IEEE TRANSACTIONS ON

INSTRUMENTATION AND MEASUREMENT

A PUBLICATION OF THE IEEE INSTRUMENTATION AND MEASUREMENT SOCIETY

IEEE INSTRUMENTATION & MEASUREMENT SOCIETY

DECEMBER 2015 VOLUME 64 NUMBER 12 IEIMAO (ISSN 0018-9456)

(Contents Continued on Page 3149)

(Contents Continued from Front Cover)

(Contents Continued on Page 3150)

(Contents Continued from Page 3149)

IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT (ISSN 0018-9456) is published monthly by The Institute of Electrical and Electronics Engineers, Inc. Responsibility for the contents rests upon the authors and not upon the IEEE, the Society/Council, or its members. **IEEE Corporate Office:** 3 Park Avenue, 17th Floor, New York, NY 10016-5997. **IEEE Operations Center:** 445 Hoes Lane, Piscataway, NJ 08854-4141. **NJ Telephone:** +1 732 981 0060. **Price/Publication Information**: Individual copies: IEEE Members $57.00 (first copy only), nonmembers $113.00 per copy. (Note: Postage and handling charge not included.) Member and nonmember subscription prices available upon request.

Periodicals Postage Paid at New York, NY and at additional mailing offices. **Postmaster:** Send address changes to IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, IEEE, 445 Hoes Lane, Piscataway, NJ 08854-4141. GST Registration No. 125634188. CPC Sales Agreement #40013087. Return undeliverable Canada addresses to: Pitney Bowes IMEX, P.O. Box 4332, Stanton Rd., Toronto, ON M5W 3J4, Canada. IEEE prohibits discrimination, harassment and bullying. For more information visit http://www.ieee.org/nondiscrimination. Printed in U.S.A.

Digital Object Identifier 10.1109/TIM.2015.2497401

Hough Transform-Based Clock Skew Measurement Over Network

Komang Oka Saputra, Wei-Chung Teng, *Member, IEEE*, and Tsung-Han Chen

***Abstract*— The accurate clock skew measurement of remote devices over network connections is crucial to device fingerprinting and other related applications. Current approaches use the lower bound of offsets between the target device and the measurer to estimate clock skew; however, the accuracy of estimation is severely affected when even a few offsets appear below the crowd of offsets. This paper adopted the Hough transform to develop a new method, which searches for the densest part of the whole distribution. This method is effective in filtering out the upper and lower outliers such that the skew values derived from the remaining offsets are stable, even when lower outliers occur, or when the measuring time is not long enough for current approaches to achieve stable results. The experimental evaluation of the proposed method has been conducted in order to compare its performance with that of linear programming algorithm (LPA) and two other approaches. During the five consecutive measurements of 1000 offsets each, skews of the proposed method varied within the range of 0.59 ppm, whereas LPA resulted in the range of 0.89 ppm. Both ranges increased to 1.34 and 63.93 ppm, respectively, when the lower bounds encountered interference from lower outliers.**

***Index Terms*— Clock skew, delay jitter, Hough transform, linear programming, low outlier.**

I. INTRODUCTION

CLOCK skew, or the clocking rate difference between two digital clocks, has been widely studied over the last few decades. Initially, researchers studied the clock skew effect that caused inaccurate delay measurement between the sender and the receiver. Earlier works [2], [4], [16], [19], [21], [29] focused on eliminating this effect and suggested methods for estimating clock skew. Furthermore, two properties of clock skew were identified [17], [19], [21]: the stability over time and the ability to distinguish between any two devices. These properties make clock skew a potential candidate for physical device fingerprinting and identification. For example, a pioneering work [17] used clock skew as a tool for fingerprinting computers in a general network. Other works [20], [27] utilized the clock skew in revealing a hidden service behind the onion router (TOR) network. Likewise, few studies [11]–[13], [18], [25] exploited the clock skew to secure time synchronization among sensor nodes in a wireless sensor network. Recently, usage of clock skew has been extended as an attack or defense instrument in various advanced technologies: 1) wireless local area network [3], [15]; 2) cloud environments [14]; 3) mobile handheld devices [23], [24]; and 4) smartphones [7]. Due to the importance of clock skew in many areas, it is crucial to pursue the accuracy of clock skew measurements.

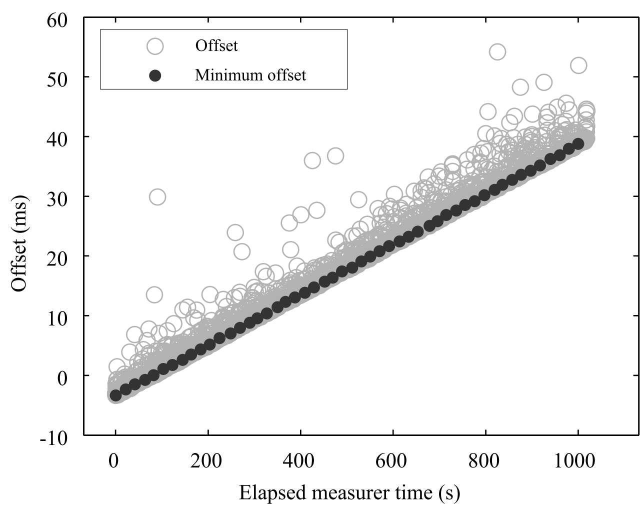

Fig. 1. Offset-set with a stable lower bound depicted by the minimum offsets.

A clock skew measurement is initialized by collecting timestamps sent from a device. The measurer can actively send ICMP requests to the device and collect timestamps from the response packets [7], [17], [23], [24]. Alternatively, the measurer can provide a service (e.g., Web application) by which devices communicate with it and send their timestamps through AJAX packets [14] or TCP timestamp options in TCP packets [23]. Fig. 1 shows a timestamp collection sample from an experiment, which will be introduced later in Section IV-A. In this scatter diagram, each point represents one received timestamp, and the offset of each point is calculated by subtracting the devices timestamp from the measurer's receiving time [14], [19]. After this, a line-fitting method such as linear regression can be used to estimate the slope or the clock skew of the whole collection [19], [21]. However, variable delays between the device and the measurer can make it difficult to determine the slope. Multipath routing and the accumulated propagation delay, especially in

Manuscript received January 26, 2015; revised April 20, 2015; accepted April 23, 2015. Date of publication July 21, 2015; date of current version November 6, 2015. This work was supported by the Ministry of Science and Technology under Grant NSC-101-2221-E-011-062-MY2. The Associate Editor coordinating the review process was Dr. Dario Petri.

