Metrologia 45 (2008) S51–S58 doi:10.1088/0026-1394/45/6/S09

On calibration of network time services

Vladimir Smotlacha[1], **Jan Čermák**[2] **and Juan Palacio**[3]

[1] CESNET z.s.p.o., Zikova 4, 160 00 Prague 6, Czech Republic
[2] Institute of Photonics and Electronics, Academy of Sciences of the Czech Republic, Chaberská 57, 182 51 Prague 8, Czech Republic
[3] Real Instituto y Observatorio de la Armada, Cecilio Pujazón, s/n, 11100 San Fernando, Cadiz, Spain

Received 16 July 2008
Published 5 December 2008
Online at stacks.iop.org/Met/45/S51

Abstract
This paper proposes methods for calibration of two network time services—NTP servers and time-stamp authorities. The calibration is described in conformity with metrological principles like other time distribution systems. The authors have built up the calibration sets and tested them in cooperation with five European time and frequency laboratories. The paper also presents and discusses experimental results collected by measurement of time servers between involved laboratories.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Recently there has been discussion about the need for calibration of the network time services which are currently provided by NTP servers and time-stamp appliances (TSA). Initially, some argued that this type of calibration would be out of the scope of time and frequency (TF) metrology because the time information is transmitted between computers and is hidden in the protocol layer. The argument in favour of the involvement of TF metrology rests on the fact that these services claim to provide Coordinated Universal Time (UTC) and that this claim cannot be relied on unless verified by a calibration based as much as possible on standard metrology procedures.

Both services have in common that they are based on UTC-synchronized time servers but the use of the time information they offer is quite different. The task of the NTP service [1] is to synchronize the client computer. To do so the service makes use of a two-way time transfer in which also the client time information is needed. Thus to measure the NTP synchronization performance, the measurement device is operated at the client site and the 'device under test' is not only the NTP server itself but also the complex two-way time transfer system that includes the transfer links and the client computer. In contrast, the TSA service [2] functions only as a check clock that provides time stamps on the client's electronic requests sent to TSA. Hence there is no need for the client time information and the 'device under test' is the TSA itself with an associated access point to which the timing uncertainty is referred.

In either case, the calibration methods must accommodate the nature of the communication between computers and therefore the measurement device must also be a computer. We call this a calibration computer (CC). In the following, we discuss two approaches, active and passive, that can be used for calibration of both services. In collaboration with some European TF laboratories, we have recently carried out a series of measurements with the CCs in various network constellations.

2. Calibration computer

The CC offers an effective platform for the calibration software running on an open-source operating system. As for the timing capability, the CC produces a UTC-synchronized software-controlled time scale $T(\mathrm{C}) = \mathrm{UTC} \pm u_\mathrm{C}$ which allows the provision of time stamps to the query and reply with uncertainties $u_\mathrm{Q} > u_\mathrm{C}$ and $u_\mathrm{R} > u_\mathrm{C}$, respectively, against UTC. The uncertainties u_Q and u_R are not equal because of the different delays in software processing of the incoming query message and outgoing reply message. Synchronization to UTC is ensured through a 1 pps signal from a calibrated GPS receiver incorporated in the CC system. The $T(\mathrm{C})$ time scale is physically represented by software-generated 1 pps ticks through which the uncertainty u_C can be verified against

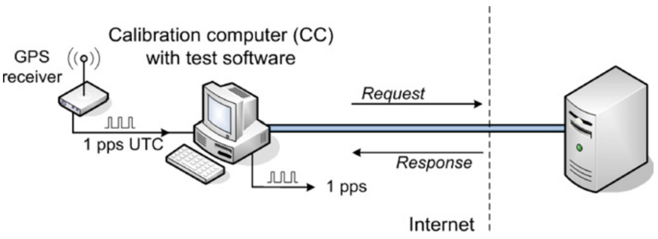

Figure 1. Active calibration of NTP.

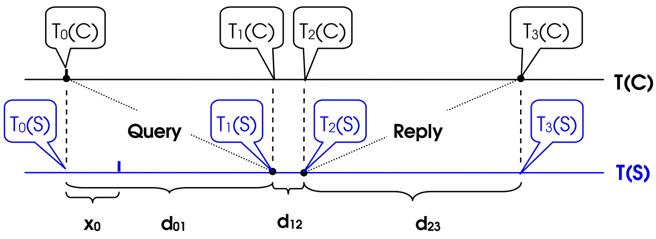

Figure 2. NTP time transfer.

the synchronization pulses. The setup of the calibration system is shown in figure 1.

3. Calibration of NTP service

3.1. Basic considerations

The synchronization accuracy of the NTP service depends essentially on to what extent the differential delay between the one-way (send/receive) delays approaches zero [3]. Each one-way delay is a sum of corresponding per-link and per-node delays. The per-link delay is composed of the propagation delay $d_P = L/(cw)$, where L is the link length, c is the speed of light and w is the velocity factor, and the serialization delay $d_S = S/C$, where S (bit) is the packet size and C (bit/s) is the link capacity. The per-node delay consists of the forwarding delay (reading the information and making forwarding decisions) and queuing delay (waiting in the queue to output the data from the node). Thus the synchronization accuracy is dependent on the network access point to which the client (or the CC as its substitute) is connected and also on the Internet connectivity and the epoch of the transfer. It turns out that the per-link delays can be assumed about equal in both directions but this does not apply to per-node delays. In some cases, the Internet routing protocol selects asymmetrical paths between two nodes which results in a large differential delay.

