CHARACTERISTICS OF TIME SYNCHRONIZATION RESPONSE OF NTP CLIENTS ON MS WINDOWS OS AND LINUX OS

Katsuhisa Sato and Kazuyoshi Asari
Mizusawa VERA Observatory
National Astronomical Observatory of Japan
Mizusawa, Oshu, Iwate, 023-0861 Japan
E-mail: ***hisa@miz.nao.ac.jp, asa@miz.nao.ac.jp***

Abstract

The clock offset between a GPS-based NTP time server and NTP time client software, installed in the MS Windows Operating System (OS) and the Linux OS on PCs, are measured and evaluated. The clock offset on MS Windows 98 OS shows a trend with a range of about 55 ms. The NTP time client software on MS Windows OS adjusts the internal clock with the Application Program Interface (API) timer function. The resolution of the API timer function depends on the hardware interrupt of the PC system timer, which is 54.9 ms of IRQ0 for MS Windows 98 OS. Thus, this range of the trend in the clock offset is considered to be caused by the resolution of API timer function. The clock offset on MS Windows XP OS shows time resolutions of 1 ms for the 1 ms API timer mode and 10 ms for the 10 ms API timer mode. The resolution of 1 ms is dependent on a hardware interrupt of IRQ8 (976 μs) that is generated by a PC real-time clock. The resolution of the system timer on MS-windows XP depends on the hardware platform and shows 10 ms or 15ms. The clock offset on Linux OS also has a trend as well as a periodic divergence. The trend interval of the drift is estimated to be about 35 minutes. The origin of this trend in interval is presumed to be the loop constant of the kernel Phase Lock Loop (PLL). The Standard Deviation of this clock offset for 24 hours is 0.95 ms. Evidently, both MS Windows OS and Linux OS adopt different algorithms for keeping the internal clock. The accuracy of time synchronization by NTP is restricted by the algorithms. The limitation of the time synchronization accuracy on MS Windows OS is related to the resolution of API that depends on a hardware interrupt generated in the PC hardware system timer and real-time clock. On the other hand, the loop constant of the kernel clock algorithm restricts the time synchronization accuracy by NTP on Linux OS.

I. INTRODUCTION

The Network Time Protocol (NTP) is widely used to synchronize the local clock in a work station (WS) or a personal computer (PC) to a NTP time server through the Internet. The time synchronization accuracies of NTP are classified in two categories. One is the synchronization accuracy of NTP time server to UTC time frame and the other is the clock offset between the referring NTP time server and the WS or PC local clock. A GPS-based NTP time server that obtains GPS time using a GPS receiver **[1]** was set up to measure and evaluate the time synchronization accuracy of the NTP. A Furuno model TS-820 GPS receiver was connected to a NTP server that runs a PC xntpd daemon version 3.4e NTP server software on FreeBSD Version 2.2.1-R. The time interval between a one pulse-per-second (pps) signal

from the GPS receiver and the NTP time server and a 1pps signal from a cesium atomic clock that keeps UTC time was measured to monitor the clock offset of the NTP time server. The clock offset between the referring NTP time server and the local WS or PC clock shows different characteristics for different operational systems (OS). The clock offset between NTP time server and a local PC clock is measured by client NTP synchronization software for MS-Windows OS and for Linux OS to evaluate the OS dependency of the NTP time synchronization.

II. TIME SYNCHRONIZATION ACCURACY OF NTP TIME SERVER TO UTC

The time synchronization residual of a GPS-based NTP time server relative to a UTC time frame is mostly found in the time synchronization accuracy of a 1pps signal from a GPS receiver to GPS time frame. This is because both the NTP PC daemon and the 1pps signal from a GPS receiver are directly related. The time interval between the 1pps signal from the TS-820 GPS receiver and the 1pps signal from the HP-5071A cesium atomic clock used to keep the UTC time frame at the National Astronomical Observatory of Japan (NAOJ) is measured using a HP-5370B time-interval counter. The results of the measurements are shown in Figure 1.