K. Oka Saputra is with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10507, Taiwan, and also with the Department of Electrical and Computer Engineering, Udayana University, Bali 80361, Indonesia (e-mail: okasaputra@ee.unud.ac.id).

W.-C. Teng and T.-H. Chen are with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10507, Taiwan (e-mail: weichung@csie.ntust.edu.tw).

Digital Object Identifier 10.1109/TIM.2015.2450293

wireless communication, might significantly affect the end-to-end delay. The result is the production of many outlier offsets that depart from most offsets, located in the lower region of the scatter diagram. Consequently, the use of linear regression alone is not sufficient [14], [19], [21].

It is worth noting that various methods have been deployed to address this problem [2], [4], [14], [15], [16], [19], [21], [29]. For instance, Aoki *et al.* [2], Huang *et al.* [14], and Paxson [21] proposed outlier filtering methods by selecting the minimum offsets of the collected offsets. In the process, Paxson [21] finalized the outlier filter using the median line procedure, Aoki *et al.* [2] used the linear regression to calculate the slope of the accumulated minimum offsets selected from several minimum windows, and Huang *et al.* [14] used the quick piecewise minimum (QPM) algorithm to calculate the slope only from the minimum offsets in the first segment and the last segment of the collected offsets. However, the most widely adopted method is linear programming algorithm (LPA), as its result is not significantly affected by outliers. Moon *et al.* [19] are the pioneers of clock skew measurement by LPA and determine clock skew from the gradient of a line that lies below all the offsets.

All the above methods focused on achieving an accurate clock skew in the presence of outliers. Most of them fully utilized the characteristics of the offset collection: most offsets gather in a cluster, outlier offsets happen only above the cluster, and the bottom of the cluster can usually be bounded by a line whose slope is the clock skew. However, this paper observed that outliers below the cluster, referred to as the low outliers, may also occur in network connections with high jitter. In addition, the slopes of the lower bound lines affected by these low outliers tend to vary in a much larger range compared with those in classic cases. Therefore, this paper aims to develop a new estimation method for offset collections with low outliers.

The contribution of this paper is twofold. First, it introduces a new method based on the Hough transform: a method that combines the concept of clock skew and the Hough transform voting process. The goal of this method is to identify the parallelogram-like region that encloses the cluster of offsets, enabling the skew estimation to be derived from offsets that are undisrupted by the low outliers as well as the outliers. Second, the proposed method is used to improve the clock skew measurement when the lower bound is unstable due to the presence of the low outliers. The experimental results show that the proposed method always provides stable estimation, compared with three other methods: 1) LPA [19]; 2) QPM [14]; and 3) Aoki's method [2], especially in short-time measurement. To conclude, the proposed method is appropriate for all the existing applications related to clock skew, but it is especially suitable for applications like device fingerprinting, which requires stable estimation even in wireless yet high-jitter network connections.

The following section explains in detail why the lower bound concept is invalid to offset collections with low outliers. Section III introduces the proposed Hough transform-based clock skew measurement method. Next, the evaluation results and the comparison of the proposed method with existing ones are shown in Section IV. Section V presents several discussions related to the proposed method. The conclusion of this paper is given in Section VI.

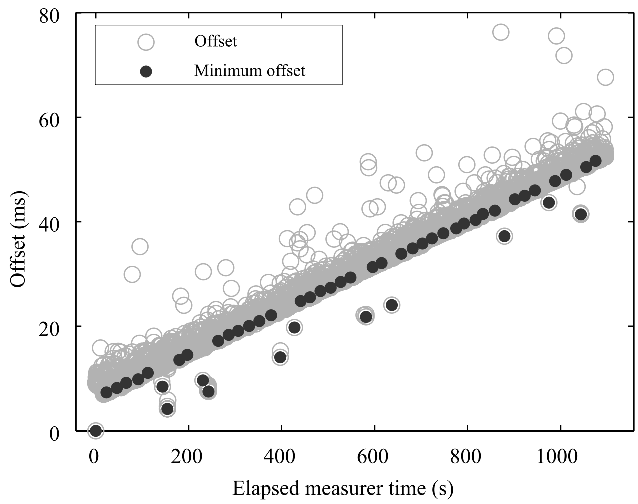

Fig. 2. Offset-set with an unstable lower bound due to the presence of low outliers.

II. Clock Skew Measurement and the Use of Lower Bound

To explain the concept of clock skew, this paper uses the terminology of [14] and [19]. For any real-world time t, the local time reported by the clock of device d is denoted by $C_d(t)$, and its first derivative, or the speed its clock progresses, is denoted by $C_d' \equiv dC_d(t)/dt, \quad \forall\ t \ \geq \ 0$. If C_m is denoted by the clock of the measurer, the skew δ of C_d can be obtained relative to C_m at time t by calculating $\delta(t) = C_d'(t) - C_m'(t)$. However, in practice, the measurer can only obtain the offset $C_m(t_2) - C_d(t_1)$, where t_1 is the sending time and t_2 is the receiving time. This situation is further aggravated by the fact that the difference between the sending time and the receiving time includes the communication delay such that the difference varies for each offset. To reach a stable and confident estimation of skew δ, one possible method is to pick offsets with very close, if not the same, communication delays. As explained in the previous section, it has been observed that offsets with minimal delays in the scatter diagram will most often line up after collecting a few hundred offsets. It is thus reasonable to assume that delays of these offsets are close to the physical minimum delay, and the slope of this straight line is very close to the real skew.