To describe the transfer, assume that the NTP server time scale $T(\mathrm{S})$ evolves with respect to $T(\mathrm{C})$ as

$$[T(\mathrm{S})](t) = [T(\mathrm{C})](t) - x_0 - \int_{t_0}^{t} y(\theta)\,\mathrm{d}\theta, \qquad (1)$$

where x_0 is the initial time scale difference (denoted as time offset, θ_0, in the NTP notation) at $T_0(\mathrm{C}) = [T(\mathrm{C})](t_0)$ and the integral represents the time difference accumulated due to relative frequency deviation $y(t)$ between $T(\mathrm{C})$ and $T(\mathrm{S})$. In the above convention, the reading $T_0(\mathrm{S})$ is smaller than $T_0(\mathrm{C})$ for $x_0 > 0$, i.e. $T(\mathrm{S})$ will be delayed by x_0 against $T(\mathrm{C})$ at t_0. For $y > 0$ the readings $T(\mathrm{S})$ will increase more slowly than $T(\mathrm{C})$. Presume further that the time quantity t in (1) is represented by $T(\mathrm{C})$, i.e. $t = [T(\mathrm{C})](t)$.

The NTP time transfer is done in two steps as shown in figure 2.

Step 1. A query is sent from CC at $T_0(\mathrm{C})$ and is time stamped by the server as

$$T_1(\mathrm{S}) = T_0(\mathrm{C}) - x_0 + d_{01}(1 - y) \qquad (2)$$

assuming that $y(t) = y =$ constant for a short period of time transfer and d_{01} is the delay it takes for the query to be referred to $T(\mathrm{S})$.

Step 2. A reply is sent from the server at $T_2(\mathrm{S})$ and referred to $T(\mathrm{C})$ by CC as

$$T_3(\mathrm{C}) = T_2(\mathrm{S}) + x_0 + d_{23}(1 + y), \qquad (3)$$

where d_{23} is the delay it takes for the reply to be referred to $T(\mathrm{C})$.

The delay due to the reaction time of the server is given by $d_{12} = [T_2(\mathrm{S}) - T_1(\mathrm{S})]/(1 - y)$. Combining (2) and (3) yields

$$x_0 = \tfrac{1}{2}\{[T_3(\mathrm{C}) - T_1(\mathrm{S})] - [T_2(\mathrm{S}) - T_0(\mathrm{C})]\} + \varepsilon, \qquad (4)$$

where $T_0(\mathrm{C})$, $T_3(\mathrm{C})$, $T_1(\mathrm{S})$ and $T_2(\mathrm{S})$ are the readings obtained from the transfer and ε is the inherent synchronization error

$$\varepsilon = \tfrac{1}{2}[d_{23} - d_{01} + y(d_{01} + d_{23})]. \qquad (5)$$

The sum of delays taken by both the query and reply is

$$d = d_{01} + d_{23} = [T_3(\mathrm{C}) - T_0(\mathrm{C})] - (1 - y)[T_2(\mathrm{S}) - T_1(\mathrm{S})]. \qquad (6)$$

For $y = 0$ we obtain the delay δ introduced by NTP [1]

$$\delta = [T_3(\mathrm{C}) - T_0(\mathrm{C})] - [T_2(\mathrm{S}) - T_1(\mathrm{S})]. \qquad (7)$$

Apparently, the component delays d_{01} and d_{23} cannot be determined from the transfer. If the contribution of $y(d_{01} + d_{23})$ to ε is negligible, we can estimate the error bounds in terms of $T_0(\mathrm{C})$, $T_1(\mathrm{S})$, $T_2(\mathrm{S})$ and $T_3(\mathrm{C})$ as

$$|\varepsilon_{\max}| < \tfrac{1}{2}\delta. \qquad (8)$$

In the real time transfer we have typically $d_{01} \sim d_{23}$ and $y \ll 1$ so that $|\varepsilon| \ll |\varepsilon_{\max}|$. In ideally symmetrical transfer where $d_{01} = d_{23} = d$, the delays in (5) cancel out giving $\varepsilon = yd$. For $y = 0$ and immediate reaction, i.e. $T_1(\mathrm{S}) = T_2(\mathrm{S})$, we obtain the common convention for the readings of two synchronized clocks

$$T_1(\mathrm{S}) = \tfrac{1}{2}[T_0(\mathrm{C}) + T_3(\mathrm{C})]. \qquad (9)$$

3.2. Implementation

Our CC system is based on PC compatible hardware equipped with a special card that makes it possible to capture the 1 pps synchronizing ticks with reduced interrupt latency (~50 ns). We make use of a mini-ITX VIA-EPIA motherboard with 1 GHz CPU and we also employ a Tedia PCT-7424 capture

Table 1. Summary of NTP time offset measurement (in microseconds).

		CESNET	BEV	INRIM	PTB	ROA	IPE
CESNET	Mean		8966	−99	326	−254	−2
	Median		8771	−6	281	27	−2
	StDev		1825	1350	281	2472	2
INRIM	Mean	124	5034		470	−135	123
	Median	45	5181		343	65	40
	StDev	1259	1537		1244	2817	1321
ROA	Mean	183	−387	257	977		478
	Median	237	−497	79	713		98
	StDev	1874	−1559	2798	1596		1722
IPE	Mean	30	−431	−100	501	−256	
	Median	35	−837	14	497	33	
	StDev	14	1908	1521	339	2481	

