Calibrating NTP

Faten Mkacher, Andrzej Duda

Calibrating NTP

Faten Mkacher*¶ and Andrzej Duda¶
*Gorgy Timing, F-38350 La Mure d'Isère, France
¶Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France
Email: {faten.mkacher@imag.fr, andrzej.duda@imag.fr}

***Abstract*—In this paper, we propose a method of improving the accuracy of NTP time synchronization by taking into account asymmetric transmission delays due to different bandwidth or routing on the forward and backward paths. The method consists of calibrating NTP synchronization by: i) deploying a time box with a GPS clock at a given client, ii) measuring the one-way transmission delay on the forward and backward path and finding the minimal delays, iii) using the minimal delays in the estimation of the clock offset at the client to take into account path asymmetry, and iv) recalibrating if routes change. The paper first reports on the measurements of one-way transmission delays between a client and a server interconnected by several routers. We then use the parameters of the delay distributions to estimate the clock offset at the client. To validate the proposed method, we have compared the clock offsets computed by standard NTP and calibrated NTP based on the GPS time reference. The measurements show significant improvement of the NTP time synchronization accuracy and precision.**

***Index Terms*—time synchronization, Network Time Protocol (NTP), one-way delay, accuracy and precision**

I. Introduction

The Network Time Protocol (NTP) is one of the most widely used protocols for clock synchronization [1], [2]. NTP uses UDP over packet-switched networks to synchronize clocks between one or several time servers and clients. NTP defines several modes of operation, the client/server mode being the most commonly used one. In this mode, a client sends a request containing a timestamp to a time server that replies providing two other timestamps. The two-way packet exchange with timestamps allows estimating the time offset between the client and the server.

The precision of NTP time synchronization depends on the validity of the assumption related to symmetric transmission delays between the client and the server. If this assumption does not hold, which is a common case in the current Internet, the NTP synchronization scheme results in significant errors.

Freris et al. [3] showed that asymmetry cannot be measured in a pairwise synchronization system based only on recorded timestamps, even with an infinite number of round trip measurements. Due to this fundamental limitation, neither NTP nor Precision Time Protocol (PTP, IEEE 1588) can mitigate the error in the clock offset due to asymmetry without some estimation of one-way delays. Nevertheless, several authors proposed ways of accuracy improvement by estimating delay asymmetry coming from different bandwidth on the forward and backward paths [4], [5], [6], [7], [8], [9], [10], [11].

Asymmetric transmission delays are common in the Internet. Pathak et al. [12] studied one-way transmission delays on the PlanetLab testbed and showed that i) asymmetry is quite prevalent and ii) it can be attributed at least in part to the asymmetry in routing paths. They also presented conclusive evidence that delay asymmetry is a dynamic property that varies depending on routing dynamics.

In this paper, we propose a method of improving the accuracy of NTP time synchronization by taking into account asymmetric transmission delays due to different bandwidth or routing on the forward and backward paths. The method consists of: i) deploying a time box with a GPS clock at given client premises (we assume that the server has an accurate time source), ii) calibrating—measuring the one-way transmission delay on the forward and backward path and finding the minimal delays, iii) using the minimal delays in the estimation of the clock offset at the client to take into account path asymmetry, and iv) detecting changes in operating conditions to re-calibrate.

After calibration, the client does not longer use the time box and relies on NTP synchronization with a modified expression for the time offset. The proposed method requires re-calibration—when the operating conditions (e.g., routing) change between the client and the server, which we can detect with the `ping` and `traceroute` tools, we need to redo calibration to find the new parameters of the one-way transmission delay.

The motivation for calibrating NTP comes from SCP-Time [13], a research project on the dissemination of the certified legal time required in many forms of commercial transactions or applications. Through SCPTime, Gorgy Timing will provide time synchronization services to various clients with different precision over various protocols including NTP over the Internet.