The use of the lower bound itself or its concept can be found in [17], [19], and [22]. As an example, the black circles in Fig. 1 represent offsets of minimal values in a sliding window (the local minimum), and it is obvious that these black circles step up slowly but constantly as time goes by. Without being affected by the outliers above, the lower bound line keeps extending in the same slope from the beginning to the end. Nevertheless, there are cases where the lower bound line breaks up, as shown in Fig. 2 (a sample that will be analyzed later in Section IV-B). Even though the offsets under the lower bound line are still close to it, these anomalies are marked as low outliers. Given a long period, it is still

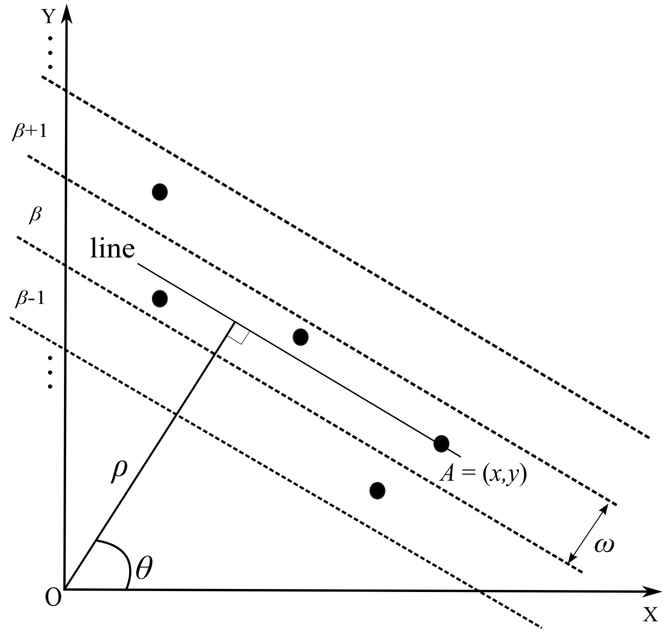

Fig. 3. Relations between ρ, θ, ω, and β in the Hough transform.

possible to obtain a constant lower bound line. For short-time measurement, however, parts of the local minimal offset fall below the others irregularly, which makes the calculated clock skew fluctuate each time.

It is clear that the existing approaches [2], [14], [19], [21] that utilize the minimum offsets to obtain accurate clock skews become invalid in the context of cases like Fig. 2. It is also a challenging task to filter out the low outliers from the concentrated area, because some low outliers are so close to the area, and normally, they should be categorized as part of the cluster of offsets. This paper abandons the use of the lower bound and instead obtains the clock skew, the slope of a line, from the concentrated area itself. As the concentrated area in normal cases like Figs. 1 and 2 forms the shape of a thick line, it is argued that this approach is promising for deriving accurate skews.

III. Hough Transform-Based Clock Skew Measurement

A. Hough Transform

The Hough transform is a commonly used technique to identify lines from an image [9], [10]. The conventional Hough transform maps the Euclidean coordinate of each dot in the image into another parameter space to quickly find the candidate lines. Fig. 3 explains the parameters used by the Hough transform in finding a straight line from a point set. For point $A = (x, y)$ and a given degree θ, there is exactly one line that passes through point A, and the angle between the x-axis and its normal line is θ. Here, the perpendicular distance from the origin to this line is denoted by ρ. The relations between these variables can thus be summarized as (1). In fact, each (ρ, θ) pair defines one unique candidate line

$$\rho = x * \cos\theta + y * \sin\theta. \tag{1}$$

To determine the straight lines from many candidates, the Hough transform uses a voting mechanism by which each point votes for all possible lines that pass through it, and the vote results are represented in the form of a matrix indexed by (ρ, θ). The most possible lines are those cells with more votes than a given threshold [5].

Unlike the conventional Hough transform, the goal of this paper is to search for a parallelogram-like region instead of a thin straight line. Thus, the result of this approach will be θ, a lower bound, and an upper bound of distance ρ, but not a specific (ρ, θ) pair. One way to denote this region is (ρ_l, ρ_u, θ), where $\rho_l \le \rho_u$. Values of ρ_l and ρ_u result from the clock difference and transmission delay, but the *thickness* $\omega = \rho_u - \rho_l$ is basically determined by the jitter. Thus, a new algorithm must be developed to search for the appropriate thickness.

B. Offset Voting for Candidate Regions

Before the new algorithm is designed, the offsets in a scatter diagram like Fig. 1 must be mapped into image points. All the points are mapped to an image plane with nonnegative coordinates. This is realized by choosing an appropriate position in the scatter diagram as the origin of the image. If the offsets are numbered by the order of their receiving time, then the coordinate of the ith offset can be expressed as (t_i, o_i), where t_i is the receiving time of the measurer and o_i is the offset value. Furthermore, let $o_{\min}$ represent the minimum value of all the offsets, then the origin chosen is at $(t_1, o_{\min})$, and the mapping can be expressed as $(x_i, y_i) = (t_i - t_1, o_i - o_{\min})$.

To find two parallel lines with the same θ accurately, Algorithm 1 was developed based on the voting process of the Hough transform in [5]. In Algorithm 1, S stands for the set of all the points; $\theta_{\min}$ and $\theta_{\max}$ bound the range of angle for line detection; p stands for the step size in radians, or the time precision sought; $\omega_{\min}$ is the lower bound of the thickness sought in the voting process; and ω_{inc} is the increment by which the thickness grows. Finally, N stands for the minimum number of points a candidate region has to cover. To start the algorithm, some variables must be declared: L is used to store all the found candidate regions, and β is the index of the regions. Given some angle θ and some thickness ω, the whole 2-D space can be divided into many regions by parallel lines with the distance between adjacent lines being ω. Starting with a line of slope θ passing though the origin, the regions can be numbered starting from 0, as shown in Fig. 3. In other words, β is the rounded value if ρ is rounded down to an integer multiple of ω. The Votes(ρ, θ) used in the Hough transform then becomes Votes(β, ω, θ) accordingly. Algorithm 1 starts by setting the thickness to $\omega_{\min}$. The following *while* loop is used to find an eligible region of some thickness in each round. Inside the *while* loop is a Hough transform-like process: each point in S votes for all possible θ between $\theta_{\min}$ and $\theta_{\max}$. If there are one or more angle-region tuples whose votes are at least N, then the tuple with the highest vote is stored in L, the returned value of function OffsetVote(). If there is no candidate found, then the thickness ω is increased by ω_{inc}, and a new voting round is started.

