Entropy-based Clock Skew Measurements for Mobile Devices

Wei-Chung Teng* and Jian-De He†
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
Email: *weichung@csie.ntust.edu.tw, †M10315089@mail.ntust.edu.tw

***Abstract*—Clock skew measurement can be realized in one-way communication by continuously collecting timestamps from the same remote device, and then find the slope of best fit to the offset dataset. However, offsets collected in mobile network may form multi-segmented distribution due to the change of base station or network adapter. This type of distribution usually cause the existing methods fail to provide accurate estimations. This paper introduces a new method to estimate the clock skew of remote network devices. In this method, a three-stage process is adapted to efficiently scan for the skew by which the entropy of adjusted offsets be minimum. Different from the existing approaches, this entropy-based method is not affected by multi-segment offset distributions and is able to deliver precise skew estimation with a short period of time. To compare our approach with existing ones, experiments of both classical and multi-segmented distributions have been conducted. The entropy-based method is the only one whose estimations be bound in 1 parts per million range.**

***Index Terms*—clock skew, entropy, time offset, mobile device**

I. Introduction

All modern digital devices have built-in digital clocks triggered by crystal oscillators. Although a crystal oscillator works in an constant frequency under stable temperature, there exists an error in the manufactured frequency to the ideal one. Therefore, every digital clock ticks slightly faster or slower than the physical time. This error of ticking rate is known as the clock skew, and the ticking rate difference between two digital clocks is called the *relative clock skew*.

The clock skews of network devices play an important role on time synchronization protocol [1], packet transmission delay measurement [2], and time-critical collaborative tasks. The value of skew, when the measurement precision reaches parts per million (ppm), tends to be different for each device. This fact makes clock skew also suitable to serve as a device fingerprint.

Sophisticated approaches have been developed to measure relative clock skews of remote devices from either one-way or two-way communications [2], [3], [4]. However, most of related researches are dedicated to hosts connected to wired networks. Mobile network, on the other hand, has become very popular nowadays with the number of connected mobile devices still increasing year by year. However, communications over mobile network usually introduce higher delay jitters, and the distribution of delay may also vary when every time a mobile device switches its base station. Therefore, it becomes challenging to precisely measure the clock skew of a mobile device in a short period of time like 10 minutes. The contribution of this work is to develop a new method which is suitable for both wired and wireless communications.

A. Relative Clock Skew and Its Measurement

Basically, the relative clock skew of a measurer m to a timestamp sender s can be formulated as follows. For any physical time t, the time reported by the clock of m and by the clock of s are denoted by $C_m(t)$ and $C_s(t)$, respectively. The time offset of m to s at time t is then

$$o_{ms}(t) = C_m(t) - C_s(t).$$

The first derivatives $C^{'}_m(t)$ and $C^{'}_s(t)$, $\forall t \geq 0$ are accordingly the speeds at which both clocks progress at time t. If the environment temperatures of both m and s are stable, it is assumed that the speeds are constant, or $C^{'}_m(t) = C^{'}_m$ and $C^{'}_s(t) = C^{'}_s$. The relative clock skew of m to s, denoted by $s_{ms}(t)$, can then be calculated by $s_{ms}(t) = C^{'}_m(t) - C^{'}_s(t)$. And its constant version is $s_{ms} = C^{'}_m - C^{'}_s$.

To measure s_{ms} over the network, m collects the sending time of each network packet from s, and calculates the time offset of each packet by subtracting the sending time from the receiving time. Let $t_m(i)$ denote the receiving time reported by the measurer's clock, $t_s(i)$ denote the timestamp inside the ith packet, and $\Delta t^s_i = t_s(i) - t_s(1)$ be the elapsed time in the sender's clock when sending the ith packet, then the time offset o_i is

$$\begin{aligned} o_i &= t_m(i) - t_s(i) \\ &\approx o_{ms}(t_s(i)) + d_{op}(i) + d_{tr}(i) \\ &= o_{ms}(t_s(1)) + d_{op}(i) + d_{tr}(i) + \Delta t^s_i \times s_{ms} \end{aligned} \tag{1}$$

where the last two functions at the second line stand for the operation delay and the transmission delay of the ith packet respectively. Both delays are considered bounded positive values, but the last item of the third line indicates that the value of o_i is not bounded and tends to increase or decrease constantly as long as $s_{ms} \neq 0$. As a classic example, Fig. 1 shows a scatter diagram of offsets. The increasing trend of the offsets near the bottom of the distribution reveals that the relative skew of the measurer to the sending device is positive. On the other hand, the outliers near the top of the distribution illustrate how large the jitter is. And a measurement of clock

