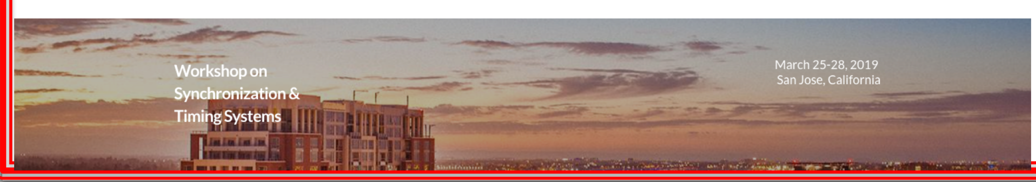
A Nonlinear Model for Time Synchronization

Frank Wang, frwang@equinix.com; Danjue Li, danli@equinix.com

1188 E. Arques, Sunnyvale, CA 94085, USA

EQUINIX



Abstract

The current algorithms of software solution are based on a linear model. For example, Precision Time Protocol (PTP), which requires frequent synchronization in order to handle the effects of clock frequency drift. This paper explores a nonlinear approach to synchronize clock time. The nonlinear approach can model the frequency shift in a better way. Therefore, the required time interval to synchronize clocks can be longer. Meanwhile, it also offers better performance and relaxes the synchronization process.

Clock Frequency

The frequency of a clock is its intrinsic property. The clock environment has little impact on frequency except in extreme cases. The server clock's ***frequency shifts slowly*** and approximately linearly related to the aging and temperature effects within a time window of a few hours.

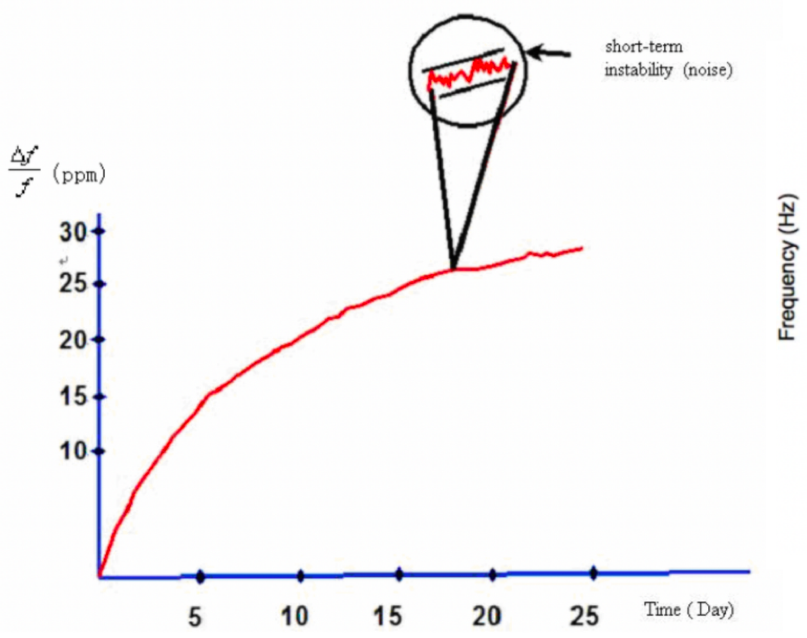


Fig 1. Aging of crystal resonator [3]