It is useful to first try a minimum thickness and gradually increase it if necessary, but not the other way around. When the thickness is small, only the area of the highest density

Algorithm 1 Offset Voting Algorithm

```
function OffsetVote S, θ_min, θ_max, p, ω_min, ω_inc, N
    Votes = 0
    L = null
    ω = ω_min
    while L == null do
        for all (x, y) in S do
            for θ = θ_min; θ ≤ θ_max; θ = θ + p do
                ρ = x * cosθ + y * sinθ
                β = ⌊ρ / ω⌋
                Votes(β, θ) = Votes(β, θ) + 1
            end for
        end for
        for all (β, θ) do
            if Votes(β, θ) ≥ N then
                if L == null or
                    Votes(β, θ) > L.Votes then
                    L = (β, ω, θ, Votes(β, θ))
                end if
            end if
        end for
        if L == null then
            ω = ω + ω_inc
        end if
    end while
    return L
end function
```

Algorithm 2 Three-Stage Process

```
Require: S, ω_min, ω_inc, N
p = 10^-5
θ_min = (π / 2) − 750 * 10^-6
θ_max = (π / 2) + 750 * 10^-6
L = OffsetVote(S, θ_min, θ_max, p, ω_min, ω_inc, N)
θ_min = L.θ − 5 * 10^-6
θ_max = L.θ + 5 * 10^-6
p = p / 10
L = OffsetVote(S, θ_min, θ_max, p, ω_min, ω_inc, N)
θ_min = L.θ − 5 * 10^-7
θ_max = L.θ + 5 * 10^-7
p = p / 10
L = OffsetVote(S, θ_min, θ_max, p, ω_min, ω_inc, N)
return (L.β, L.ω, L.θ)
```

will be chosen, but the enclosed number of offsets may not be enough to prove the majority. Conversely, a large thickness will risk including low outliers and normal outliers in the area. Appropriate minimum thickness and increment values will save computation time, yet still ensure accurate results. This paper used a 500-μs minimum thickness ($\omega_{\min}$) and a 100-μs increment (ω_{inc}) in all the experiments. This configuration means that OffsetVote() will start the voting process from an initial thickness of 500 μs. If no candidate region is found in the second **for** loop, the thickness to try (ω) increases by 100 μs at the end of the **while** loop. Thus, the value of the final thickness stored in L can be expressed as $500 + 100k$, where k is a nonnegative integer. On the other hand, the minimum number of points N serves as a threshold to ensure the representativeness of the elected regions. If the value of N is too small, OffsteVote() will stop early with an undesirably small thickness, and the angle of the chosen region can obviously differ from the clock skew. The value of N should be at least 35% of the number of offsets, even in high-jitter cases. In this paper, 50% was used as the threshold.

C. Three-Stage Processing

One of the advantages of the Hough transform is that it can detect lines of all the directions. To save computation time, however, most applications of the Hough transform use one degree or larger resolution [9], [10]. In contrast, clock skew measurement requires precision of at least ppm level [15], [17]. If the parameter p in OffsetVote() is set to 10^{-6}, this function will have to execute voting one million times in the 1° range, given any thickness ω. To keep the computation time short, it is vital to limit the range of angle by adopting reasonable $\theta_{\min}$ and $\theta_{\max}$. Of all the reviewed studies, Jana and Kasera [15] reported the highest clock skew of 1105.69 ppm when they measured several access points in their experiments. The second highest number is 750 ppm according to [7]. Since a difference of 750 ppm means that two clocks will differ by more than 30 min/month, it is really a rare case. In fact, most of the observed skews fall into the range of −200 to 200 ppm [2]. Incidentally, a measured skew will become its additive inverse if the target device and the measurer are switched ($\text{skew}_{\text{AB}} = -\text{skew}_{\text{BA}}$) [3], [12], so it is practical to assume that a clock skew of −750 ppm is as possible as one of 750 ppm. This paper suggests first choosing a maximum skew value, and then using its additive inverse as the minimum skew value.

A clock skew value can be viewed as the slope of a straight line, and this slope s can be further expressed as a fraction y/x. On the other hand, if the counter-clockwise angle from the x-axis to this line is a, then $s = \sin a/\cos a = \tan a$. Since s is a very small number, s can be used as the approximation value of a. Finally, θ in function OffsetVote() is the angle of normal vector, so the values of $\theta_{\min}$ and $\theta_{\max}$ are $\pi/2$ added to the minimum and the maximum clock skew, respectively. If −750 and 750 ppm are used as the minimum and maximum clock skews, respectively, the range to angle will become 1.570046–1.571546 rad, or 89.96°–90.04°, a much smaller range than the 0°–180° in the conventional Hough transform voting process. However, there are still 1501 angles to test if the 1-ppm resolution is used, or 15 001 angles for the 0.1-ppm resolution. To solve this problem, this paper designed a three-stage process as shown in Algorithm 2.

Function OffsetVote(), or Algorithm 1, is called once in each stage. In the first stage, OffsetVote() scans the whole angle range in 10 ppm units to find the region with the highest votes and store the result in variable L. The second stage switches to 1-ppm scan level, but OffsetVote() only scans the angle range of 5 ppm near the angle of region L this time. The third stage is basically a copy of the second stage, but it uses a 0.1-ppm scan level over the range of 0.5 ppm near the angle of region L. In this way, the number of angles that OffsetVote() evaluates sharply decreases from 15 001 to 173. This process succeeds because all the offsets are considered to line up roughly in only one direction, so the region found in the first stage must have the closest clock skew in the 10-ppm resolution. Stages 2 and 3 simply pursue higher resolutions. The outcome of Algorithm 2 is the index, the thickness, and the angle of the region in the 0.1-ppm precision, or a parallelogram-like region that covers the most offsets.