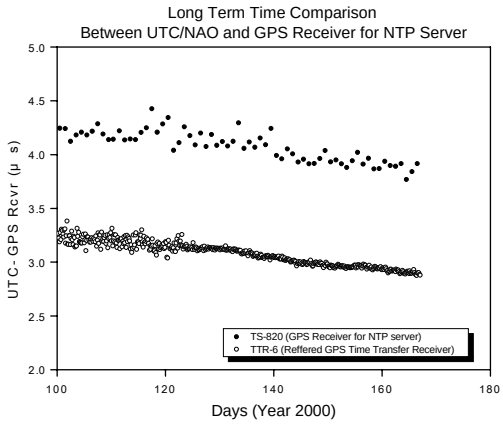

Figure 1. Results of the long-term time comparison measurement of the 1pps signals between a TS-820 GPS receiver and a HP-5071A cesium atomic clock that keeps UTC (NAOJ).

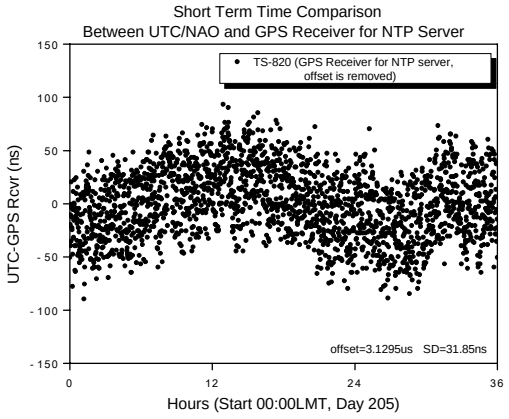

Figure 2. Results of the time comparison measurement for 36 hours. The time interval of the 1pps signals between a TS-820 GPS receiver and a HP-5071A cesium atomic clock that keeps UTC(NAOJ) are measured.

As shown in Figure 1, the GPS selective availability (SA) code ceased on day 123 in the year 2000. Since then, the 1pps signal deviation from the GPS receiver has been reduced to one fifth of the value derived prior to that. The standard deviation (SD) of the time interval measurement for the 1pps signals is calculated as 31.85 ns for the data of day 205, shown in Figure 2 where the SA code had already been turned off. The daily variation of the time interval measurement in Figure 2 is considered to be variation of propagation delay in the ionosphere.

The time offset between the GPS 1pps signal and the 1pps signal of the cesium clock is 3.13 μs. This offset includes a propagation delay of the GPS signal and an internal delay of the GPS receiver. The

internal receiver delay differs for each GPS receiver. The offset of 1pps signal is considered to be the synchronization accuracy and will cause systematic synchronization error in the NTP time server relative to the GPS time frame.

The SD of the time interval measurement between the GPS 1pps signal and the 1pps signal from the cesium clock is the precision of synchronization between the NTP time server and the GPS time frame. This precision is negligibly small (31.85 ns) for the conventional NTP daemon where the overall synchronization accuracy is at the millisecond level **[2]**. The measured precision meets the requirement for the nano kernel NTP daemon **[3]**.

III. CLOCK OFFSET OF NTP TIME SYNCHRONIZATION ON MS-WINDOWS

The clock offset of the NTP time synchronization between the NTP time server and the internal local PC clock is measured each minute using TCP/IP based NTP time client software which is called AboutTime. AboutTime has a logging function in MS-windows 98 OS. The NTP time server and the client PC are connected to the same HUB. The results are shown in Figure 3. The network delay time of NTP packet in 10 MHz Ethernet is also measured by the same software and is shown in Figure 4.

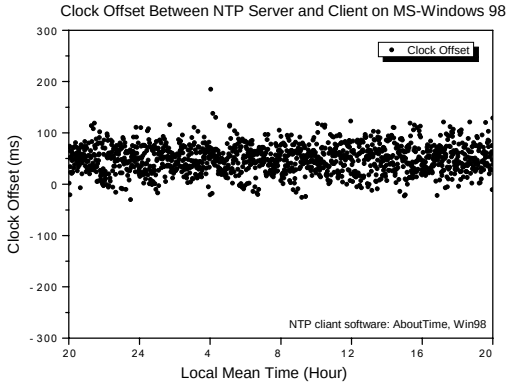

Figure 3. Clock offset of the NTP time synchronization between the GPS-based NTP time server and the local PC clock with an MS-Windows 98 OS.