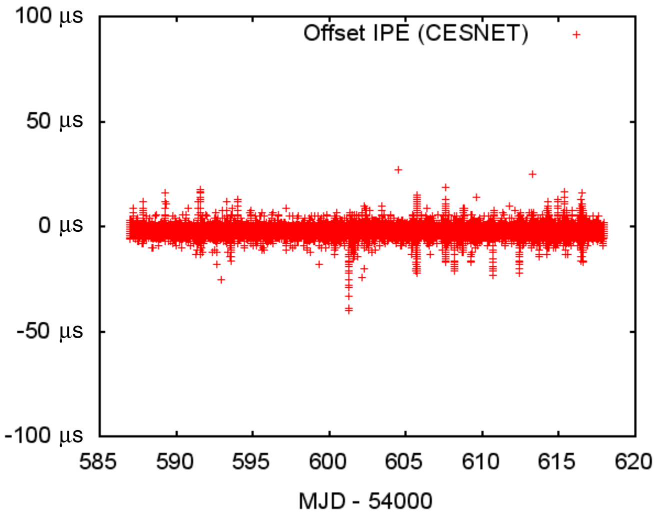

Figure 3. NTP time offset (CESNET to IPE).

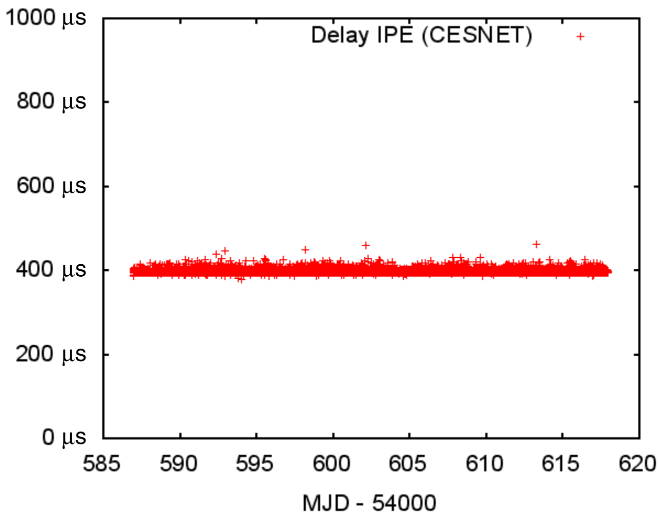

Figure 4. NTP delay (CESNET to IPE).

card [4] with firmware produced according to our design. The card device driver also includes the 1 pps processing routine specified in RFC 2783 [5]. The optional OCXO module contains a 10 MHz ovenized oscillator and a frequency converter (OCXO replaces the standard 14.318 MHz oscillator on the motherboard). We have installed the Debian 3.1 Linux distribution with the kernel version 2.4.33 and the nanokernel patch.

The T (C) time scale is implemented in the operating system and is accessed by the kernel function *gettimeofday()* to provide a resolution of 1 µs or *ntp_gettime()* function for a resolution of 1 ns. Synchronization of the T (C) time scale is performed by the *ntpd* routine that processes the 1 pps signal and runs a digital PLL to discipline the quartz oscillator. The T (C) internal time scale is represented by 1 pps ticks that are generated through the serial port by our *pps_gen* utility. The result of T (C) synchronization represented by the time difference 1 pps (in) −1 pps (out), see figure 1, gives a mean of 0.9 µs with a standard deviation of 1.2 µs.

The calibration is made by the *ntp_mea* utility which behaves like a NTP client. It sends out the NTP query, receives the NTP server reply and calculates the delay δ and the time offset θ_0 [1].

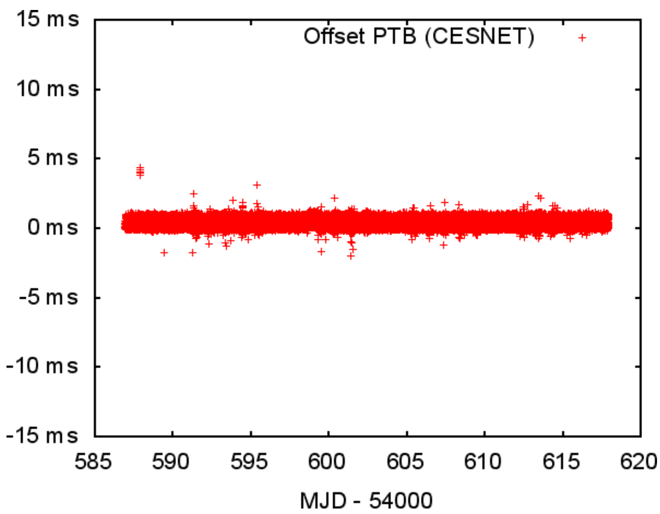

Figure 5. NTP time offset (CESNET to PTB).

3.3. Experimental results

Five European TF laboratories: BEV (Austria), INRiM (Italy), PTB (Germany), ROA (Spain) and IPE (Czech Republic) plus CESNET(Czech Republic)—an academic

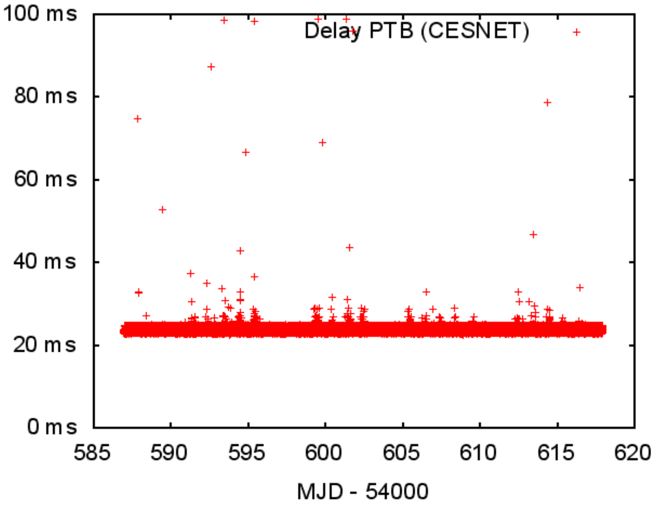

Figure 6. NTP delay (CESNET to PTB).

