Measuring clock skews of remote devices via wireless communications

Wei-Chung Teng
Dept. of Computer Science & Information Eng.
National Taiwan University of Science and Technology
(a.k.a. Taiwan Tech)
What is clock skew?

- Almost all digital device has a clock (crystal oscillator), and quartz crystal in every device works in slightly different frequency.

- Thus the speeds (sec/sec) of each two clocks are slightly different
  - we call the difference *relative clock skew*
Why should we care about clock skew?

- Clock skews of the same clock remain the same in normal temperature.
- Past researches (e.g. Kohno, 2005) show that every clock skew measured remotely differs with others at μs precision.
- Clock skew is suitable to serve as the physical identity of a digital device.
How to measure a (relative) clock skew?

- Let $C_x(t)$ be the time reported by the clock of device $x$.
- **Offset**: The difference between the time reported by $C_c$ and $C_s$.
- **Frequency**: The rate at which the clock ticks. The frequency of $C_c$ at time $t$ is $C'_c(t)$.
- **Skew ($\delta$)**: The difference in the frequencies of two clocks, e.g., the skew of $C_c$ relative to $C_s$ at time $t$ is $\delta(t) = C'_c(t) - C'_s(t)$. 

![Diagram of host C and host S with time ticks and skew calculation]

- $t_1^c$ to $t_2^c$ on host C
- $t_1^s$ to $t_2^s$ on host S
- Offset calculation $o_1 = t_1^s - t_1^c$ and $o_2 = t_2^s - t_2^c$
How to measure a (relative) clock skew? cont.

• Since there exists communication delay, we are unable to know the exact offset, but (offset + delay)

• but the delay is irrelevant to measuring the clock skew if the delay is a constant

• We have $\delta(t_2) = \frac{o_2 - o_1}{t_2^s - t_1^s}$
How to measure a (relative) clock skew? cont.

• Since the communication delay is never a constant (there exists jitter), we can not use just two timestamps, we need more samples.
How to measure a (relative) clock skew? cont.

• We can use linear regression to find out the slope which best fits the *trend* of sampled offset
  • might be affected severely by outliers
• We can use linear programming instead
  • not an efficient method if the jitter is large (we need to sample more)
• In a classic sample, most samples are close to (possibly) the minimum delay, so we can pick up points of least delay and run LP with these points.
Question: how to detect a faked clock skew?

- Timestamps are just a series of increasing numbers, sender may alter the speed it increase easily.
- We have found that even for one hop transmission, sender may adjust its skew as it likes.
- However, if we ask the sender to slightly change its sending period from time to time, the fluctuation scale of a faked skew would be more than 10 times of the true skew.
Example: Flooding Time Synchronization Protocol

## Question: what is the possible range of clock skew?

<table>
<thead>
<tr>
<th>Research Title</th>
<th>Min (ppm)</th>
<th>Max (ppm)</th>
<th>Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lanze, F.; Panchenko, A.; Braatz, B.; Zinnen, “Clock skew based remote device fingerprinting demystified,” 2012 IEEE Global Communications Conference (GLOBECOM)</td>
<td>-30.0</td>
<td>30.0</td>
<td>200 APs</td>
</tr>
<tr>
<td>S. Sharma; A. Hussain; H. Saran, “Experience with heterogenous clock-skew based device fingerprinting,” the 2012 ACM Workshop on Learning from Authoritative Security Experiment Results (LASER ‘12)</td>
<td>-150</td>
<td>750</td>
<td>52 devices</td>
</tr>
</tbody>
</table>
An example application: client device identification for cloud services

- Personal devices of private use
- Cloud servers

- Two-factor authentication
- Device identity
- Account & password

- iCloud
- Dropbox
- aero fs
- CrashPlan
- Evernote
- Gmail
The estimated skews for the same device under different environments

- The estimated skews vary from -21.08 ppm to -23.71 ppm. However, skews of the same network type differ no more than 1.31 ppm.

- Notice that skew of a virtual machine might change every time it reboots.

<table>
<thead>
<tr>
<th>Network type</th>
<th>Skew estimation</th>
<th>Packets</th>
<th>IP amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAN</td>
<td>-21.91 ppm</td>
<td>1001</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-23.24 ppm</td>
<td>207</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-22.74 ppm</td>
<td>13322</td>
<td>1</td>
</tr>
<tr>
<td>ADSL</td>
<td>-21.48 ppm</td>
<td>5837</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-21.08 ppm</td>
<td>1400</td>
<td>1</td>
</tr>
<tr>
<td>3G</td>
<td>-23.24 ppm</td>
<td>951</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-23.71 ppm</td>
<td>1027</td>
<td>1</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>-21.79 ppm</td>
<td>9810</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-23.06 ppm</td>
<td>1470</td>
<td>1</td>
</tr>
<tr>
<td>Tor</td>
<td>-22.53 ppm</td>
<td>15007</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>-23.22 ppm</td>
<td>12922</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>-22.88 ppm</td>
<td>24120</td>
<td>108</td>
</tr>
<tr>
<td>VM</td>
<td>-113.19 ppm</td>
<td>868</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-114.22 ppm</td>
<td>1001</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-6.40 ppm</td>
<td>1001</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-6.83 ppm</td>
<td>890</td>
<td>1</td>
</tr>
</tbody>
</table>
Some new issues on clock skew measurement for WiFi/mobile communications

1. Jump points
2. (varying) Minimum sampling time period
3. Outliners below the crowd
Jump Points

- Caused by a sudden change of offset or delay
- Happen when a device run SNTP/NTP with time servers
- Happen when a mobile device changes base station during a mobile communication sessions
- Happen when a mobile device switches from WiFi to 4G or vice versa
A jump point example

- A jump point of offset occurs if the client is performing time synchronization with a time server or roaming between different network providers.
Another type of jump point
Minimum sampling time period

server located in AWS EC2
Minimum sampling time period cont.

0.5 ts/s, 1 client

0.5 ts/s, 2 clients
The raining phenomenon

• Always the same slope per receiver (e.g. ~ -1600 ppm)
• Multiple (2~4) lines at the same moment
• Possibly caused by the queuing scheme of network adapter drivers and OSes
Outliners below the crowd

- Only observed in wireless communication till now
Conclusions

• Continuous check for jump points and raining are necessary.

• Adaptive algorithm necessary to adjust the sending period of timestamps

• Hough line transform is effective to eliminate the error caused by “lower” outliers.

• Finally, if the sample is clean, we need no more than 2,000 offset values to reach ppm level precision.
תודה