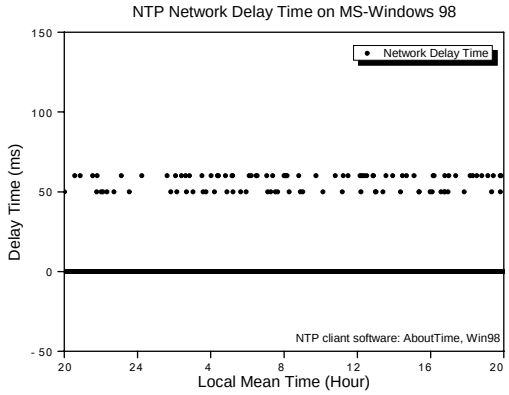

Figure 4. Network delay time of NTP packet for a 10 MHz Ethernet on an MS-Windows 98 OS.

The measured clock offset for NTP time synchronization shows the trend seen in Figure 3. The range of the trend in the clock offset is about 55 ms. On the other hand, the network delay time of NTP packet shows a discrete trend seen in Figure 4. This occurs due to the NTP time client software which applies SNTP on an MS Windows OS adjusting the internal clock with the Application Program Interface (API) timer functions. The resolution of the synchronization by NTP time client software depends on the resolutions of the API timer functions and the values for MS Windows 98 OS are 1 ms, 5 ms and 54.9 ms. The resolution of 1 ms is related to IRQ8 (976μs) that is generated by a PC real-time clock. The resolution of 54.9 ms is related to IRQ0 that is generated by hardware interrupt of the PC system timer.

Table 1. Resolutions of Application Program Interface on MS Windows OS.

	Win98	WinXP (NT family: workstation type)	WinXP (NT family: server type)
Sleep	5ms	10ms	15ms
Sleep(timeBeginPeriod=1ms)	1ms	1ms	1ms
timeGetTime	1ms	10ms	15ms
timeGetTime(timeBeginPeriod=1ms)	1ms	1ms	1ms
GetTickCount	5ms	10ms	15ms
GetTickCount(timeBeginPeriod=1ms)	1ms	10ms	15ms
SetTimer	54.9ms*	10ms	15ms
SetTimer(timeBeginPeriod=1ms)	54.9ms*	10ms	15ms

*:specification, others:nominal value

Table 1 shows the API timer resolutions. Thus, the range of the trend in the clock offset is considered to be caused by the resolution of API timer function that is used in the NTP time client software. The resolution for this case is 54.9 ms.

The NTP time client software calculates the clock offset according to the following equations.

$$\boldsymbol{d} = (T4-T1) - (T2-T3) \quad (1)$$

$$\boldsymbol{t} = ((T2-T1) + (T3-T4)) / 2, \quad (2)$$

where ***d*** is the network time delay and ***t*** is clock offset between the clock of the NTP server and the clock of the client. T1 is the time request sent by the client, T2 is the time request received at the server, T3 is the time reply sent by the server, and T4 is the time reply received at the client. The clock offset of the NTP time client ***t*** is isolated from network delay time ***d***. The NTP time client software for MS-Windows shows a linear clock offset in Figure 3 that is the difference between the clock in the NTP server and the clock in the client PC. Therefore, the discrete pattern remains in the other network delay time ***d***. Once the linear clock offset ***t*** exceeds the IRQ0 interrupt period (54.9 ms) of the system timer, it is presumed from the data shown in Figure 4 that the NTP time client software resets the clock offset ***t*** to zero, adjusting the internal real-time clock IRQ8 (976 μs) by the API timer function. This reset sequence causes the trend in the measured clock offset data.

The clock offset of the NTP time synchronization between the NTP time server and the PC internal local clock is also measured by two NTP time client software packages for MS-windows XP OS. One of the NTP time client software packages is AboutTime and the other is Automachron. The results are shown in Figure 5.

The measured clock offset for the AboutTime NTP time client software shows two different data groups on two different PCs. One is around 1 ms and the other is around 10 ms. This indicates that some PCs do not allow the execution of some API timer functions in 1 ms mode. Generally, the clock offset for MS-windows XP shows a time resolution of 1 ms for the 1 ms API timer function and 10 ms for the 10 ms API timer function. The resolution of 1 ms is related to IRQ8 (976 μs) that is generated by a real-time PC clock. The resolution of the system timer for MS-windows XP depends on the hardware platform and shows 10 ms or 15 ms. The value in this case is 10 ms.