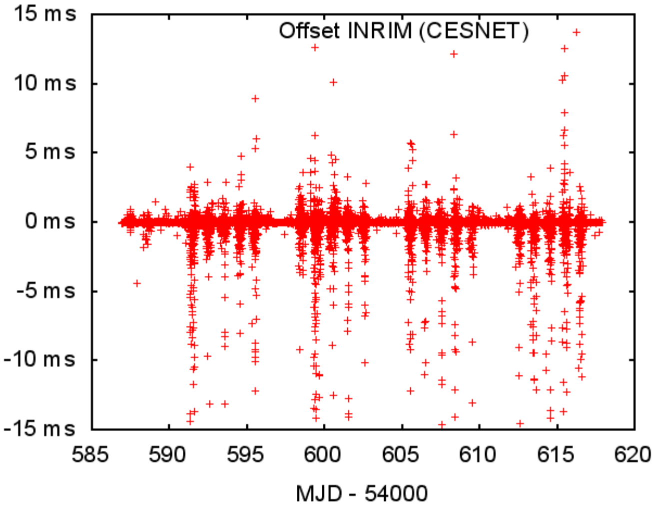

Figure 7. NTP time offset (CESNET to INRiM).

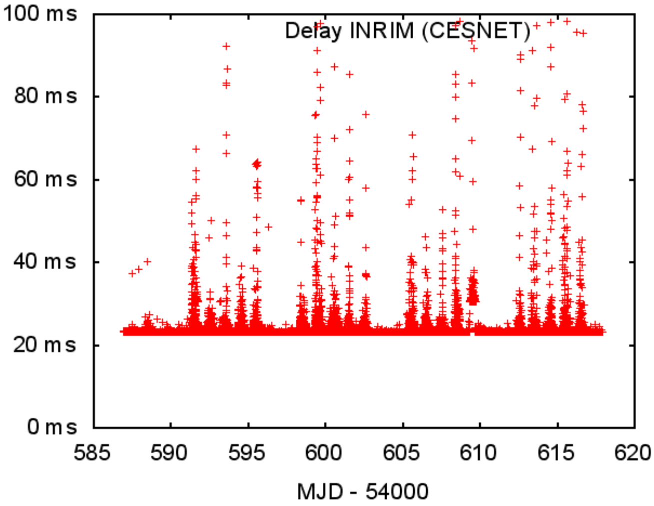

Figure 8. NTP delay (CESNET to INRiM).

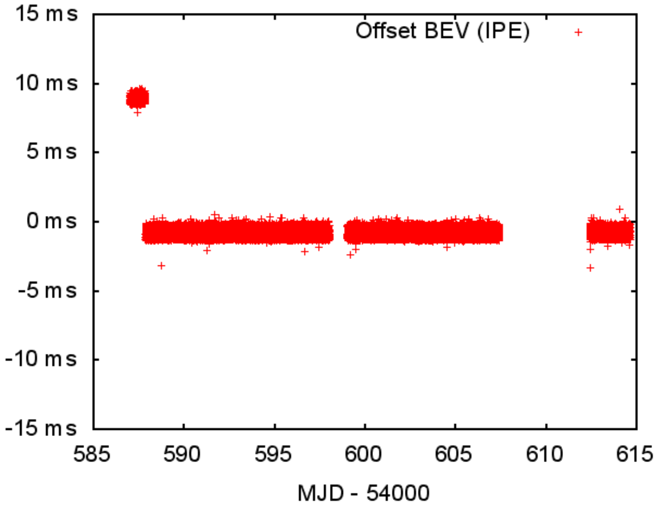

Figure 9. NTP time offset (IPE to BEV).

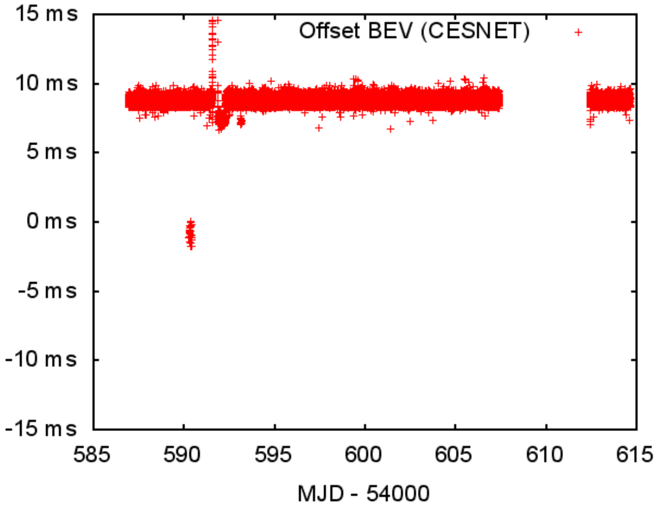

Figure 10. NTP time offset (CESNET to BEV).