In this paper, we first recall the principles of NTP time synchronization (Section II) and report on the measurements of one-way transmission delays between a client and a server interconnected by several routers (Section III). We present the histograms of the delays for different hours of a day. The shape of the delay distributions corresponds to a constant and a variable random part. We then use the parameters of the delay distributions to estimate the clock offset at the client (Section IV). Unlike many papers that validate proposed schemes with simulations, we perform measurement experiments to compare the clock offsets computed by standard NTP and calibrated

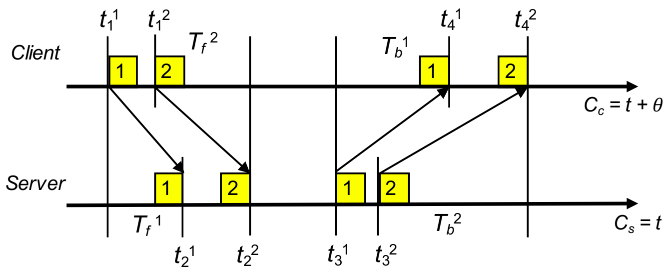


Figure 1. Principles of NTP

NTP based on the GPS time reference (Section V). The measurements show significant improvement of the NTP time synchronization accuracy and precision. Finally, we discuss some related work (Section VI) and conclude (Section VII).

II. Basics of NTP

Fig. 1 presents the principle of NTP time synchronization in client/server mode. We adopt the standard NTP assumptions: the server has a perfect clock $C_s = t$ and the client wants to synchronize its clock $C_c = t + \theta$ with the server, θ being the time offset between the client and the server.

Time synchronization relies on the two-way packet exchange. In general, the client can send n NTP requests to the server that responds with n responses ($n = 2$ in Figure 1; the implementation we use, can send up to 8 requests). The instants of sending and receiving packets are recorded and included in NTP packets: with packet j, the client learns timestamps t_2^j, t_3^j and knows timestamps t_1^j, t_4^j. Note that the client records timestamps according to its clock C_c.

We denote forward (from the client to the server) and backward (from the server to the client) one-way transmission times: T_f^j and T_b^j. If we assume that the clock drift during the exchange is constant, we have the following relations:

$$t_2^j = t_1^j + T_f^j - \theta^j, j = 1, ..., n, \tag{1}$$

$$t_4^j = t_3^j + T_b^j + \theta^j, j = 1, ..., n. \tag{2}$$

If one-way transmission times are symmetric ($T_f^j = T_b^j$), the time offset becomes:

$$\theta^j = \frac{(t_2^j - t_1^j) + (t_3^j - t_4^j)}{2}, j = 1, ..., n. \tag{3}$$

NTP uses this equation to compute the time offset.

In general, one-way transmission times are asymmetric ($T_f^j \neq T_b^j$) and vary in time ($T_f^j \neq T_f^{j+1}$). In this case, the time offset becomes:

$$\theta^j = \frac{(t_2^j - t_1^j) + (t_3^j - t_4^j)}{2} + \frac{T_b^j - T_f^j}{2}, j = 1, ..., n \tag{4}$$

We can observe that if one-way transmission times are symmetric ($T_b^j = T_f^j$), the expression is the same as Eq. 3.

Eq. 4 shows that the accuracy of NTP time synchronization depends on the difference of one-way transmission times so the assumption of symmetric one-way transmission times is the main source of accuracy errors. If we can estimate one-way transmission times in a more precise way, we can improve the accuracy of time synchronization.

Note that most NTP implementations send several NTP requests and compute round trip network delay δ^j as follows:

$$\delta^j = (t_4^j - t_1^j) - (t_3^j - t_2^j), j = 1, ..., n \tag{5}$$

They can then choose the best set j of timestamps for computing the time offset θ^j based on the smallest δ^j.