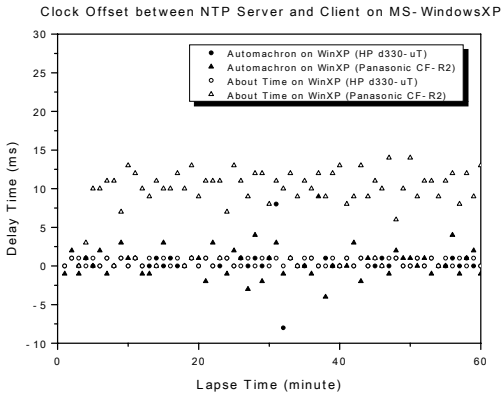

Figure 5. Clock offset of the NTP time synchronization between the GPS-based NTP time server and the local PC clock on MS-Windows XP OS.

IV. CLOCK OFFSET OF NTP TIME SYNCHRONIZATION ON LINUX

The clock offset of the NTP time synchronization, the delay time of the NTP packet in 10MHz Ethernet, and the jitter of clock offset on LINUX OS are measured every minute by ntpd daemon version 3-5.93e and ntpq. The NTP server and the client PC are connected to the same HUB. The results are shown in Figures 6, 7, and 8.

The measured clock offset in the NTP time synchronization has the trend and a periodic divergence shown in Figure 6. There is correlation between the network delay time in Figure 7 and the periodic divergence of the clock offset in Figure 6. The large divergences of the jitter that exceeds 128 ms in Figure 8 do not affect the clock offset disturbance, because a clock offset that exceeds 128 ms is discarded by the ntpd algorithms. The large network delay time divergences are occurring at the same time that the packet traffic in the LAN is quite large. Also, comparing Figure 6 and Figure 7, the NTP packet network delay time shows discrete trends while the clock offset shows a series transition. The standard deviation of the clock offset for 24 hours of data shown in Figure 6 is 0.95 ms.

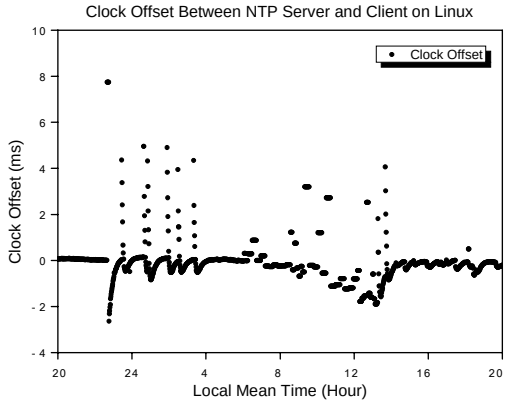

Figure 6. Clock offset of the NTP time synchronization between a GPS-based NTP time server and a local PC clock on Linux OS.

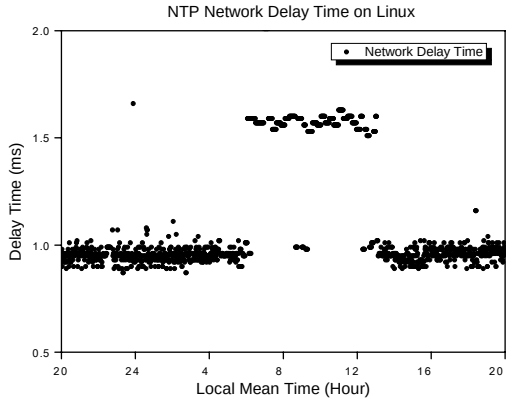

Figure 7. Network delay time of NTP packet in a 10 MHz Ethernet on Linux OS.

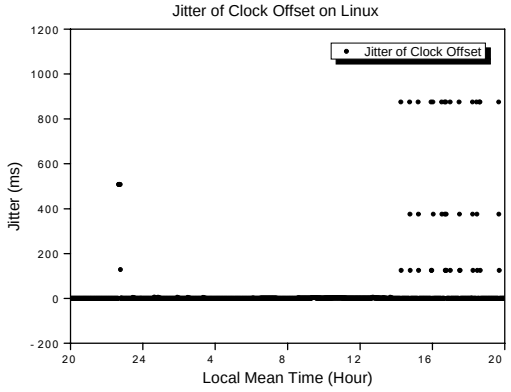

Figure 8. Jitter of clock offset on Linux OS.