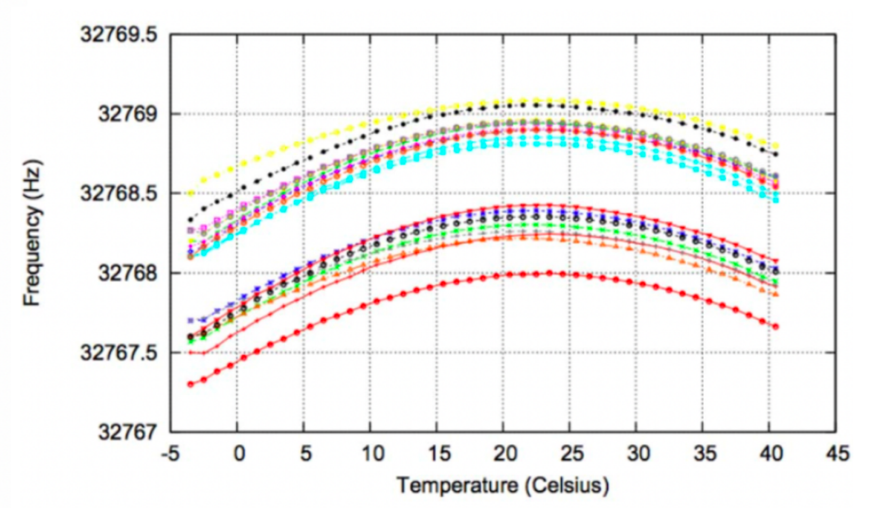


Fig 2. Measured temperature effect on the frequency of several crystal oscillators (nominal frequency f0=32768.5Hz) [5]

Clock Software Limitations

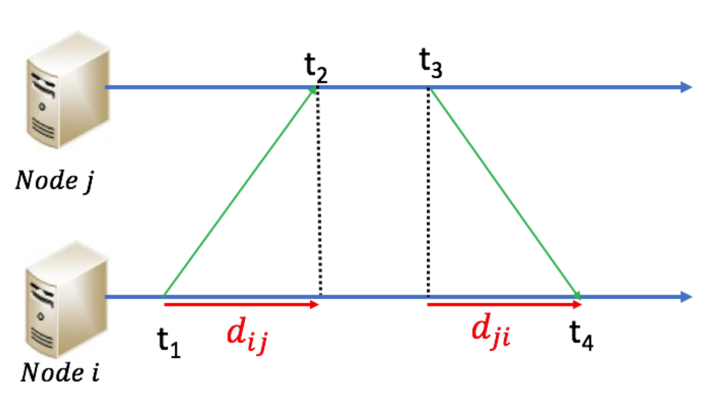


Fig 3. Two-way communication schema

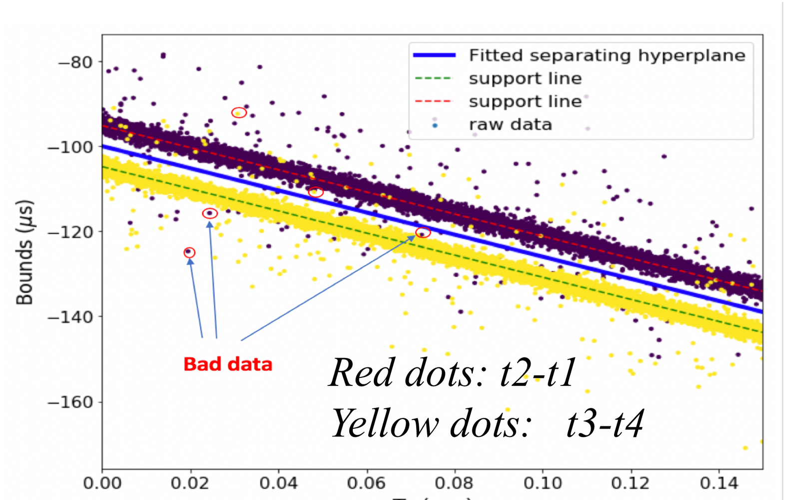


Fig 4. Linear SVM for skewness and offset

The software-based algorithm suffers from ***large errors and jitters*** from the time measurement (timestamps).

Learning methods with noisy data:

- Large data set
- Regression with *inliers*
- Regression with *outlier*
- *Bad data (*points between the two bounds*)* removal
- ***Customized algorithm***