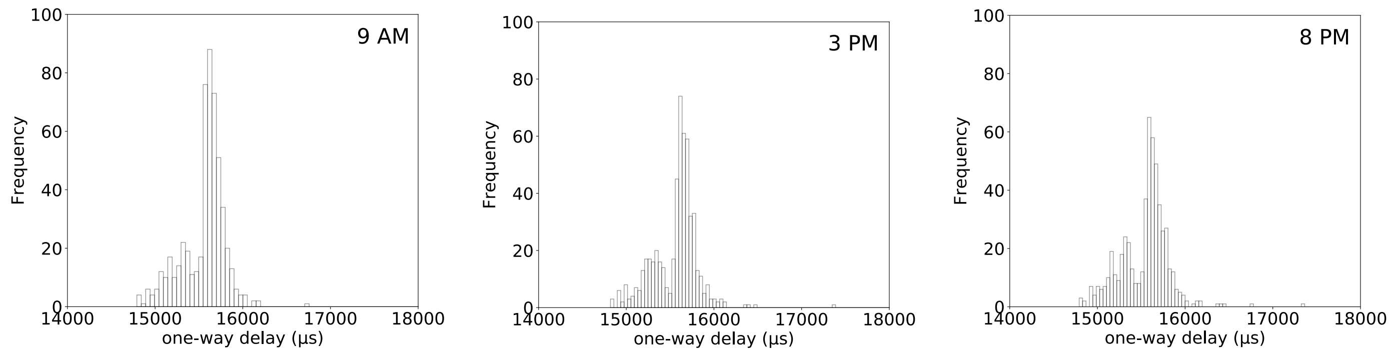
III. Estimation of One-Way Transmission Times

The most common delay metric is Round Trip Time (RTT) defined by the IPPM (IP Performance Metrics) IETF group [14] as the time interval between the injection of the first bit of the packet into the network and the reception of the last bit of the packet, supposing the receiver resent the packet immediately after its reception. Measuring RTT does not require synchronized clocks at the hosts since the sender computes RTT only based on its clock.

RTT cannot capture *path asymmetry*, the fact that the path from a source to a destination (forward path) may differ from the path from the destination back to the source (backward path). The difference in the transmission delays over forward and backward paths may come from two sources. First, the links between routers may have different capacity, e.g., an ADSL line [5]. Second, it may also come from *hot potato routing*: when two ASes (Autonomous Systems) are competitors and have a peer-to-peer relationship in propagating Border Gateway Protocol (BGP) route advertisements, they have interest in getting rid of a given packet as soon as possible by forwarding them to the closest egress point in terms of the internal routing cost. Hot potato routing is a consequence of a rule in the BGP decision process stating that a router always prefers to use a route learned over an eBGP session compared to a route learned over an iBGP session. Hot potato routing results in asymmetric paths because each AS sends packets through its favorable exit point, the points being different for different ASes.

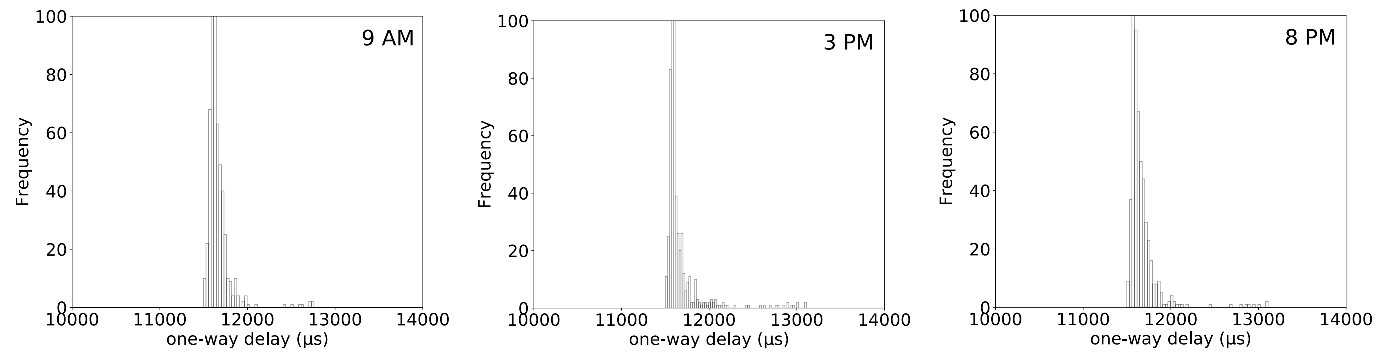
The one-way delay (OWD) metric [15] is the time interval between the injection of the first bit of the packet in the link and the reception of the last bit of the packet in the other measuring point. Its measurement requires synchronized clocks at the sender and the receiver.