IV. Evaluation Results

This paper evaluated the correctness and robustness of the proposed measurement using both normal and low-outlier conditions, as shown in Figs. 1 and 2. For each condition, the result of the proposed method was compared with three other methods: 1) LPA [19]; 2) QPM [14]; and 3) Aoki's method [2]. One long-term sample was initially used to generate a reliable estimation of clock skew, and then this sample set was cut into several segments of short but equal durations. The short-term skews calculated from these segments revealed how skew would fluctuate as time went by. The skews of accumulated offsets and separate skews were further calculated to observe how the estimated skew converged.

Both the normal and the low-outlier samples were taken using the same devices. Therefore, the clock skews in both samples should be the same, and the measurement methods would be expected to produce relatively similar clock skews for both samples. An ASUS A46C notebook computer with Microsoft Windows 7 operating system was used as the target device and another ASUS DUO T9300 notebook running Ubuntu 12.04 operating system as the measurer. In each measurement, the target device sent 5000 timestamps at 200-ms intervals, which took approximately 17 min.

A. Estimation on an Offset-Set Without Low Outlier

The result of long-term measurement by the proposed method is detailed in Table I. The final result in the third stage, angle $\theta = 1.5708382$ rad with thickness $\omega = 500$ μs, was used to create a bounded region as shown in Fig. 4. By applying linear regression on only the bounded offsets, an estimated clock skew of 42.03 ppm was obtained. This result is relatively close to those of LPA (42.04 ppm), of QPM (42.16 ppm), and of Aoki's method (42.06 ppm).

To compare the results of these four methods in short-term measurement, Tables II and III summarize the skews of accumulated offsets and separate skews of short-term segments, respectively. In Table II, 500 offsets increment were used to calculate all the skews. As expected, LPA and the proposed method contributed stable estimations. It is clear from the row Max–Min that the proposed method provides more stable estimation compared with QPM and Aoki's method. By comparing the clock skew column and the $\theta - \pi/2$ column, it can be observed that the Hough transform is suitable to search for the bounded region, but other analytic tools are still required, like linear regression, to refine the clock skew estimation. Finally, the skews of accumulated offsets of all the four methods do not differ more than 0.5 ppm in the cases of 2000 or more offsets.

Table III shows separate skews of the four methods in 1000 offset segments. Since all the outliers are excluded at first, the proposed method is able to contribute even more stable estimation, on average, than LPA.

TABLE I
Results of the Three-Stage Process on an Offset-Set Without Low Outlier

First-stage process	
θ range	1.570046 rad to 1.571546 rad
θ step-size	10^{-5} rad
Number of voted θ	755
Result (θ, ω)	1.5708360 rad, 900 μs
Second-stage process	
θ range	1.5708310 rad to 1.5708410 rad
θ step-size	10^{-6} rad
Number of voted θ	11
Result (θ, ω)	1.5708380 rad, 500 μs
Third-stage process	
θ range	1.5708375 rad to 1.5708385 rad
θ step-size	10^{-7} rad
Number of voted θ	11
Result (θ, ω)	1.5708382 rad, 500 μs
θ - π/2	41.9 ppm
Clock skew	42.03 ppm

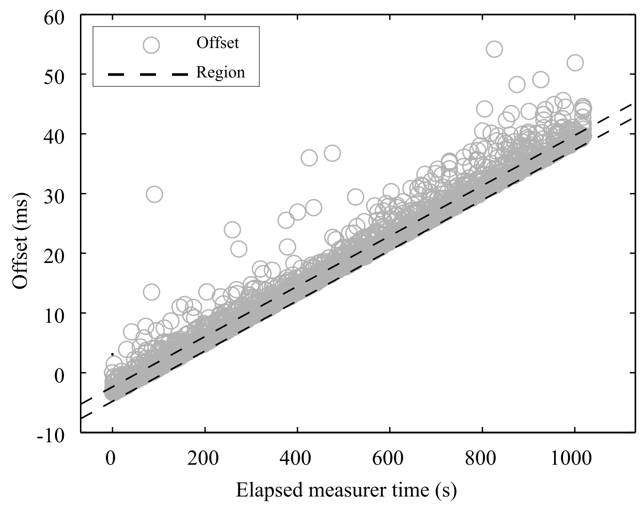

Fig. 4. Enclosed region for an offset-set without low outlier, where $\theta = 1.5708382$ rad and $\omega = 500$ μs.

B. Estimation on an Offset-Set With Low Outliers

The result of long-term measurement by the proposed method is detailed in Table IV. The final result of the third stage, angle $\theta = 1.5708387$ rad with thickness $\omega = 700$ μs, was used to create a bounding region as shown in Fig. 5. By applying linear regression on only the bounded offsets, an estimated clock skew of 42.29 ppm was obtained, which was very close to the result of the previous measurement. As the offset distribution was not as dense as the previous

TABLE II
SKEWS OF ACCUMULATED OFFSETS ON AN OFFSET-SET WITHOUT LOW OUTLIER

Offset	LPA (ppm)	QPM (ppm)	Aoki's method (ppm)	Proposed method			
				θ (rad)	ω (μs)	θ - $\pi/2$ (ppm)	Clock skew (ppm)
500	42.03	41.13	40.99	1.5708379	500	41.6	41.84
1000	42	42.69	41.49	1.5708390	500	42.7	42.46
1500	42.29	41.88	41.89	1.5708380	500	41.7	42.18
2000	42.6	42.16	42.25	1.5708380	500	41.7	42.16
2500	42.18	41.93	42.23	1.5708385	500	42.2	42.22
3000	42.04	42.27	42.23	1.5708389	500	42.6	42.28
3500	42.04	42	42.06	1.5708388	500	42.5	42.23
4000	42.04	42.02	42.07	1.5708387	500	42.4	42.17
4500	42.04	41.97	42.07	1.5708381	500	41.8	41.98
5000	42.04	42.16	42.06	1.5708382	500	41.9	42.03
Max	42.6	42.69	42.25			42.7	42.46
Min	42	41.13	40.99			41.6	41.84
Average	42.13	42.02	41.93			42.1	42.16
Max - Min	0.6	1.56	1.26			1.1	0.62

