intensive processes (short endless loop) had no influence on echo delay.

6.4 Recommendation for design of PPS disciplined NTP server

From previous results we can conclude this recommendation for building primary NTP servers:

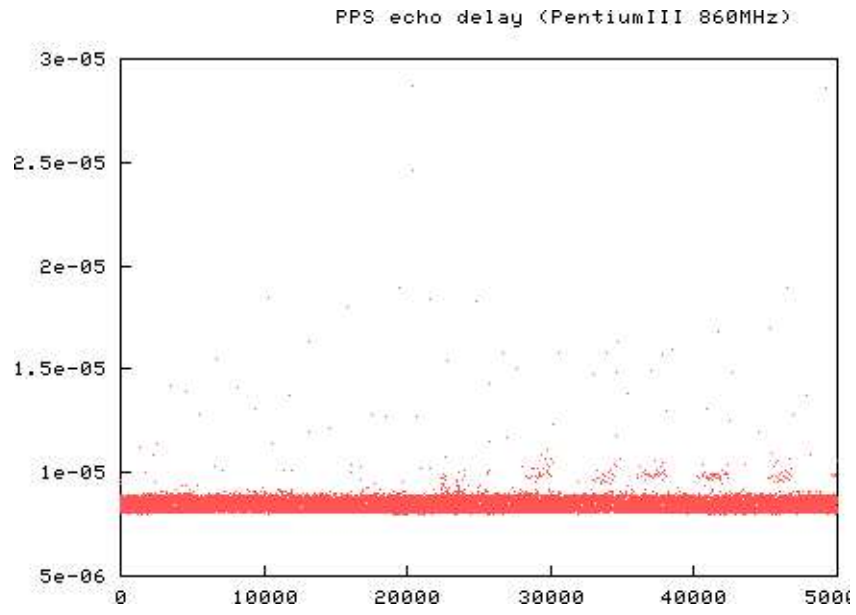


Figure 3: PPS echo delay (PIII 860MHz)