We have set up an experiment to measure the one-way delays between a client and a server connected to different ASes. The client is at the Gorgy Timing premises in Grenoble, France, and the server is at the Observatoire de Paris in Meudon, France. Both the client and the server locally connect to 100 Mb/s Ethernets. Gorgy Timing uses Orange as an ISP over a 1 Gb/s link and Observatoire de Paris connects to Renater also over a 1 Gb/s link. Traceroute between the two end-points shows 14 intermediate routers on the forward and 11 routers on the backward path with an average RTT of 27 ms. The Internet connectivity between the client and the server has several interesting characteristics: there are several ASes on the path (3215 Orange, 5511 Opentransit, 3257 GTT, 2200 Renater), the end-points are connected to different



(a) (b) (c)

Figure 2. Histograms of T_f



(a) (b) (c)

Figure 3. Histograms of T_b

types of ISPs (Orange: commercial provider, Renater: public network with large capacity), and there is an asymmetric number of routers.

The client runs Ledi Network ATS with Digi ConnectCore 9P 9215 card that provides NTP timestamps over SNMP. The server is Meinberg M1000/MRS at the Observatoire de Paris. The client and the server are synchronized with GPS with a precision of 50 ns.

We have measured T_f and T_b (index j skipped) at different hours of a day: 9AM, 3PM, and 8PM. Figures 2 and 3 show the histograms for each direction. We can notice that the one-way delays are highly asymmetric with the difference of around 3 ms. The shape of the distributions also varies: the variance of the forward distribution is much greater than that of the backward one.

We can observe that one-way delays T_f and T_b include a constant and a variable random part:

$$T_f = d_f^{min} + d_f, \tag{6}$$

$$T_b = d_b^{min} + d_b, \tag{7}$$

where d_f^{min}, d_b^{min} are constant and d_f, d_b are random variables. We denote the average values of the distributions as $\hat{T}_f$ and $\hat{T}_b$.

We can notice that the shape of the histograms is significantly different for forward and backward paths and the distributions do not vary much in time. To find the best fitting distributions of T_f and T_b, we have tested several most important distributions: Gamma, Weibull, Normal, and Log Gamma. Figure 4 shows the results of fitting distributions at 9AM using the Maximum Likelihood Estimation (MLE) algorithm. For T_f, the best fitting distribution is a mixture of two Normal distributions:

$$p(x) = \lambda_1 \mathcal{N}(\mu_1, \sigma_1) + \lambda_2 \mathcal{N}(\mu_2, \sigma_2) \tag{8}$$

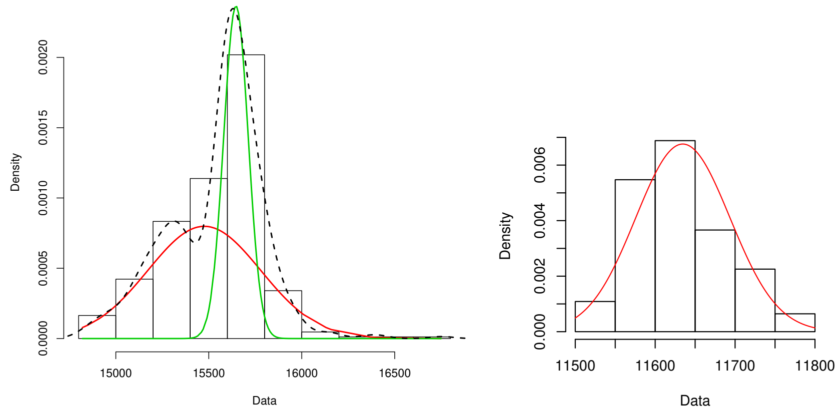
with $\lambda_1 = 0.5540935, \lambda_2 = 0.4459065$, the mean values are $\mu_1 = 15659.26, \mu2 = 15465.70$, with standard variation of $\sigma_1 = 76.42764, \sigma_2 = 315.51818$. For T_b, we have fitted a Gamma distribution:

$$f(x; \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}, \; x, \alpha, \beta > 0, \tag{9}$$

with the following parameters: shape $\alpha = 38521$ and rate $\beta = 3.3$, where $\Gamma(\alpha)$ denotes the Gamma function.

IV. Calibrating NTP

The goal of calibration is to take advantage of the one-way delay measurements to mitigate the systematic error



(a) T_f – mixture of two Normal distributions (b) T_b – Gamma distribution

Figure 4. Fitting the distribution of T_f and T_b, 9AM

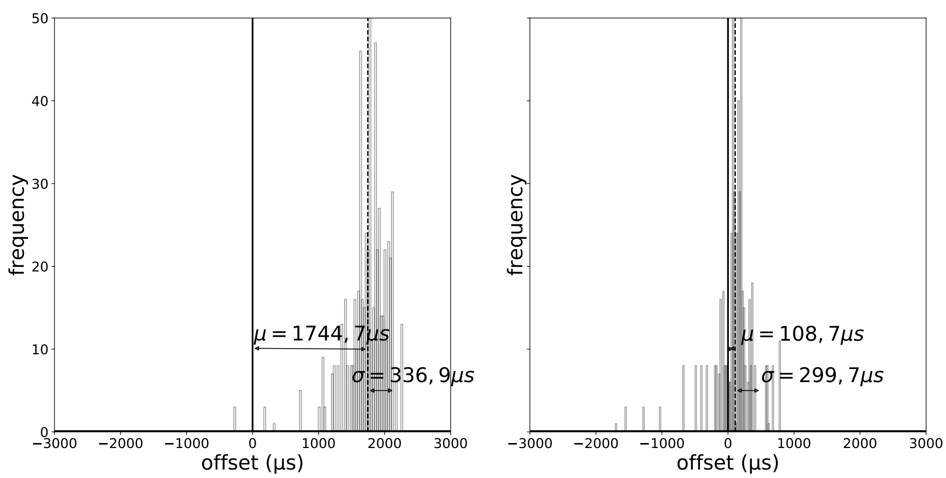


Figure 5. Histograms of the time offset for standard NTP (left) and calibrated NTP (right), at 9AM.

introduced by asymmetric paths in NTP time synchronization. Thus, we propose the following method for calibrating NTP:

1) *Calibration* – deploy the time box synchronized with GPS (like Ledi Network ATS) on given client premises. Measure the distributions of T_f and T_b for a given NTP server and find their d_f^{min} and d_b^{min}.
2) *Regular operation* – replace the time box with an NTP client without GPS for time synchronization. The client will operate according to the NTP protocol with a modified way of computing the time offset.
3) *Change detection* – detect changes in operating conditions with the `ping` and `traceroute` tools and redo calibration.

We assume that the NTP client sends n NTP requests and obtains n responses with corresponding timestamps.

- Find packets that experience the shortest transmission delays (called *lucky packets*):

$$j_f = \arg\min_j (t_2^j - t_1^j), \tag{10}$$

$$j_b = \arg\min_j (t_4^j - t_3^j), j = 1, ..., n, \tag{11}$$

and use their timestamps for computing the time offset.