ISBN: 978-1-4673-9378-2 ©2016 IEEE

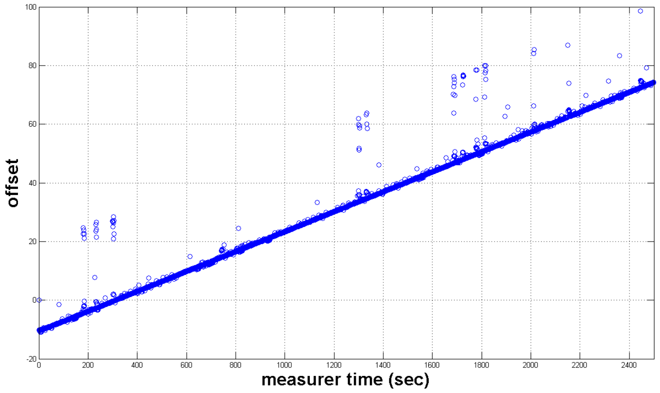


Fig. 1. A scatter diagram of 5000 offsets collected under wired network. The measurer's time is normalized by setting the receiving time of the first packet as 0 second. The values of offsets, in unit of millisecond, are also shifted such that the offset of the first packet is always 0.

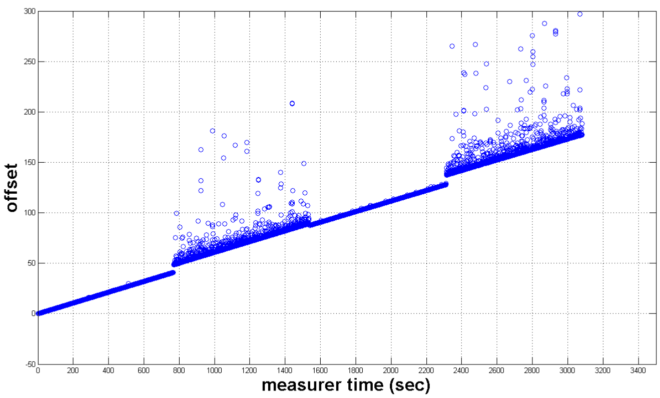


Fig. 2. A scatter diagram of 6000 offsets collected partially under wired network and partially under wireless network.

skew is to estimate the slope of the line formed by the offset majority.

In the case of mobile devices, the scatter diagram of offsets usually differs from the classic one. Fig. 2 illustrates how the pattern of offsets may be affected by the network route and the network adapter. It is obvious that offsets in this figure break into four segments. The first and the third segments are collected when both the sender and the measurer are connected to a campus network via a wired adapter. However, the sender is connected via Wi-Fi during the time interval of the second and the fourth segments. Although the *slope*s of the four segments are very close to each other, it is not an easy task to detect each segment and to estimate the skew from this type of offset distributions.

Another similar case, firstly reported by Huang et al. [5], is the *jump* caused by NTP (Network Time Protocol) synchronization as shown in Fig. 3. Since most network device perform time synchronization once per day, the jump rarely happens and it happens only once during the measuring period. However, the scale of jump might be large enough to cause errors of more than 10 ppm by the existing measuring

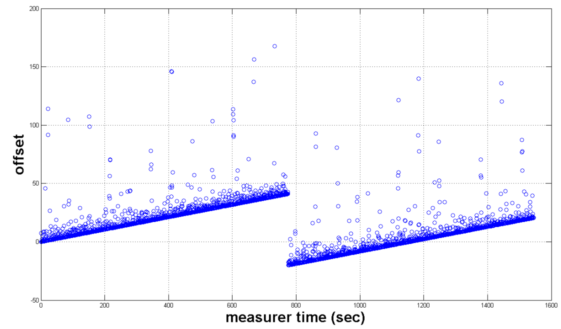


Fig. 3. A scatter diagram of 5000 offsets. The sender conducted a time synchronization during the measurement.

approaches. Therefore, an approach not affected by the segmentation of offset distribution is necessary for mobile device clock skew measurement.