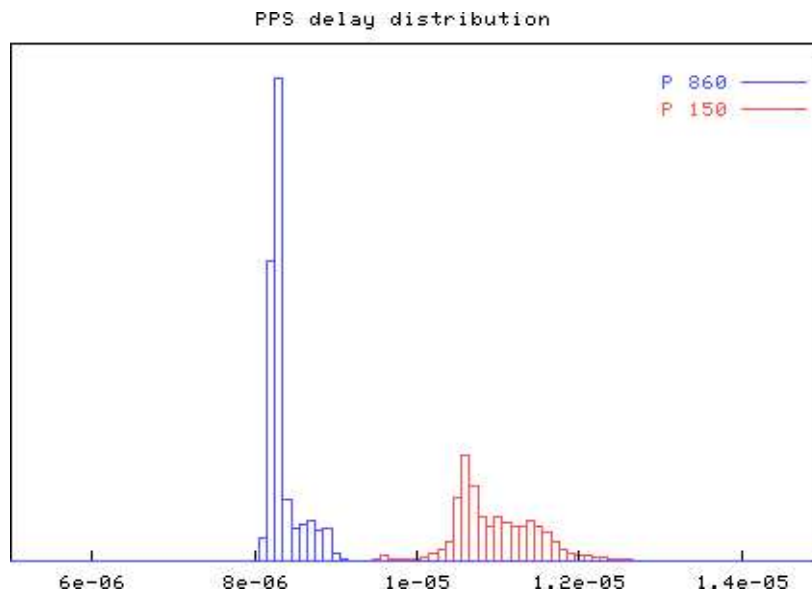


Figure 4: PPS echo distribution

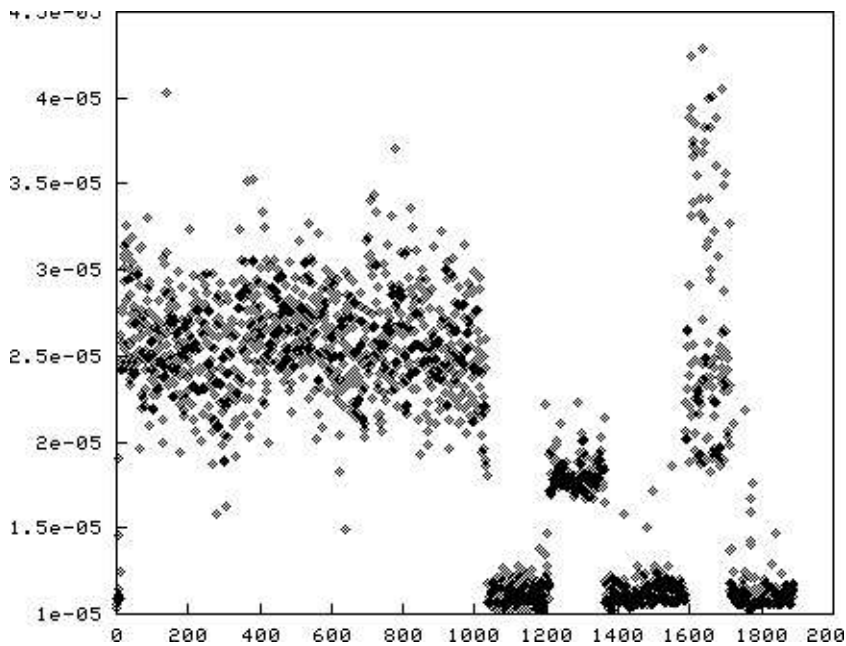


Figure 5: PPS echo delay - loaded box (P 150MHz)

- a CPU speed is important but not essential. Older Pentium PC is good enough
- disk should be UltraDMA or SCSI; using DMA is essential (switching off DMA increased the delay by more then 100us)
- it is necessary to avoid disk intensive operations or running large CPU intensive processes (cache memory should not be overwrite too often)

7 Measuring of the absolute accuracy

An important topic of this report was the measurement of the absolute accuracy of stratum-1 NTP server disciplined by GPS receiver. The Czech National Time and Frequency Standard was used as a reference clock.

An essential goal of a primary NTP server is to synchronize the internal clock as close as possible to an external clock. Measuring of the offset between these two clocks is not simple because only the later one has real representation but the internal clock of the NTP server is realized by software and the only one way to access it is to call a routine which returns actual time. We used method I, which was described in Section 2.

Offset was measured by a universal counter SR 620 (Stanford Research Systems). NTP server was implemented on DELL 1300 (PIII 500MHz), kernel 2.2.19 with

nanokernel patch and NTP 4.0.99m-rc2 disciplined by the Motorola UT+ receiver. No delay compensation was applied. As this measurement is not yet finished, following graphs show raw difference between output signal and reference clock (National Standard) and the same difference after compensating drift of the pulse generator.

In the observed interval (more than 100 minutes) the absolute offset of the internal clock was between 8 and 16 μs , mean value 12.11 μs and standard deviation 1.35 μs . The result (output from counter) is shown at Figure 6, while Figure 7 displays the same data after the compensation of pulse generator drift.

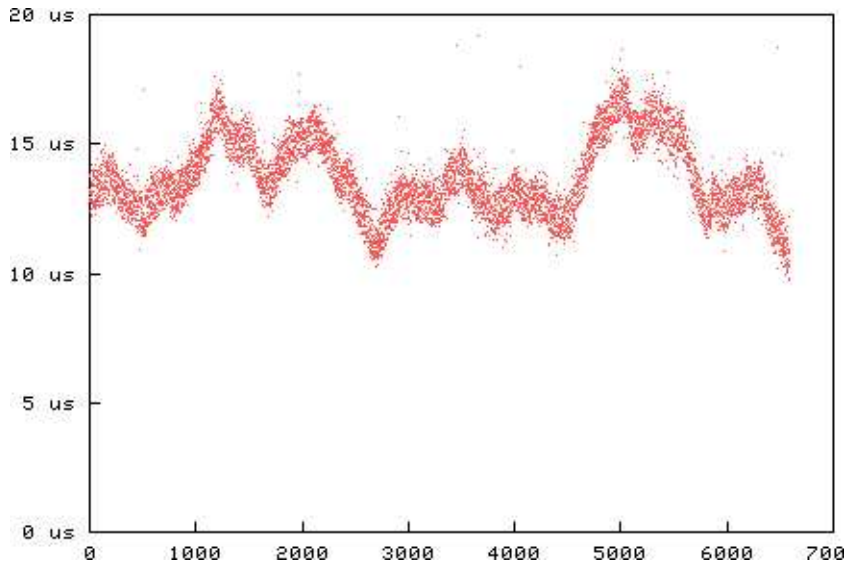


Figure 6: Absolute offset

7.1 Delays discussion

Once we know real absolute offset of the internal clock in primary NTP servers, it is important to describe all influencing factors.

a) input delay (PPS from GPS receiver)

- cable delay: about 1 μs / 200 m
- voltage converter: < 1 μs
- delay in computer was discussed in part 2

The biggest influence come from the third factor; it also brings jitter which is at least one order of magnitude higher than primary jitter from an inexpensive GPS receiver.