- Use the estimates of minimum transmission times d_f^{min} and d_b^{min} in Eq. 4 for the time offset with the timestamps of packets j_f and j_b:

$$\theta = \frac{(t_2^{j_f} - d_f^{min} - t_1^{j_f}) + (t_3^{j_b} + d_b^{min} - t_4^{j_b})}{2}. \tag{12}$$

This expression corresponds to Eq. 4 with T_f^j and T_b^j replaced by d_f^{min} and d_b^{min}.

V. Validation

To evaluate the improvement in accuracy of the proposed method, we compare the accuracy of time synchronization obtained by standard NTP given by Eqs. 3 and 5 with proposed calibrated NTP measured with respect to the GPS time reference—we deploy the NTP client with the modified way of computing the time offset (Eqs. 10, 11, and 12) and compare the accuracy of its estimation with GPS.

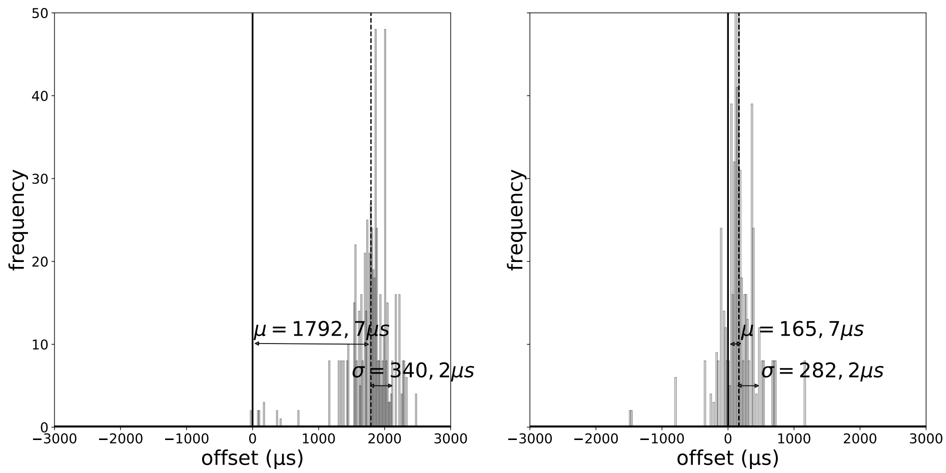


Figure 6. Histograms of the time offset for standard NTP (left) and calibrated NTP (right), 3PM.

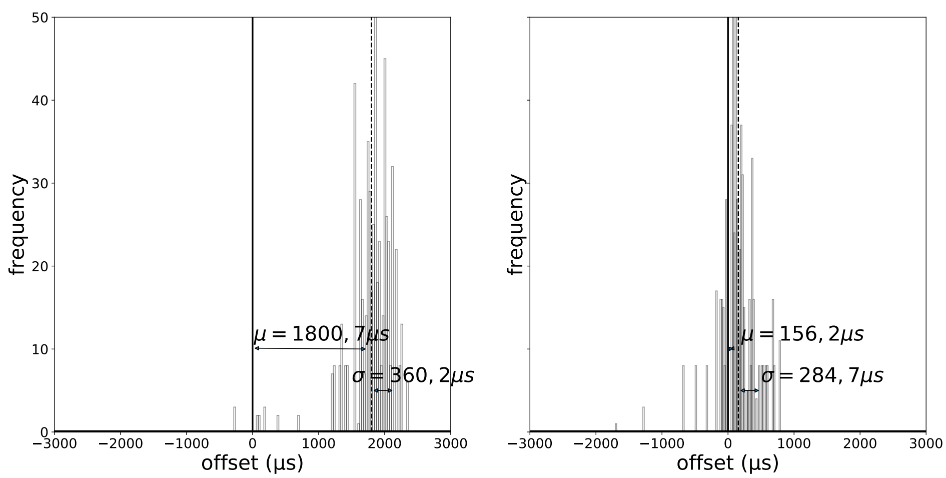


Figure 7. Histograms of the time offset for standard NTP (left) and calibrated NTP (right), 8PM.