The clock discipline algorithm of ntpd adjusts the internal PC clock, compensates for the intrinsic frequency error, and adjusts the server update interval and loop time constant dynamically in response to measured network jitter and the internal WS or PC clock stability **[4]**. The clock in Linux is implemented by the software defined as the kernel clock. This kernel clock is not restricted from the hardware PC clock. The kernel clock algorithm of the ntpd consists of a phase detector, clock filter, loop filter, frequency discipline, and variable frequency oscillator **[4]**. The response characteristic of this algorithm shows the same characteristic as the lock-in process of a phase lock loop (PLL). The basic transfer function of the PLL is approximated **[5]** as

$$F(s) = \frac{\omega_c^2}{s^2\tau^2}\left(1+\frac{s\tau}{\omega_z}\right)e^{-sT}, \qquad (3)$$

where ω_c is the gain (crossover frequency), ω_z the corner frequency of the lead network (necessary for PLL stability), T is the data-filter delay, and τ is a bandwidth parameter. The simulation shows that the PLL reaches zero error in 39 minutes, at the widest bandwidth and a 100 ms phase change **[5]**.

The clock offsets shown in Figure 6 indicate characteristics of the lock-in process. The magnified jitter divergences from Figure 8 and the clock offset of Figure 6 are both plotted in Figure 9.

In Figure 9, the large jitter divergence arises simultaneously when the PLL lock in the kernel clock is off. The trend interval of the drift in Figure 8 is estimated at about 35 minutes. This recovery time is almost the same as the approximated value of the simulated transfer function for the PLL **[5]**. Thus, the origin of the trend interval in Figure 9 is presumed to be the loop constant of the PLL.

The NTP version 3 discipline selects either PLL or frequency-lock loop (FLL) mode on the basis of τ. This discipline corrects the internal PC clock and compensates for its intrinsic frequency error. The discipline also adjusts the various parameters dynamically in response to measured network latency variations or jitter as well as oscillator frequency stability or drift **[6]**. The relation between the clock offset in Figure 6 and the network delay time in Figure 7 coincide with the switching algorithm between

PLL and FLL, since the trend of the clock offset is calm while the network delay time is large.

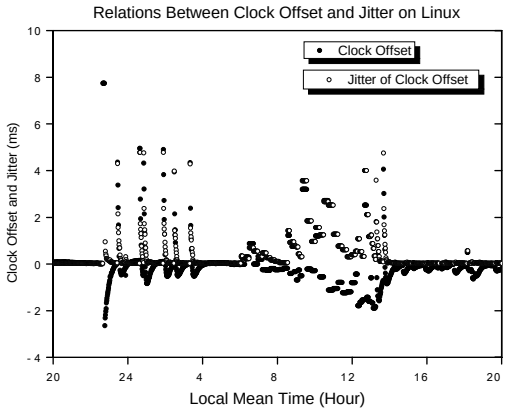

Figure 9. Relations between clock offset and clock offset jitter.

V. CONCLUSION

The synchronization accuracy of the NTP time server to the UTC time frame is evaluated measuring the time interval between the 1 pps signal produced by a GPS receiver that is connected to a NTP time server and a 1pps signal from the cesium atomic clock that keeps UTC time. The SD of the time interval measurement is calculated as 31.85 ns. The time offset between the GPS 1pps signal and the 1pps signal from the cesium clock is 3.13 μs. The offset of the 1pps signal is considered to be the synchronization accuracy and will cause systematic synchronization error between the NTP time server clock and GPS time. This offset is comparable to the accuracy of a millisecond NTP daemon and is constant, since the origin is considered an internal hardware delay of the GPS receiver. The SD of the time interval measurement between the GPS 1pps signal and the 1pps signal from the cesium clock is the precision of synchronization between the NTP time server clock and GPS time. This precision, measured as 31.85 ns, is negligibly small for the conventional NTP daemon, where the overall synchronization accuracy is at the millisecond level **[2]**.