The rest of this paper is organized as follows. Section 2 introduces two state-of-the-art technologies to clock skew measurement. Section 3 explain the idea of entropy-based measurement and the developed process. Section 4 illustrates some experimental results of the proposed approach on above-mentioned cases. The effectiveness of the proposed approach is showed. Finally, Section 5 concludes this paper.

II. Related Work

A. Linear Programming Method

Linear Programming Method (LPM) is proposed by Moon [2] and is commonly used in skew measurements for the last decade. The idea of LPM is to find a line which forms the lower bound to all offsets in the scatter diagram, but this line should pass as more offset as possible. Given a sequence of n 2-tuple $((t_m(1), d_1), (t_m(2), d_2), ..., (t_m(n), d_n))$ where $t_m(i)$ and d_i stand for the receiving time and the offset value of the ith packet respectively, the goal is to find a line $y = \delta x + \phi$ such that

$$\delta t_m(i) + \phi \leq d_i, \forall i = 1 \ldots n$$

and the mean of error

$$\frac{1}{n}\sum_{i=1}^{n}(d_i - (\delta t_m(i) + \phi))$$

is minimized. The derived δ is the relative clock skew. The advantage of LPM is to provide precise estimation with reasonable computation time.

B. Hough Transform Method

Recently, Oka [4] proposed another method which utilizes Hough transform (HT) to detect a *region of majority* (ROM), and use the slope of its lower boundary line as the estimated skew. The origin HT is able to detect a strict line or a curve, and the modified version (HTM) is able to detect a bold line of any given width ω. The origin HT usually scans the whole [-180, 180) degree range in a 1 or 0.5 degree resolution, but

ISBN: 978-1-4673-9378-2 ©2016 IEEE

HTM pursues for 0.1 ppm resolution inside the range of [-7.5e-5, 7.5e-5] degree, so the authors developed a three-stage process to efficiently determine the best region, or the region with the most votes. Since the ROM is determined by the offset majority, HTM is considered robust to the cases that lower outliers happen due to high delay jitter. LPM, on the other hand, needs a longer period of time to have the estimated skew converge to 1 ppm precision.

III. The Entropy-based Approach

A. Entropy of Offsets

To calculate the entropy of a given offset set of size n, the times of each each value is counted. Assume that there are m different values in the offset set, and the count of each value is denoted by $n_i, \forall i = 1 \dots m$. Then

$$n = \sum_{i=1}^{m} n_i$$

Now let $p_i = \frac{n_i}{n}$, then the entropy of the offset set can be calculated as

$$H = -\sum_{i=1}^{m} p_i \log p_i$$

Use Fig. 1 as an example, the lower half of Fig. 4a shows a bar chart of the counts of offset inside a range of -20 to 100 millisecond. As revealed by (1), an offset set without skew should be bounded to a small range, or the delay jitter, no matter how large its size n grows. Assume that the estimated relative clock skew s'_{ms} is very close to s_{ms}, then the skew-free offset o'_i can be approximated by