TABLE III
SEPARATE SKEWS ON AN OFFSET-SET WITHOUT LOW OUTLIER

Segment	LPA (ppm)	QPM (ppm)	Aoki's method (ppm)	Proposed method			
				θ (rad)	ω (μs)	θ - $\pi/2$ (ppm)	Clock skew (ppm)
1–1000	42	42.69	41.49	1.5708390	500	42.7	42.46
1000–2000	42.50	43.49	42.78	1.5708382	500	41.9	42.20
2000–3000	41.66	41.58	42.61	1.5708388	500	42.5	42.08
3000–4000	42.14	42.74	42.52	1.5708386	500	42.3	41.99
4000–5000	41.61	41.83	41.71	1.5708385	500	42.2	41.87
Max	42.50	43.49	42.78			42.7	42.46
Min	41.61	41.58	41.49			41.9	41.87
Average	41.98	42.47	42.22			42.3	42.21
Max - Min	0.89	1.91	1.29			0.8	0.59

TABLE IV
RESULTS OF THE THREE-STAGE PROCESS ON AN OFFSET-SET WITH LOW OUTLIERS

First-stage process	
θ range	1.570046 rad to 1.571546 rad
θ step-size	10^{-5} rad
Number of voted θ	906
Result (θ, ω)	1.5708360 rad, 1000 μs
Second-stage process	
θ range	1.5708310 rad to 1.5708410 rad
θ step-size	10^{-6} rad
Number of voted θ	44
Result (θ, ω)	1.5708390 rad, 800 μs
Third-stage process	
θ range	1.5708385 rad to 1.5708395 rad
θ step-size	10^{-7} rad
Number of voted θ	33
Result (θ, ω)	1.5708387 rad, 700 μs
θ - $\pi/2$	42.4 ppm
Clock skew	42.29 ppm

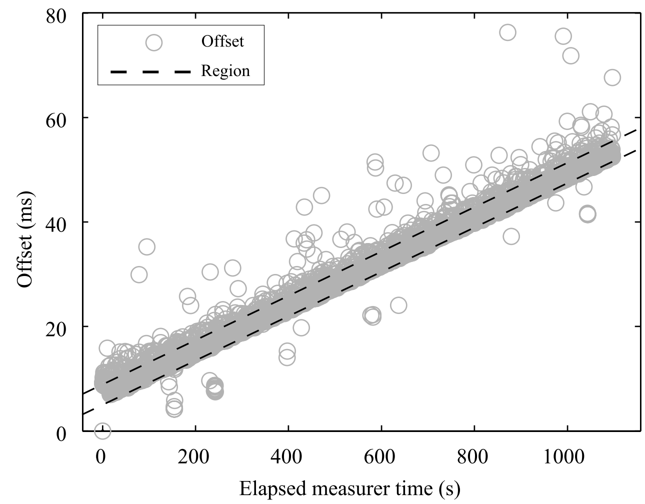

Fig. 5. Enclosed region for an offset-set with low outliers, where $\theta = 1.5708387$ rad and $\omega = 700$ μs.

TABLE V
SKEWS OF ACCUMULATED OFFSETS ON AN OFFSET-SET WITH LOW OUTLIERS

Offset	LPA (ppm)	QPM (ppm)	Aoki's method (ppm)	Proposed method			
				θ (rad)	ω (μs)	θ - $\pi/2$ (ppm)	Clock skew (ppm)
500	33.76	39.20	39.14	1.5708374	900	41.1	41.52
1000	27.05	43.23	26.15	1.5708376	900	41.3	41.29
1500	37.89	43.11	31.40	1.5708392	800	42.9	42.67
2000	37.89	18.62	34.56	1.5708391	800	42.8	42.59
2500	42.29	31.17	42.07	1.5708378	800	41.5	41.67
3000	41.91	41.20	39.46	1.5708383	700	42	41.98
3500	41.91	41.20	42.46	1.5708382	700	41.9	41.94
4000	41.91	58.73	43.59	1.5708385	700	42.2	42.15
4500	41.91	45.62	42.94	1.5708385	700	42.2	42.16
5000	41.91	45.62	42.62	1.5708387	700	42.4	42.29
Max	42.29	58.73	43.59			42.9	42.67
Min	27.05	18.62	26.15			41.1	41.29
Average	38.84	40.77	38.44			42.03	42.03
Max - Min	15.24	40.11	17.44			1.8	1.38

TABLE VI
SEPARATE SKEWS ON AN OFFSET-SET WITH LOW OUTLIERS

Segment	LPA (ppm)	QPM (ppm)	Aoki's method (ppm)	Proposed method			
				θ (rad)	ω (μs)	θ - $\pi/2$ (ppm)	Clock skew (ppm)
1–1000	27.05	43.23	26.15	1.5708376	900	41.3	41.29
1000–2000	42.29	42.29	45.27	1.5708391	800	42.8	42.63
2000–3000	-21.64	-3.96	11.19	1.5708383	700	42	41.85
3000–4000	41.76	41.83	41.41	1.5708387	900	42.4	42.14
4000–5000	25.36	25.36	43.74	1.5708386	700	42.3	42.13
Max	42.29	43.23	45.27			42.8	42.63
Min	-21.64	-3.96	11.19			41.3	41.29
Average	22.96	29.74	33.55			42.2	42
Max - Min	63.93	47.19	34.08			1.5	1.34

one, a larger ω was necessary to enclose a sufficient number of offsets.

Similar to Table II, Table V lists the skews of accumulated offsets obtained using these four methods. Compared with the long-term result of the proposed method, LPA gives a stable result of 41.91 ppm, Aoki's method has a somewhat inaccurate estimation of 42.62 ppm, and QPM produces an unacceptable result of 45.62 ppm. Observing how the estimated skews converge in Table V, it is surprising that even LPA is significantly affected by the low outliers until the number of offsets accumulates to 2500. The skew difference between the maximum and the minimum is 15.24 ppm, and this value is still superior to those of QPM and Aoki's method. In contrast, the skews of accumulated offsets by the proposed method do not converge notably because the skews fluctuate within the range of only 1.38 ppm. It is clear that the proposed method is able to provide much more stable results in short-term data.