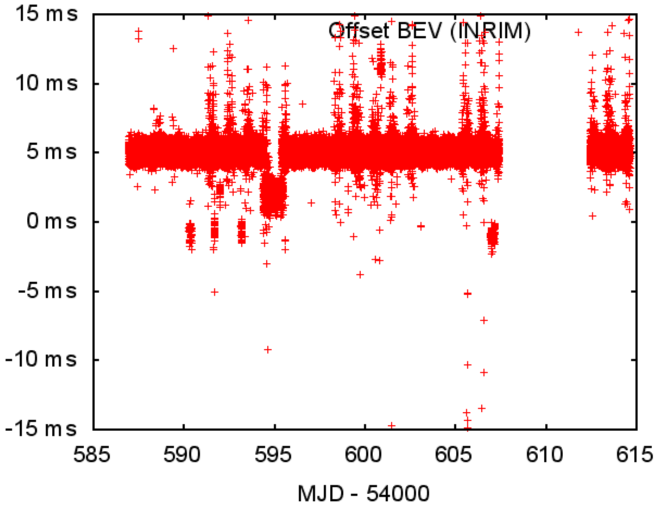

Figure 11. NTP time offset (INRiM to BEV).

network operator—took part in testing their NTP servers. Two identical experimental CCs described above were operated at CESNET and IPE, and two other CCs with the same *ntp_mea* program at ROA and INRiM. Each CC polled all NTP servers every minute. The following is a survey of the results obtained in May 2008.

3.3.1. Time offsets. Table 1 summarizes the statistics of the time offsets found by the measurement. The CCs are listed in the first column and the NTP servers in the first row.

3.3.2. Network influence. We can identify specific characteristics, depending on the Internet Service Provider (ISP) involved in the transfer.

- *Low noise, no periodic changes in delay*. This is the case for the measurement between IPE and CESNET, which are connected through the overprovisioned Prague metropolitan network with additional available bandwidth, which is never congested. Also the transfer delay is low (400 µs). The plots of the time offset and delay are shown in figures 3 and 4, respectively.
- *Large noise, small periodic changes in delay*. An example is the measurement between CESNET and PTB shown in figures 5 (time offset) and 6 (delay). The small periodic changes in delay are partly masked by background noise.
- *Large periodic changes in delay*. Large delays during rush hours of working days are symptoms of an overloaded network. An example is the measurement of INRiM by CESNET as depicted in figures 7 (time offset) and 8 (delay). We observe peaks of time offset higher than 30 ms while the typical median value is 6 µs.
- *Large jumps in delay*. Large unpredictable jumps in delay were observed between BEV and all CCs with a great impact on the time offset as illustrated in figures 9–12. Since each CC found different jumps, it is evident that the source of jumps is not the BEV server itself.

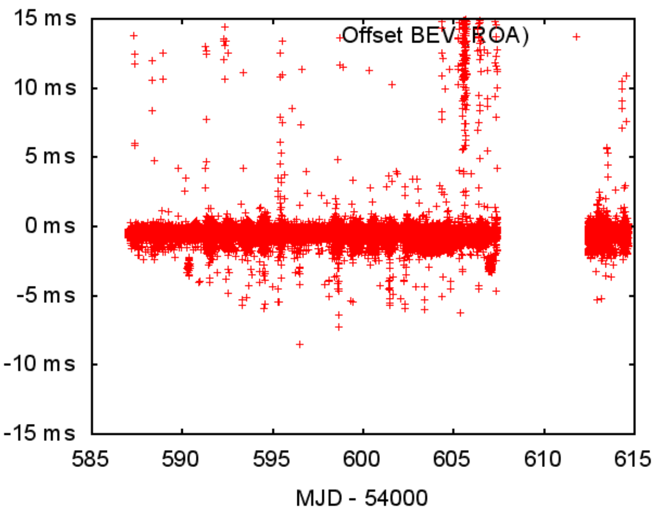

Figure 12. NTP time offset (ROA to BEV).

4. Calibration of TSA Service

4.1. Basic consideration

As hinted above, the purpose of the TSA service is to provide the time stamps in the form of cryptographically signed certificates. These time stamps are generated from the server time scale $T(\mathrm{S})$ synchronized to UTC with an uncertainty $\pm u_\mathrm{S}$. The uncertainty u_TS of the time stamps is in principle larger and practically much larger than u_S due to limited resolution of the time stamps (RFC 3161 [2] states that u_TS can be up to 1 s). Uncertainties u_S and u_TS are sometimes confused with the uncertainty u_UTC of the local source of UTC (typically $u_\mathrm{TS} \gg u_\mathrm{S} \gg u_\mathrm{UTC}$).

It should be pointed out that in contrast to the NTP service, the TSA service is not aimed at the synchronization of the client's clock. Thus the basic calibration objective is to verify the uncertainty u_TS which is associated with the access point defined by the TSA provider (typically at the server interface) and which we call provider-oriented calibration (POC). Since the POC may have some legal implications, it should be made in compliance with common metrology standards. Another type of calibration we call the client-oriented calibration (COC) is the measurement associated with the access point specified by the client. The objective of COC is to assess the TSA timing performance at the client site, i.e. including the time delay it takes for the client request for time stamp to reach the TSA.

The time-stamp protocol is based on a request message sent by a client and a response message sent by the server, denoted as time-stamp query (TSQ) and time-stamp reply (TSR), respectively, in the TSA notation. The TSQ is sent at time $T_\mathrm{Q}(\mathrm{C})$ and the TSR that contains the time stamp $T_\mathrm{S}(\mathrm{S})$ is received at time $T_\mathrm{R}(\mathrm{C})$. Our *tsa_mon* utility evaluates times $T_\mathrm{Q}(\mathrm{C})$ and $T_\mathrm{R}(\mathrm{C})$, matches corresponding TSQ and TSR and decodes the $T_\mathrm{S}(\mathrm{S})$.