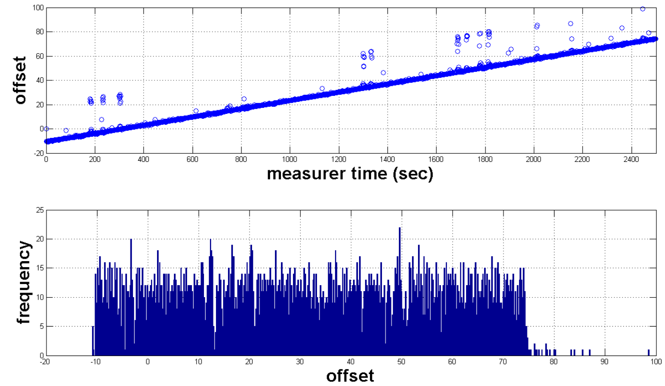
$$\begin{aligned} o'_i &= o_i - \Delta t_i^s \times s'_{ms} \\ &= o_{ms}(t_s(1)) + d_{op}(i) + d_{tr}(i) + \Delta t_i^s \times (s_{ms} - s'_{ms}) \\ &\approx o_{ms}(t_s(1)) + d_{op}(i) + d_{tr}(i) \end{aligned}$$

The expected distribution of skew-free offsets in both the scatter diagram and in the bar chart is illustrated by Fig. 4b. It is obvious that the counts of offset value will gather inside a smaller region as $s_{ms} - s'_{ms}$ be closer to 0. It is also known that a random variable with the standard deviation σ of its probability distribution smaller, it contributes lower entropy [6]. Therefore, the entropy of an *adjusted* offset set is able to serve as a mean for clock skew detection.

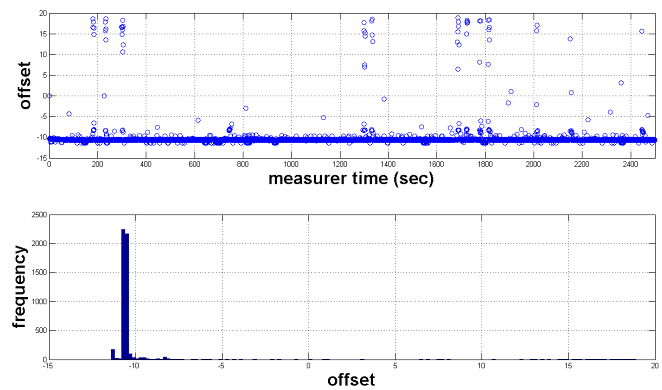
B. Proposed Method

The entropy-based method adapts the three-stage process of HTM [4] to scan an angle range for the best slope which derives the smallest entropy. The algorithm is shown below. The required input O and n stands for the offset set and the number of offsets. Variable O is assumed to be an array with index $1 \dots n$. For the ith offset, $O[i].v$ and $O[i].t$ store the value of offset and the elapsed time Δt_i^s respectively. Another array variable C is used to store the counts of offset. Therefore, $C[v]$ should record the count of offset for any value v.

Given O, n, the angle lower bound θ_{min}, the angle upper bound θ_{max}, and the resolution p, the function MinEntropy() at line 9 will return the angle by which the entropy of offset values becomes minimum. The first call to MinEntropy() at line 2 gives the estimated skew in 10 ppm resolution. The second call will enhance the estimation to 1 ppm, and the third call to 0.1 ppm resolution respectively. Using the three-stage technique, a precise estimation can be determined by less than 200 runs of MinEntropy().



(a) Original offset distribution



(b) Target offset distribution

Fig. 4. An classic offset distribution in wired network.

IV. Experimental Results

To verify the effectiveness and robustness of the proposed method, three types of distributions are measured in our experiments. The results of the proposed method are also compared with existing approaches LPM and HTM. The measurer used in all experiments is a desktop computer of 64 bits Ubuntu 14.04 operating system.

A. Classical Distribution Case

The first experiment is to measure the clock skew of a MacBook Air notebook with OS X 10.10.5 installed. Both sender and measurer connected to a LAN. The sender was programmed to send a packet by UDP once per 0.5 second, with each packet containing only a timestamp of the sending time.