It becomes more obvious that the proposed method has an advantage over others when the separate short-term skews in Table VI are compared. Table VI shares the same format

TABLE VII

SOME RESULTS ON MEASURING CLOCK SKEWS OF THREE DIFFERENT DEVICES

Device	Sending interval (ms)	Number of offsets	θ (rad)	ω (μs)	Number of low-outliers	Number of bounded offsets	Clock skew (ppm)
ASUS A46C Notebook	200	5000	1.5708387	700	103	2870	42.29
	500	3000	1.5708385	600	97	1810	42.09
	1000	1000	1.5708382	700	56	542	41.85
HTC OneX Mobile device	200	5000	1.5707613	700	209	2509	-34.84
	500	3000	1.5707616	600	177	1741	-34.66
	1000	1000	1.5707619	900	58	503	-34.24
iPad 2 Tablet device	200	5000	1.5707864	800	522	2545	-9.35
	500	3000	1.5707866	600	255	1727	-9.79
	1000	1000	1.5707860	600	78	510	-9.24

TABLE VIII

COMPARISON OF COMPUTATION TIME AND ACCURACY BETWEEN FULL SET RESULTS AND HEAD-AND-TAIL RESULTS

Offset	All offsets are used					Only beginning and ending parts are used				
	θ (rad)	ω (μs)	θ - π/2 (ppm)	Clock skew (ppm)	Computation time (sec.)	θ (rad)	ω (μs)	θ - π/2 (ppm)	Clock skew (ppm)	Computation time (sec.)
1000	1.5708376	900	41.3	41.29	8.34	1.5708376	900	41.3	41.29	8.34
1500	1.5708392	800	42.9	42.67	11.29	1.5708392	800	42.9	42.67	8.42
2000	1.5708391	800	42.8	42.59	11.36	1.5708391	800	42.8	42.59	9.18
2500	1.5708378	800	41.5	41.67	15.85	1.5708378	800	41.5	41.67	8.94
3000	1.5708383	700	42	41.98	23.24	1.5708385	700	42.2	42.12	8.95
3500	1.5708382	700	41.9	41.94	33.82	1.5708379	800	41.6	41.73	8.43
4000	1.5708385	700	42.2	42.15	38.41	1.5708387	900	42.4	42.28	9.24
4500	1.5708385	700	42.2	42.16	46.84	1.5708386	700	42.3	42.56	8.94
5000	1.5708387	700	42.4	42.29	46.99	1.5708386	700	42.3	42.22	8.9
Max			42.9	42.67				42.9	42.67	
Min			41.3	41.29				41.3	41.29	
Average			42.13	42.08				42.14	42.12	
Max - Min			1.6	1.38				1.6	1.38	

with Table III, but reveals more information. Skews by LPA happen to vary more significantly than the other methods. It is thus observed that LPA is vulnerable to the cases in which low outliers occur only near the middle of the whole measuring period. Finally, short-term skews by the proposed method fluctuate within the range of only 1.34 ppm, which is much smaller than 63.93 ppm by LPA, 47.19 ppm by QPM, and 34.08 ppm by Aoki's method.

Despite the use of the ASUS notebook, low outliers were also observed when the skews of other devices were measured. Table VII shows a few results from the ASUS notebook and two other devices as a reference. The number of low outliers in Table VII is defined as the number of offsets below the bounding region.

V. COMPUTATIONAL COMPLEXITY

The previous sections demonstrated the use of developed techniques and evaluated the robustness of the proposed method, namely, the three-stage Hough transform-based measurement. To analyze the time complexity of this method, some variables must be defined first. Let n denote the number of measured offsets, and ω_1, ω_2, and ω_3 be the counts of thickness tried at Stages 1–3, respectively. The first *for* loop in OffsetVote() is clearly $O(n)$, and the second *for* loop does not take more than $O(n)$ time. Thus, the total computation time is $O((151\omega_1 + 11\omega_2 + 11\omega_3)n)$, which is still $O(n)$.

In the case of 5000 offsets, it takes 47 s to compute the clock skew by a computer with a 2-GHz RAM and an Intel Core-i7 processor. There are two possible ways to reduce the computation time. The first is to choose a shorter angle range in the first stage, like $[\pi/2 - 200 \times 10^{-6}, \pi/2 + 200 \times 10^{-6}]$. For the second method, previous studies [17], [19], [21] have revealed that the clock skew of an offset-set is constant from the beginning to the end of the measurement. Thus, only the beginning and the ending parts of all the offsets may be used to reach very close results. Table VIII compares the computation time and accuracy between the full set results and their head-and-tail versions. The left-hand side of Table VIII is copied from the part of Table V, and the right-hand side of Table VIII is derived from the results using only the beginning 500 offsets and the ending 500 offsets. As expected, the computation time of the full set results grows in a linear fashion as the number of offsets increases. However, the time used in head-and-tail version remains within a small range in all the experiments. On the other hand, the measured skews in both versions are basically of the same value. The biggest difference between two versions is 0.4 ppm in the 4500 offsets case.

VI. CONCLUSION

This paper proposed a new method for estimating the clock skews of remote devices with a network connection. The advantage of this method is that it provides stable estimations, especially when the number of timestamps is limited to a few hundreds. It also reported the existence of low outliers, outliers below the crowd of offsets, which may be caused by high-jitter or other issues. Low outliers make the lower bound of an offset-set unstable, and thus severely interfere with the estimations by existing approaches. Since the proposed method aims to find a parallelogram-like region that encloses the densest part of the distribution, it is not affected by these low outliers. To conclude, the proposed method is the most robust way to measure clock skews in only a few minutes.

ACKNOWLEDGMENT

The authors would like to thank L.-C. Cheng for developing the initial system and for conducting preliminary experiments in this paper.

REFERENCES

[1] T. E. Abrudan, A. Haghparast, and V. Koivunen, "Time synchronization and ranging in OFDM systems using time-reversal," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 12, pp. 3276–3290, Dec. 2013.

[2] M. Aoki, E. Oki, and R. Rojas-Cessa, "Measurement scheme for one-way delay variation with detection and removal of clock skew," *ETRI J.*, vol. 32, no. 6, pp. 854–862, Dec. 2010.