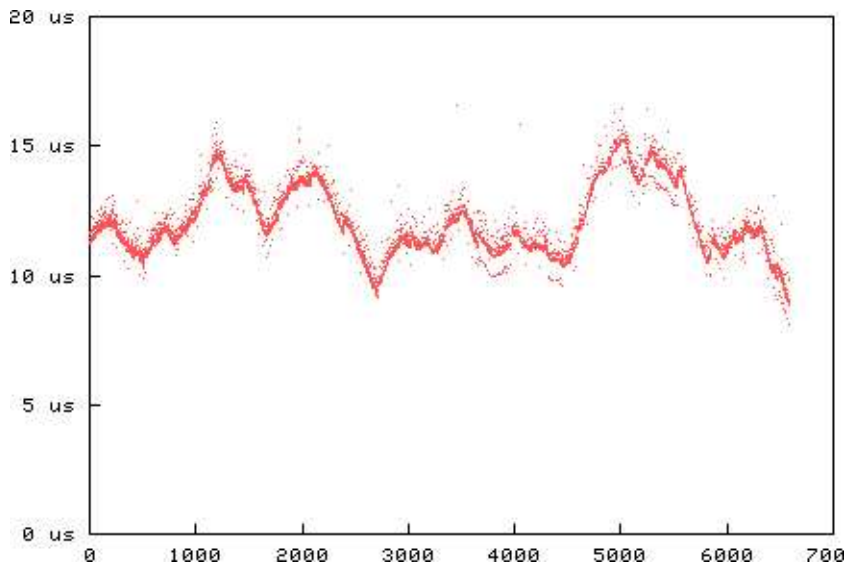


Figure 7: Compensated absolute offset

b) output delay (pulse generator)
similar delays are present in output signal

- output circuit delay
- cable delay
- internal error of software pulse generator

8 Stratum-2 server measurement

The goal was to validate that accuracy of stratum-2 could be better than 1 millisecond and to find conditions for the best accuracy. Reliable millisecond or better resolution is necessary for WAN one-way delay measurement in real networks, where it is not possible to discipline all testing workstation by an external clock source. The essential problem of NTP protocol is how to detect or even compensate asymmetry in packet propagation delay between a client and a server.

The same NTP primary server (disciplined by GPS Garmin) was used in all following tests. Stratum-2 servers were configured to use only this one primary server with 'maxpoll' factor set to 4 (which means to ask primary server once per 16 seconds).

Stratum-2 servers were installed at:

- adjacent LAN (one router between boxes), ping 0.3-0.5 ms
- more distant LAN (3 routers, 2 switches, LANE, MPLS), ping 1.9-3.5 ms with spikes above 15 ms
- WAN (metropolitan network with ATM backbone, load balancing, 4 routers, about 15 km air distance), ping 2.1-3.0 ms

8.1 Measurement results

Figures 8 and 9 show offset of server connected to two different LANs: an adjacent LAN and a remote LAN (with the worse connectivity I found in my organization). The mean offset in the first case is 26.5 us, standard deviation 6.2 us, maximum offset 51.4 us and minimum offset 6.8 us. For the second case mean value is -178.5 us, standard deviation 15.5 us, minimum -252 us, maximum -128 us. Clearly there is an asymmetry in routing between these two LANs.

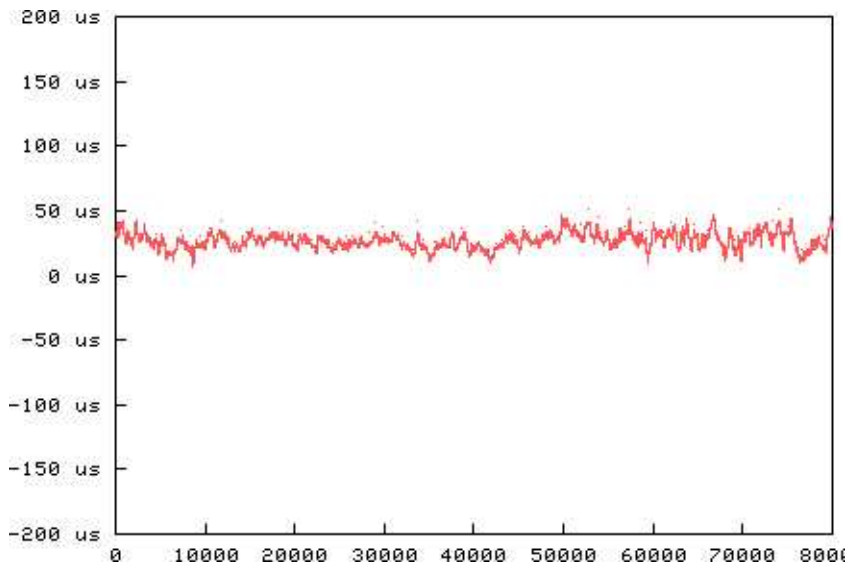


Figure 8: Offset of Stratum-2 server (LAN 1)

Figure 10 shows offset of NTP client connected via WAN (Metropolitan network). Mean offset is -41.7 us, standard deviation 21.8 us, minimum offset value -141.4 us and maximum offset value 38.2 us.