The experiments took place at a given time of a day (9AM, 3PM, 8PM) with the measurement phase of one-way delay for 15 minutes during which the client sends trains of 8 NTP requests every 10 s followed by the measurements of the modified client with calibrated NTP during 15 minutes. Finally, we measured the performance of standard NTP also during 15 minutes. We repeated the experiments for several days and observed similar results.

Figures 5, 6, and 7 show the histograms of the time offset for standard NTP (left) and calibrated NTP (right) at 9AM, 3PM, and 8PM. The figures present *accuracy* measured by μ, the mean value of the distribution and *precision* measured by σ, the standard deviation. We can observe significant elimination of the accuracy error due to asymmetry of one-way delays and improved precision (lower σ).

VI. Related Work

Freris et al. [3] analyzed fundamental limits on synchronizing clocks over networks and showed that asymmetry cannot be measured only based on timestamps in a pairwise synchronization system even with an infinite number of round trip measurements. Lévesque and Tipper [10] surveyed the state of the art in clock synchronization over packet-switched networks and presented different mechanisms proposed to improve the synchronization accuracy of NTP and PTP.

Several authors considered exploiting delay characteristics to improve the precision of time synchronization by trying to estimate the difference in paths to take into account in the offset calculation. Tsuru et al. [5] considered the case of asymmetric delays due to the difference of bandwidth on the paths. The authors validated the scheme over an ADSL link with asymmetric bandwidth. The scheme does not address the case of asymmetric routes that we consider in our work.

The scope of our paper is NTP, nevertheless, PTP introduced some mechanisms to mitigate the negative effects of asymmetric links on synchronization accuracy [10]: residence time at intermediate nodes, D_{asym}, asymmetric delay parameter, and peer-to-peer path correction. We review some other proposed mechanisms for mitigating asymmetry in PTP below.

In a network with non PTP switches, Zarick et al. [16] measured synchronization accuracy as low as 450 μs under the presence of asymmetric delays. Lee [6] proposed to take into account asymmetric bandwidth based on link speed measurements. A slave initiates a block burst transmission at the master by an `Asymm_Check_Req` message, it measures the time interval for receiving the message burst from the master, and sends the burst back. The scheme allows to estimate the link speed ratio between the forward and backward path.

Schriegel et al. [7] characterized the variable delays of wireless links and used the characteristics to compensate non-deterministic forwarding delays of PTP synchronization frames by using different send rates for `Sync` messages. They evaluated the method in a real setup consisting of a Real-Time Ethernet with a wireless extension. Murakami and Horiuchi [17] proposed to add probing messages before and after `Sync` and `Delay_Req` messages in PTP to detect link utilization, which improved synchronization accuracy.

Exel [8] analyzed various asymmetry mitigation approaches for software timestamping and proposed a timestamp correction-based asymmetry compensation scheme that takes into account bandwidth asymmetry. He showed precision improvement with measurements using WLAN synchronization hardware.

Lévesque and Tipper [9] considered a PTP set up with PTP support on only a subset of nodes on the path between a client and a server. They proposed a probing-based mechanism to estimate asymmetry and improve the synchronization performance. The protocol is similar to that by Lee, but is lightweight and takes into account the per-packet control delays.

Hajikhani et al. [11] considered PTP in asymmetric packet-based networks and proposed a method to estimate the asymmetric random parts in one-way delays, however, they assumed that the constant parts of asymmetric delays are equal in both directions.

White Rabbit defines a calibration procedure [18] to estimate the asymmetry in fiber propagation latencies by connecting the WR master and the WR slave with oscilloscopes. The goal is similar to that of our scheme, but the procedure concerns the low physical layer with important measurement instrumentation.

VII. Conclusion

In this paper, we propose to calibrate NTP with the measurements of one-way transmission delay on the forward and backward path. Then, in the set up without the precise clock at the client, the NTP expression for the time offset takes into account asymmetry, which results in improved accuracy and precision.