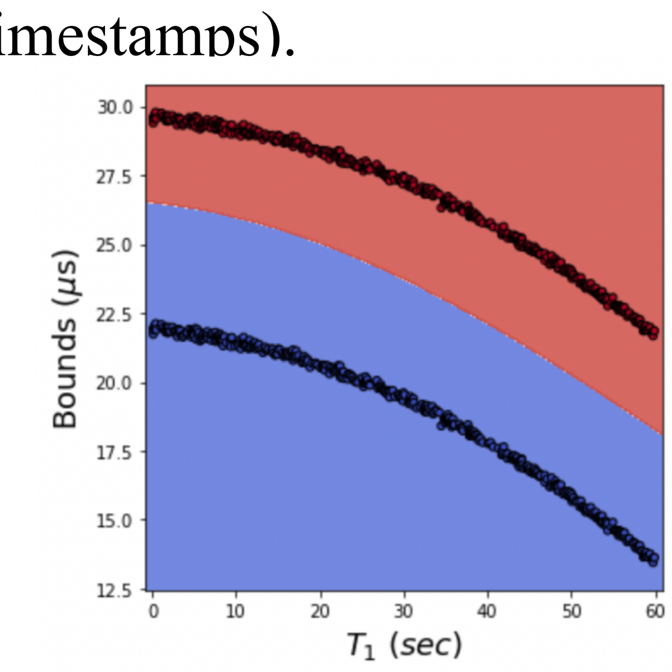


Fig. 5: Learning of nonlinear model with time stamps

Clock Model

- In the linear model: $s(t) = constant$
- In the nonlinear model: $s(t) = \alpha t + \beta$, or higher order

Bottle Neck of linear timing (PTP here)

- The large variance in frequency contributes to the large variance in offset.

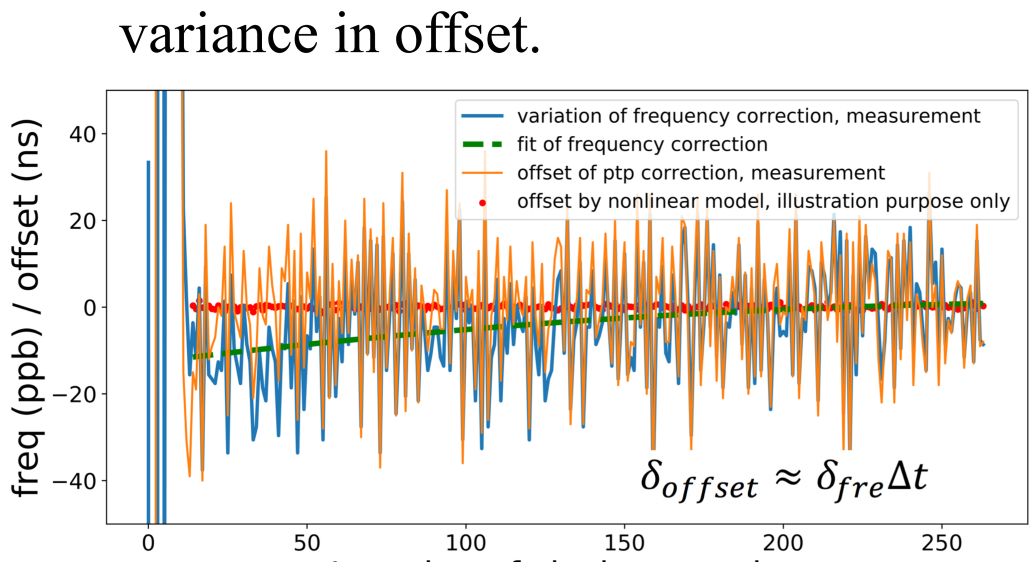
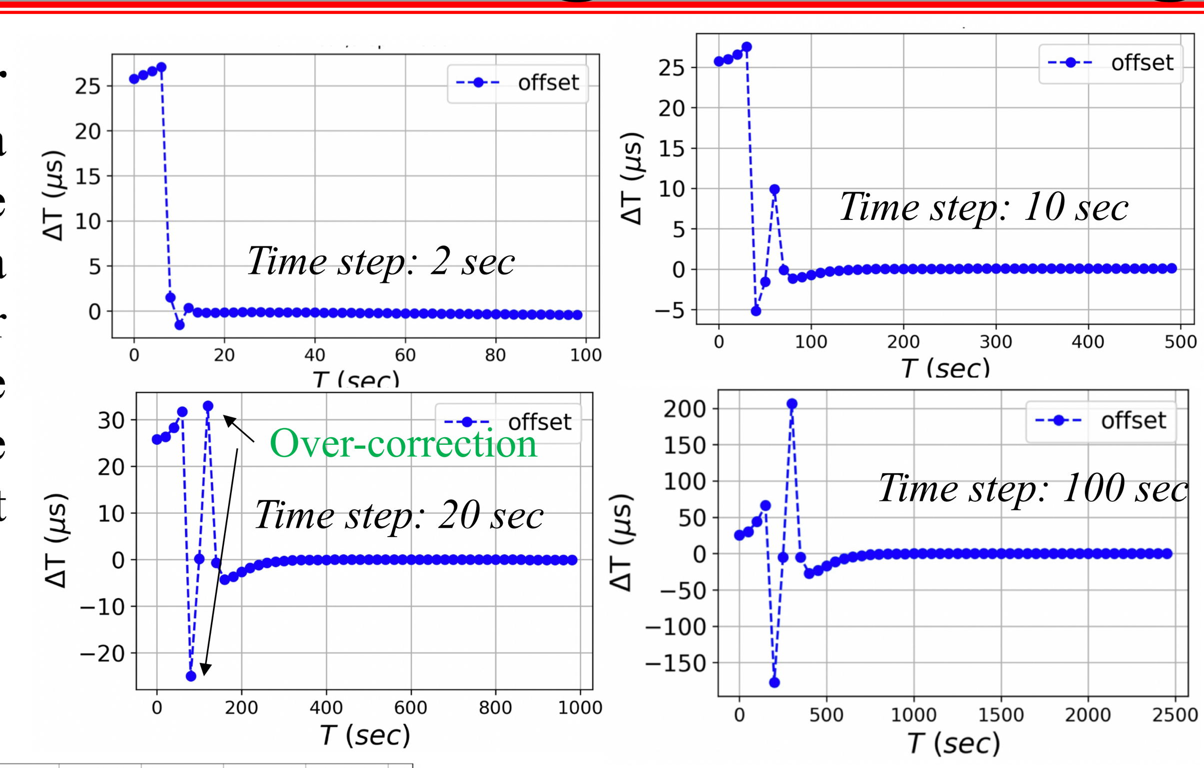