[3] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz, "On the reliability of wireless fingerprinting using clock skews," in *Proc. 3rd ACM Conf. Wireless Netw. Security*, 2010, pp. 169–174.

[4] J. Bi, Q. Wu, and Z. Li, "On estimating clock skew for one-way measurements," *Comput. Commun.*, vol. 29, no. 8, pp. 1213–1225, May 2006.

[5] Z.-H. Chen, A. W. Y. Su, and M.-T. Sun, "Resource-efficient FPGA architecture and implementation of Hough transform," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 8, pp. 1419–1428, Aug. 2012.

[6] T. Cooklev, J. C. Eidson, and A. Pakdaman, "An implementation of IEEE 1588 over IEEE 802.11b for synchronization of wireless local area network nodes," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 5, pp. 1632–1639, Oct. 2007.

[7] M. Cristea and B. Groza, "Fingerprinting smartphones remotely via ICMP timestamps," *IEEE Commun. Lett.*, vol. 17, no. 6, pp. 1081–1083, Jun. 2013.

[8] C. M. De Dominicis, P. Pivato, P. Ferrari, D. Macii, E. Sisinni, and A. Flammini, "Timestamping of IEEE 802.15.4a CSS signals for wireless ranging and time synchronization," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 8, pp. 2286–2296, Aug. 2013.

[9] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[10] P. E. Hart, "How the Hough transform was invented [DSP History]," *IEEE Signal Process. Mag.*, vol. 26, no. 6, pp. 18–22, Nov. 2009.

[11] D.-J. Huang, W.-C. Teng, C.-Y. Wang, H.-Y. Huang, and J. M. Hellerstein, "Clock skew based node identification in wireless sensor networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov./Dec. 2008, pp. 1–5.

[12] D.-J. Huang and W.-C. Teng, "A defense against clock skew replication attacks in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 39, pp. 26–37, Mar. 2014.

[13] D.-J. Huang, W.-C. Teng, and K.-T. Yang, "Secured flooding time synchronization protocol with moderator," *Int. J. Commun. Syst.*, vol. 26, no. 9, pp. 1092–1115, 2013.

[14] D.-J. Huang, K.-T. Yang, C.-C. Ni, W.-C. Teng, T.-R. Hsiang, and Y.-J. Lee, "Clock skew based client device identification in cloud environments," in *Proc. 26th IEEE Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2012, pp. 526–533.

[15] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," *IEEE Trans. Mobile Comput.*, vol. 9, no. 3, pp. 449–462, Mar. 2010.

[16] H. Khlifi and J.-C. Grégoire, "Low-complexity offline and online clock skew estimation and removal," *Comput. Netw.*, vol. 50, no. 11, pp. 1872–1884, Aug. 2006.

[17] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Trans. Dependable Secure Comput.*, vol. 2, no. 2, pp. 93–108, Apr./Jun. 2005.

[18] X. Mei, D. Liu, K. Sun, and D. Xu, "On feasibility of fingerprinting wireless sensor nodes using physical properties," in *Proc. 27th IEEE Int. Symp. Parallel Distrib. Process. (IPDPS)*, May 2013, pp. 1112–1121.

[19] S. B. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proc. INFOCOM Conf.*, Mar. 1999, pp. 227–234.

[20] S. J. Murdoch, "Hot or not: Revealing hidden services by their clock skew," in *Proc. 13th ACM Conf. Comput. Commun. Security*, 2006, pp. 27–36.

[21] V. Paxson, "On calibrating measurements of packet transit times," in *Proc. ACM SIGMETRICS Conf.*, 1998, pp. 11–21.

[22] L. Polcák, J. Jirásek, and P. Matousek, "Comment on 'remote physical device fingerprinting,'" *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 5, pp. 494–496, Sep./Oct. 2014.

[23] S. Sharma, H. Saran, and S. Bansal, "An empirical study of clock skew behavior in modern mobile and hand-held devices," in *Proc. 3rd Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2011, pp. 1–4.

[24] S. Sharma, A. Hussain, and H. Saran, "Experience with heterogenous clock-skew based device fingerprinting," in *Proc. LASER Workshop*, 2012, pp. 9–18.

[25] M. B. Uddin and C. Castelluccia, "Toward clock skew based wireless sensor node services," in *Proc. 5th Annu. ICST Wireless Internet Conf. (WICON)*, 2010, pp. 1–9.

[26] A. Vakili and J.-C. Grégoire, "Accurate one-way delay estimation: Limitations and improvements," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 9, pp. 2428–2435, Sep. 2012.

[27] S. Zander and S. J. Murdoch, "An improved clock-skew measurement technique for revealing hidden services," in *Proc. 17th Conf. Security Symp.*, 2008, pp. 211–225.

[28] K. Zeng, K. Govindan, and P. Mohapatra, "Non-cryptographic authentication and identification in wireless networks [Security and Privacy in Emerging Wireless Networks]," *IEEE Wireless Commun.*, vol. 17, no. 5, pp. 56–62, Oct. 2010.

[29] L. Zhang, Z. Liu, and C. H. Xia, "Clock synchronization algorithms for network measurements," in *Proc. INFOCOM Conf.*, 2002, pp. 160–169.

Komang Oka Saputra received the B.Eng. degree in electrical engineering from Brawijaya University, Malang, Indonesia, in 2004, and the M.Eng. degree in electrical engineering from the University of Indonesia, Depok, Indonesia, in 2006, with a specialization in telecommunication engineering. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan.

He has been a Faculty Member with the Department of Electrical and Computer Engineering, Udayana University, Bali, Indonesia, since 2008. His current research interests include computer network, networking protocol, and security issues of communication systems.

Wei-Chung Teng (M'09) received the D.Eng. degree from the University of Tokyo, Tokyo, Japan, in 2001.

He is currently an Associate Professor with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. His current research interests include human–computer interaction focusing on remote robot manipulation, network communication protocols of time synchronization, and network security issues.

Tsung-Han Chen received the B.S. degree from the Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan, in 2012, and the M.S. degree from the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan.

His current research interests include software engineering and network communication applications.