The clock offset between a GPS-based NTP time server and NTP time client software, installed in a PC MS Windows OS or Linux OS, are measured and evaluated. The clock offset for MS Windows 98 OS shows a trend with an interval range that is about 55 ms. The NTP time client software which applies SNTP on MS Windows OS adjusts the internal clock with the API timer function. The resolution of the API timer function depends on the PC hardware interrupt of the system timer, which is 54.9 ms of IRQ0 for MS Windows 98 OS. Thus, this trend of the range in the clock offset is known to be caused by the resolution of API timer function. The clock offset on MS Windows XP OS shows time resolutions of 1 ms for the 1 ms API timer mode and 10 ms for the 10 ms API timer mode. The resolution of 1 ms is dependent on a hardware interrupt of IRQ8 (976 μs) that is generated by a PC real-time clock. The resolution of the system timer on MS-windows XP depends on the particular hardware platform used and shows 10 ms or 15 ms.

The clock offset of the NTP time synchronization, the network delay time of NTP packet in 10 MHz Ethernet, and the jitter of clock offset are measured each minute on Linux OS. The clock on Linux OS is implemented by the software defined as the kernel clock. This kernel clock is not restricted by the PC

internal hardware clock, whose resolution is limited to 54.9 ms by IRQ0 and limited to 1 ms by IRQ8. The response characteristic of the kernel clock algorithm shows the same characteristic as the lock-in process of the PLL. The measured clock offset in the NTP time synchronization has a trend and a periodic divergence. The interval to reset the trend is estimated at about 35 minutes. This recovery time is almost the same as the approximated value of the simulated transfer function for the PLL **[5]**. Thus, the origin of the trend interval is presumed to be the loop constant of the PLL. The SD of the clock offset for 24 hours is 0.95 ms.

Evidently, both MS Windows OS and Linux OS adopt different algorithms for keeping the internal clock. The accuracy of time synchronization by NTP is restricted by the algorithms. The limitation of the time synchronization accuracy on MS Windows OS is caused by the resolution of API timer functions that depends on a PC hardware interrupt generated in the hardware system timer and the PC real-time clock. On the other hand, the loop constant of the kernel clock algorithm restricts the time synchronization accuracy by NTP on Linux OS.

ACKNOWLEDGMENT

The authors would like to thank Mr. Bernie McCune for proofreading this manuscript.

REFERENCES

[1] D. L. Mills, A. Thyagarajan, and B. C. Huffman, 1998, *"Internet timekeeping around the globe,"* in Proceedings of the 29th Annual Precision Time and Time Interval (PTTI) Applications and Planning Meeting, 2-4 December 1997, Long Beach, California, USA (U.S. Naval Observatory, Washington, D.C.), pp. 365-371.

[2] D. L. Mills, 1990, *"On the accuracy and stability of clocks synchronized by the Network Time Protocol,"* **ACM Computer Communications Review, 20,** 65-75.

[3] D. L. Mills and P.-H. Kamp, 2001, *"The nanokernel,"* in Proceedings of the 32nd Annual Precision Time and Time Interval (PTTI) Applications and Planning Meeting, 28-30 November 2000, Reston, Virginia, USA (U.S. Naval Observatory, Washington, D.C.), pp.423-430.

[4] D. L. Mills, 1997, *"The network computer as precision timekeeper,"* in Proceedings of the 28th Annual Precision Time and Time Interval (PTTI) Applications and Planning Meeting, 3-5 December 1996, Reston Virginia, USA (U.S. Naval Observatory, Washington, D.C.), pp. 97-107.

[5] D. L. Mills, 1991, *"Internet time synchronization: the Network Time Protocol,"* **IEEE Transactions on Communications, 39,** 1482-1493.

[6] D. L. Mills, 1998, *"Adaptive Hybrid Clock Discipline Algorithm for the Network Time Protocol,"* **IEEE ACM Transactions on Networking, 6,** 505-514.

M. A. Dietz, C. S. Ellis, and C. F. Starmer, 1995, *"Clock Instability and its Effect on Time Intervals in Performance Studies,"* in Proceedings of the Computer Measurement Group 1995 (CMG95), pp. 439-448.

D. L. Mills, 1989, *"Measured performance of the Network Time Protocol in the Internet system,"* DARPA Network Working Group Report RFC-1128, University of Delaware.

J. Rourke and C. Zacker, 2000, **PC Hardware: The Complete Reference** (McGraw-Hill, New York).