The scatter diagram of the collected offset set is the one shown in Fig. 1 and Fig. 4a. And the results of three methods is summarized in Table I. Although the proposed method is not as stable as HTM does when the measuring period is less

Algorithm 1 Three-stage entropy-based clock skew measurement

Require: O, n
$p \leftarrow 10^{-5}$
θ = MinEntropy(O, $(\pi/2) - 750*10^{-6}$, $(\pi/2) + 750*10^{-6}$, p, n)
$p \leftarrow p/10$
θ = MinEntropy(O, $\theta - 5*p$, $\theta + 5*p$, p, n)
$p \leftarrow p/10$
θ = MinEntropy(O, $\theta - 5*p$, $\theta + 5*p$, p, n)
return θ

function MINENTROPY(O, θ_{min}, θ_{max}, p, n)
  $H_{min} \leftarrow 0$
  $\theta_{skew} \leftarrow 0$
  $\theta \leftarrow \theta_{min}$
  while $\theta \leq \theta_{max}$ **do**
    initialize C
    for $i \leftarrow 1, n$ **do**
      $v' \leftarrow O[i].v - \theta * O[i].t$
      $C[v'] \leftarrow C[v'] + 1$
    end for
    $H \leftarrow 0$
    for all $C[j]$ **do**
      $H \leftarrow H - (C[j]/n) * \log(C[j]/n)$
    end for
    if $H_{min} \neq 0$ **and** $H < H_{min}$ **then**
      $H_{min} \leftarrow H$
      $\theta_{skew} \leftarrow \theta$
    end if
    $\theta \leftarrow \theta + p$
  end while
  return θ_{skew}
end function

TABLE I
SKEWS OF A CLASSIC OFFSET-SET

Offset	LPM (ppm)	HTM (ppm)	proposed method (ppm)
1-1000	34.05	33.8	33.8
1001-2000	34.07	33.9	34.1
2001-3000	33.96	33.8	33.9
3001-4000	33.83	33.8	33.9
4001-5000	34.02	33.9	34.0
Max-Min	0.24	0.1	0.4
1-5000	33.97	33.9	34.0

than 10 minutes, it provides precise estimation as the two other methods do (the 1-5000 row).

B. Segmented Distribution Cases

There are two cases in this subsection: the switching-adapter case as shown in Fig. 2, and the NTP time-sync case as shown in Fig. 3. The sender used in both experiments was another desktop computer of Windows 7 OS. The sending period was set to 0.5 second, and 6000 packets were collected in both experiments.

TABLE II
SKEWS OF MULTI-SEGMENT OFFSET-SETS

Case	LPM (ppm)	HTM (ppm)	proposed method (ppm)
Wired LAN	53.07	53.2	53.1
Switching Adapter	56.37	56.37	52.49
Time Sync	-25.95	-1	52.89

Table II shows the results of the three methods. It is obvious that the precision of estimations in multi-segment cases become lower than the classic case. However, only the proposed method was able to contribute estimated clock skews with less than 1 ppm errors.

V. CONCLUSION

This paper introduced an innovative method to estimate the relative clock skew of remote device over network. This method was shown to be robust especially to cases when the time offsets break into more than one segment in the scatter diagram. Since this method uses entropy to determine the best fit to clock skew, there is no need to detect for segmentation or even the boundary of each segment. Consider that mobile devices in motion may change the route of network connection for multiple times during a session, causing the collected time offsets to be of multi-segment distribution, this method is especially suitable to measure clock skew of mobile devices.

ACKNOWLEDGMENT

This work was supported under the Ministry of Science and Technology Grants MOST 104-2923-E-011-005-MY3 and 104-2221-E-011-070.

REFERENCES

[1] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: a survey,” *Network, IEEE*, vol. 18, no. 4, pp. 45–50, 2004.
[2] S. B. Moon, P. Skelly, and D. Towsley, “Estimation and removal of clock skew from network delay measurements,” in *INFOCOM’99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 1999, pp. 227–234.
[3] M. Aoki, E. Oki, and R. Rojas-Cessa, “Measurement scheme for one-way delay variation with detection and removal of clock skew,” *ETRI journal*, vol. 32, no. 6, pp. 854–862, 2010.
[4] K. Oka Saputra, W.-C. Teng, and T.-H. Chen, “Hough transform-based clock skew measurement over network,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 64, no. 12, pp. 3209–3216, 2015.
[5] 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 *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*. IEEE, 2012, pp. 526–533.
[6] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

ISBN: 978-1-4673-9378-2 ©2016 IEEE