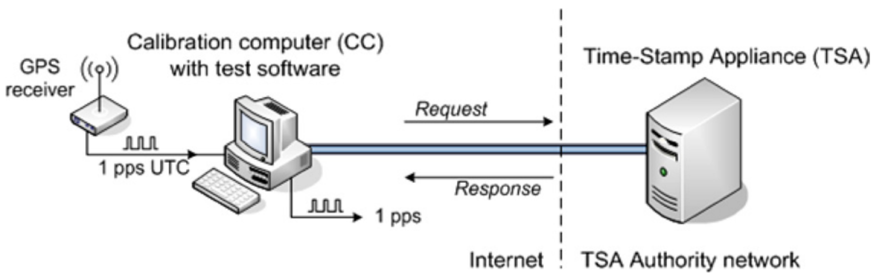

Figure 13. Active calibration of TSA.

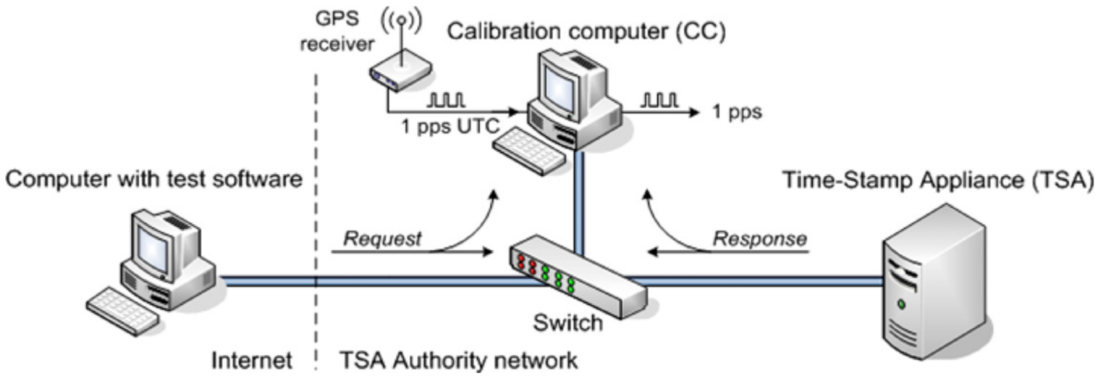

Figure 14. Passive calibration of TSA.

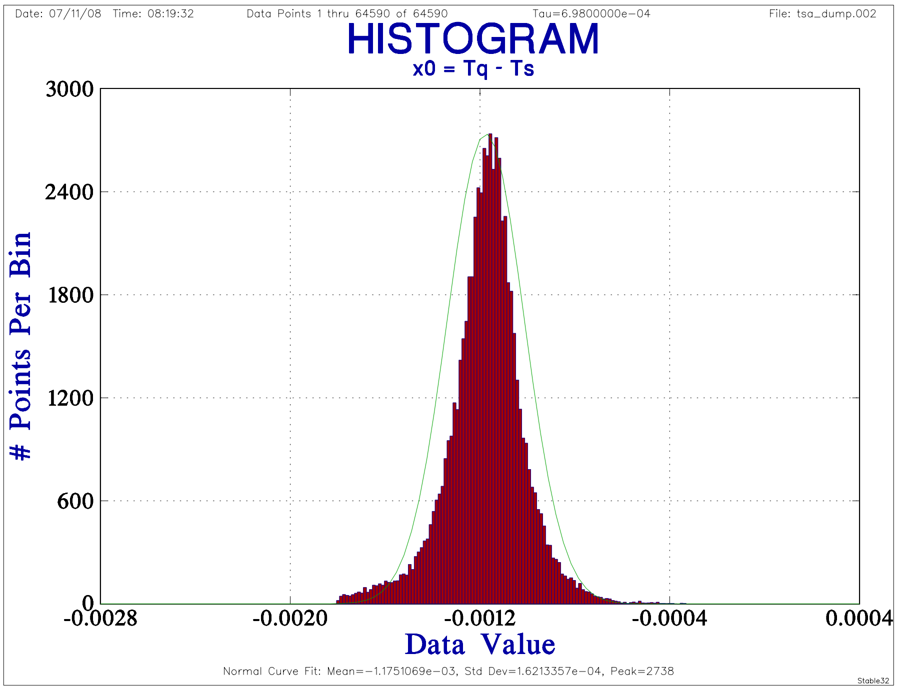

Figure 15. TSA time offset histogram.

4.1.1. Provider-oriented calibration. In [6] we proposed active and passive calibration methods applicable in POC. The active method simply employs the CC as a substitute for the client as shown in figure 13. The CC requests the time stamp at the TSA-defined access point. TSQ is issued at $T_Q(C)$ (with the uncertainty u_Q) and thus the time scale difference is

$$x_0 = T_Q(C) - T_S(S), \tag{10}$$

where the uncertainty of x_0 is given by u_{TS} since $u_Q \ll u_{TS}$.

The passive method is depicted in figure 14. In this case, the CC listens to the TSA communication that is replicated using a suitable arrangement depending on the link protocol. Examples are a splitter for optical links or an active network element (a hub or a properly configured switch) for the Ethernet link. Compared with the active method, the only difference is that instead of u_Q we use u_R (also $u_R \ll u_{TS}$).