Fig. 6: **Strong correlation** (0.96) between the measured frequency correction (blue) and clock offset (yellow) in a ptp4l test. The constant part of the frequency correction is removed

Numerical Convergence Testing

Fig. 7: **Nonlinear timing** with a constant time step; there is a large over correction at large time step (the algorithm is not optimized)



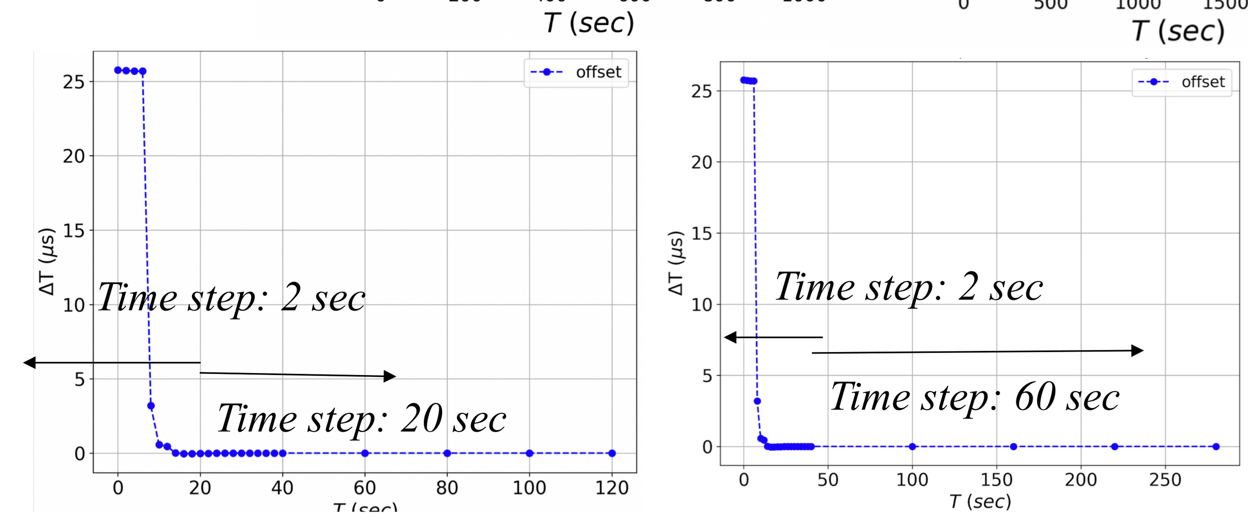


Fig. 8: **Hybrid time steps** to mitigate the over correction and reduce the convergence time at large time step