Unlike many papers that use simulations, we have validated the proposed method by measurements of the clock offsets computed by standard NTP and calibrated NTP based on the GPS time reference showing significant improvement in accuracy and precision.

In future work, we plan to measure the stability of paths between a client and a server over longer time intervals to evaluate how frequently routing changes impact the operation of calibrated NTP.

The method may also benefit from maintaining several calibrated paths to multiple NTP servers. If the path to the primary server changes, the client can still use other NTP servers as backup. We plan to investigate this approach in future work.

Acknowledgments

This work has been partially supported by the French Ministry of Research project PERSYVAL-Lab under contract ANR-11-LABX-0025-01.

References

[1] D. Mills, "The Network Time Protocol (NTP)," RFC 958, 1985.

[2] D. Mills, J. Martin, and J. Burbank, "Network Time Protocol Version 4: Protocol and Algorithms Specification," RFC 5905, 2010.

[3] N. M. Freris, S. R. Graham, and P. R. Kumar, "Fundamental Limits on Synchronizing Clocks over Networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1352–1364, June 2011.

[4] T. Gotoh *et al.*, "Improvement of NTP Time Offset under the Asymmetric Network with Double Packets Method," in *IEEE CPEM*, June 2002, pp. 448–449.

[5] M. Tsuru *et al.*, "Estimation of Clock Offset from One-Way Delay Measurement on Asymmetric Paths," in *SAINT Workshops*, 2002.

[6] S. Lee, "An Enhanced IEEE 1588 Time Synchronization Algorithm for Asymmetric Communication Link using Block Burst Transmission," *IEEE Communications Letters*, vol. 12, no. 9, pp. 687–689, Sep. 2008.

[7] S. Schriegel *et al.*, "Enhancement for a Clock Synchronization Protocol in Heterogeneous Networks," in *IEEE ISPCS*, Oct 2009, pp. 1–5.

[8] R. Exel, "Mitigation of Asymmetric Link Delays in IEEE 1588 Clock Synchronization Systems," *IEEE Communications Letters*, vol. 18, no. 3, pp. 507–510, March 2014.

[9] M. Lévesque and D. Tipper, "Improving the PTP Synchronization Accuracy under Asymmetric Delay Conditions," in *IEEE ISPCS*, Oct 2015, pp. 88–93.

[10] ——, "A Survey of Clock Synchronization Over Packet-Switched Networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2926–2947, Fourthquarter 2016.

[11] M. J. Hajikhani, T. Kunz, and H. Schwartz, "A Recursive Method for Clock Synchronization in Asymmetric Packet-Based Networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2332–2342, Aug 2016.

[12] A. Pathak, H. Pucha, Y. Zhang, Y. C. Hu, and Z. M. Mao, "A Measurement Study of Internet Delay Asymmetry," in *ACM PAM*, 2008, pp. 182–191.

[13] Minalogic, "Gorgy Timing SCPTime Box," http://www.minalogic.com/en/product/box-scptime.

[14] G. Almes, S. Kalidindi, and M. Zekauskas, "A Round-trip Delay Metric for IPPM," RFC 2681, 1999.

[15] G. Almes, S. Kalidindi, M. Zekauskas, and A. Morton, "A One-Way Delay Metric for IP Performance Metrics (IPPM)," RFC 7679, 2016.

[16] R. Zarick, M. Hagen, and R. Bartoš, "The Impact of Network Latency on the Synchronization of Real-World IEEE 1588-2008 Devices," in *IEEE ISPCS*, Sep. 2010, pp. 135–140.

[17] T. Murakami and Y. Horiuchi, "Improvement of Synchronization Accuracy in IEEE 1588 Using a Queuing Estimation Method," in *IEEE ISPCS*, Oct 2009, pp. 1–5.

[18] G. Daniluk, "White Rabbit Calibration Procedure, Version 1.1," CERN BE-CO-HT, November 9, 2015.