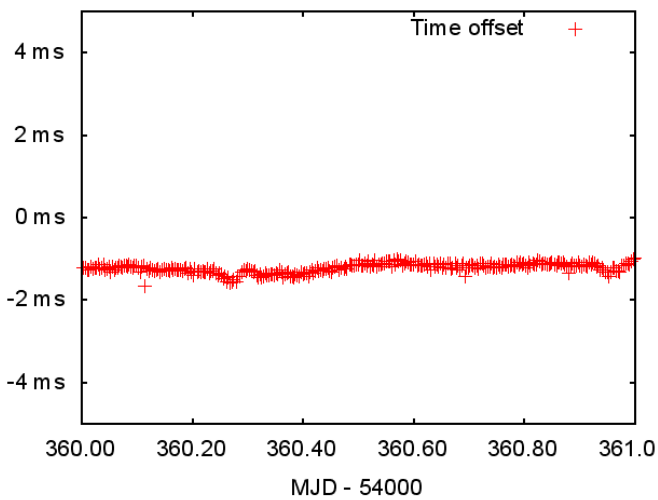

Figure 16. TSA time offset (provider-oriented measurement).

Table 2. TSA calibration results (delays in milliseconds).

		CESNET		BEV		ROA		IPE	
		TS_{OFFS}	TS_{DEL}	TS_{OFFS}	TS_{DEL}	TS_{OFFS}	TS_{DEL}	TS_{OFFS}	TS_{DEL}
CESNET	Mean			−446.7	221.4	31.2	72.2	1.0	17.6
	StDev			303.7	223.6	13.0	18.0	1.1	2.7
INRIM	Mean	13.4	44.1			29.9	71.5	13.6	44.3
	StDev	9.7	17.1			17.8	27.0	9.7	17.3
ROA	Mean	30.8	100.1					31.5	80.7
	StDev	19.6	82.8					22.9	27.5
IPE	Mean	1.2	17.8	−448.2	221.0	31.6	73.0		
	StDev	1.3	2.9	310.8	185.7	13.1	17.9		

4.1.2. Client-oriented calibration. The COC includes the network influence on the transmission of TSQ and TSR and offers the client the results in terms of the time-stamp offset

$$TS_{OFFS} = T_S(S) - T_Q(C) \tag{11}$$

and the time-stamp delay

$$TS_{DEL} = T_R(C) - T_Q(C), \tag{12}$$

where $TS_{OFFS} = -x_0$ defined in (10).

4.2. Experimental results

4.2.1. Provider-oriented calibration. We installed an experimental TSA with OpenTSA [7] software—an open source software package that implements the TSA functionality. We applied the passive method when several clients periodically asked the TSA for time stamps and the traffic was replicated to our CC. Each time stamp issued was recorded as a triplet (T_Q, T_R, T_S) representing $T_Q(C)$, $T_R(C)$ and $T_S(S)$. The difference $x_0 = T_Q - T_S$ showed a mean of –1.17 ms with standard deviation of 0.16 ms out of ~60 000 samples. The explanation for $|x_0| \gg u_S$ (in our TSA on the order of microseconds) is that the source code of the OpenTSA does not evaluate the time stamp upon TSQ arrival but after it is parsed. Figure 15 depicts the probability density function of x_0 and figure 16 shows the plot of x_0 over one day.

4.2.2. Client-oriented calibration. In this 'common view' measurement the CESNET, IPE and ROA operated their experimental TSA servers with the OpenTSA software and BEV its commercial TSA with proprietary software. Two identical CCs were operated at CESNET and IPE, and two other CCs with the same *tsa_mon* program at ROA and INRiM. Each CC requested each TSA for a time stamp once a minute.

Table 2 summarizes the mean and standard deviation of the TS_{OFFS} ($T_S - T_Q$) and TS_{DEL} ($T_R - T_Q$). The CCs are in the first column and the TSA servers in the first row.

The graphs in figures 17–20 display the results of measurement over one day: the difference $T_S - T_Q$ and the total response delay $T_R - T_Q$.

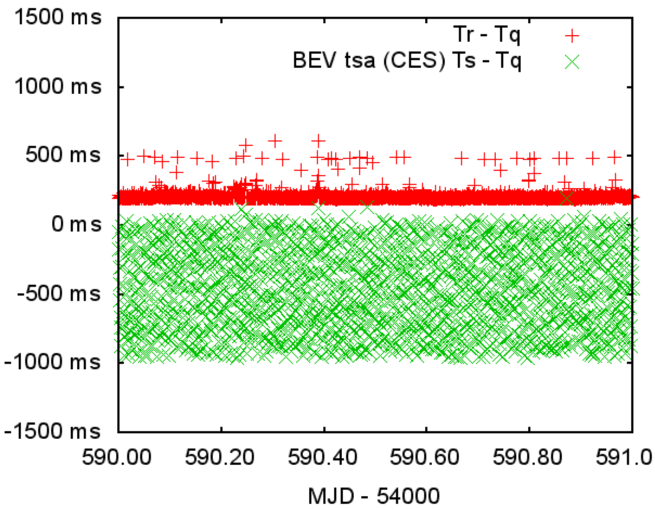

Figure 17. TSA measurement (CESNET to BEV).

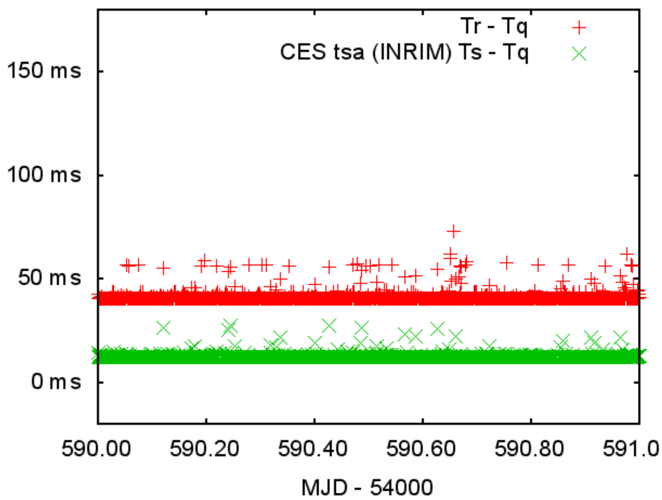

Figure 18. TSA measurement (INRiM to CESNET).