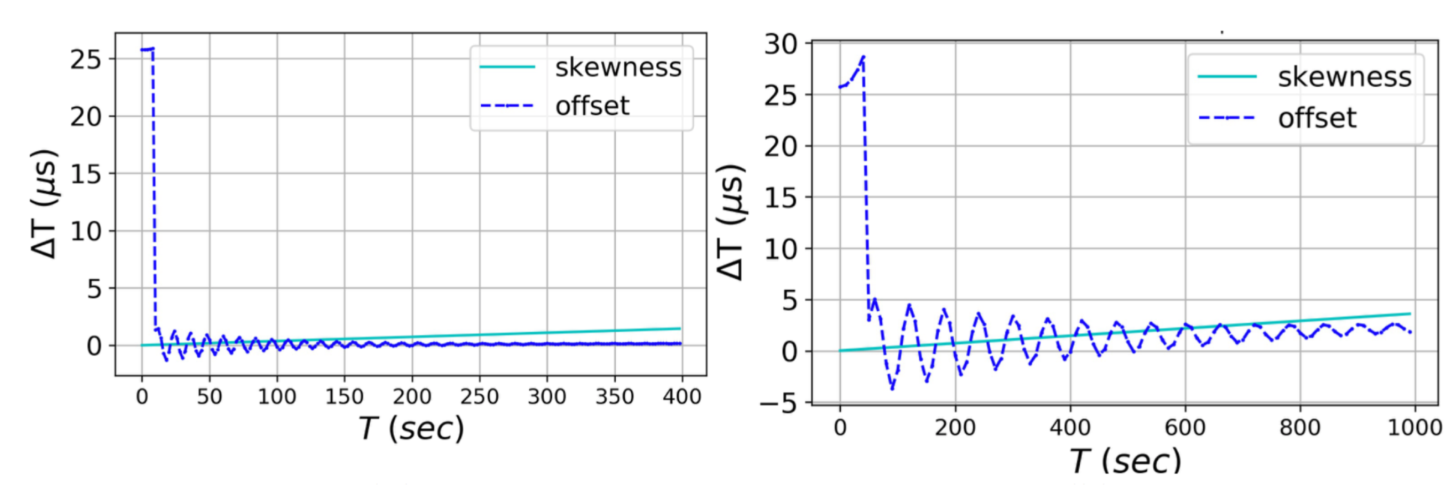


Fig. 9: Linear model for a time step of 2 sec (left) and 10 sec (right). The 2 sec case converges well; however the 10 sec step has large error accumulated

Summary

	Linear model	Nonlinear model
Clock model	Constant frequency	Frequency varies with time
Performance	Sensitive to time steps and frequency drift speed	Outperforms linear model in general
Time interval	• Small (~ sec) • frequent synchronization • Heavy communication traffic • Long computation time	• Large (> 20s Sec) • In-frequent synchronization • Light communication traffic • Short computation time
Noise effect	Sensitive to noise	Immune to noise
Fast/slow frequency drifting clocks	Sensitive to clock frequency drift speed	Insensitive to clock frequency drift speed

References and Acknowledgements

We would like to thank Equinix timing service team for fruitful discussions.

[1] Corinna Cortes and Vladimir Vapnik (1995). "*Support- vector networks*". Machine Learning, 20 (3):273– 297. doi:10.1007/BF00994018.
[2] Yilong Geng, et.al., *Exploiting a Natural Network Effect for Scalable*, Fine-grained Clock Synchronization, 15th USENIX Symposium on Networked Systems Design and Implementation, NSDI'18
[3] John R. Vig, "*Quartz Crystal Resonators and Oscillators for Frequency Control and Timing Applications - A Tutorial*", 2004 IEEE International Frequency Control Symposium Tutorials, May 2004
[4]https://en.wikipedia.org/wiki/Crystal_oscillator#Temperatu re_effects
[5] Martin Lévesque and David Tipper, "*A Survey of Clock Synchronization Over Packet-Switched Networks*", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 18, NO. 4, FOURTH QUARTER 2016