9 Conclusion

Both introduced methods were approved by practical implementation.

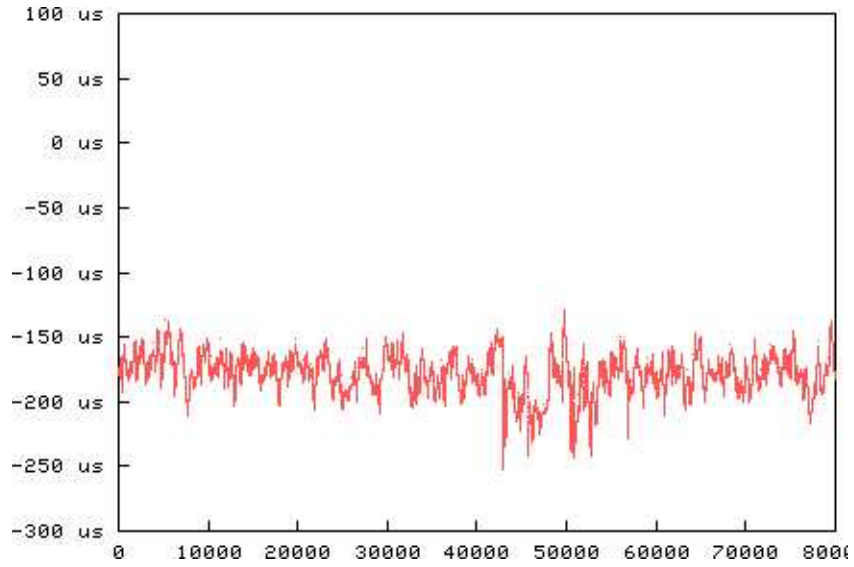


Figure 9: Offset of Stratum-2 server (LAN 2)

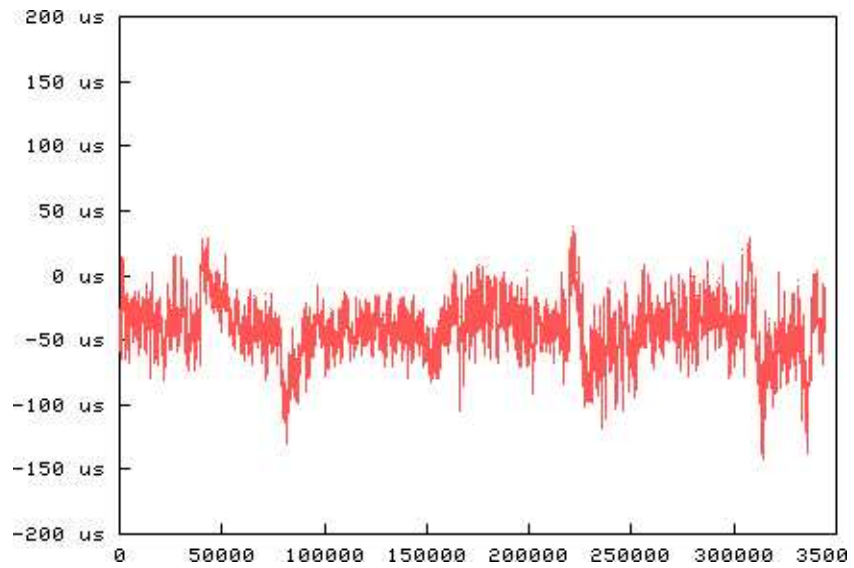


Figure 10: Offset of Stratum-2 server (WAN)

The performed measurements proved that building primary NTP server with absolute accuracy of internal clock far better than 10 us is possible using standard PC and low-cost GPS receiver. For secondary (stratum-2) NTP servers we proved that, if connected via LAN or good WAN, the accuracy not worse than 200 us can be reached. However it depends on network characteristic. When using such a NTP server as timebase with the best available accuracy, every installation should be tested separately.

10 Credits

The measurement of absolute accuracy would not be possible without the cooperation with the Standard Time and Frequency Department of the Institute of Radio Engineering and Electronics (the Czech Academy for Science). The main credit belongs to Mr. Jan Cermak, head of this department. We would like to thank to Dr. Jaroslav Roztocil from Czech Technical University, who lent us the counter which we used for interrupt latency measurement.

11 References

- [1] D. L. Mills. Network Time Protocol Specification. RFC-1305, IETF, March 1992.
- [2] J. Mogul, D. L. Mills, J. Brittonson, J. Stone and U. Windl. Pulse-per-second API for Unix-like operating systems, version 1. RFC-2783, Internet Engineering Task Force, March 2000.