- *TSA software influence*. The commercial TSA operated by BEV was configured to issue time stamps with a resolution of 1 s, i.e. $u_{TS} > 1$ s. Figure 17 shows that TSA in BEV truncated the fraction of a second, i.e. the $T_S(S)$ was delayed against the actual $T(S)$ by up to 1 s. The mean of $T_S - T_Q$ is –447 ms and the mean response delay is 221 ms.

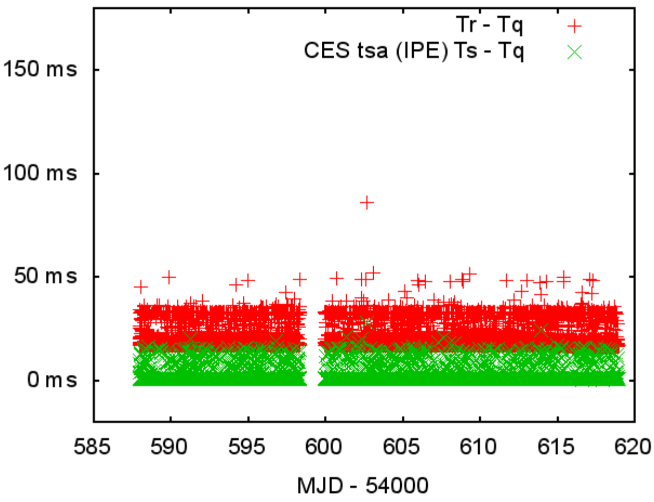

Figure 19. TSA long time measurement (IPE to CESNET).

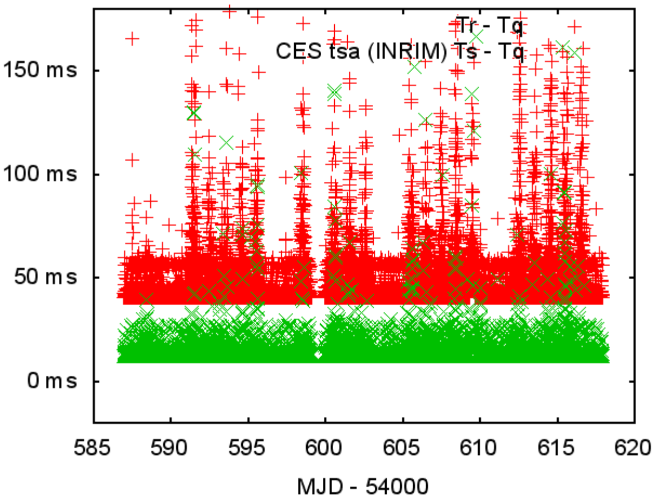

Figure 20. TSA long time measurement (INRiM to CESNET).

The experimental TSAs of CESNET, INRiM, IPE and ROA set the time-stamp resolution to 1 µs. Figure 18 shows typical results of measurement over one day between CC in INRiM and TSA in CESNET. The mean of $T_S - T_Q$ is 13 ms and the mean delay is 44 ms.

- *Network influence*. We can observe similar network effects as in the NTP calibration: small changes in delay between IPE and CESNET and large periodic changes in delay between INRiM and CESNET. Figures 19 and 20 show $T_S - T_Q$ and $T_R - T_Q$ in a month-long period.

5. Conclusions

NTP service. With a small (<1 ms) and constant network delay between the NTP server and CC, the server can provide time information with an uncertainty of the order of microseconds. In the wide area network within Europe, the time information uncertainty is hundreds of microseconds with a median of tens of microseconds. In an unstable network, as we observed in one case, the accuracy was degraded to tens of milliseconds, which made the service unpredictable and hardly usable. Performing the COC, we focused on one-shot synchronization by the NTP service which, however, is designed to provide synchronization (including frequency control) on a long-term basis that allows additional filtering.

TSA service. We have demonstrated the use of the CC in a POC of an experimental TSA. A systematic uncertainty of about 1 ms due to software delays in TSA has been observed which indicates the achievable uncertainty in time stamping.

Acknowledgments

The authors thank Werner Mache and Robert Sebestyan of BEV, Michael Rost of PTB, Luis Batanero of ROA and Giuseppe Vizio of INRiM for their collaboration within EURAMET project #917. The work was supported by the research intent of the Czech government 'Optical National Research Network and Its New Applications' and by the Czech Office for Standards, Metrology and Testing.

References

[1] Mills D L 1992 *Network Time Protocol Specification, Implementation and Analysis* RFC 1305
[2] Adams C, Cain P, Pinkas D and Zuccherato R 2001 *Internet X.509 Public Key Infrastructure, Time-Stamp Protocol (TSP)* RFC 3161
[3] Mills D L 1998 Adaptive hybrid clock discipline algorithm for the network time protocol *IEEE/ACM Trans. Networking (TON)* **6** 505–14
[4] TEDIA web pages, http://www.tedia.cz/produkty/pct7424.html
[5] Mogul J, Mills D L, Brittenson J, Stone J and Windl U 2000 *Pulse-Per-Second API for UNIX-like Operating Systems* RFC 2783
[6] Smotlacha V, Tyml P and Čermák J 2007 Calibration of time stamp authorities *Proc. 15th IMEKO TC4 Int. Symp. on Novelties in Electrical Measurements and Instrumentations (Iasi, Romania)* vol I
[7] Glozik Z, OpenTSA project web pages, http://www.opentsa.org/