Poster Abstract: Robust Time Synchronization in Wireless Sensor Networks Using Real Time Clock

Hessam Mohammadmoradi
Omprakash Gnawali
Computer Science Department
University of Houston
{hmoradi,gnawali}@cs.uh.edu

Nir Rattner
Andreas Terzis*
Computer Science Department
Johns Hopkins University
{nrattne1,terzis}@jhu.edu

Alex Szalay
Physics and Astronomy Department
Johns Hopkins University
szalay@jhu.edu

Abstract

Time synchronization is an essential service in many sensor network applications. Harsh environment which causes nodes to fail, go offline, or reboot can challenge many time synchronization protocols. In this work, we first characterize this challenge and use a real time clock in one of the nodes in the network to improve robustness of time synchronization. Our experiments show that our approach improves the robustness of state-of-the-art offline time synchronization protocols.

1 Introduction

In many WSN applications, we analyze sensor data over space and time. Synchronized and accurate timing is essential part of those applications. Many inexpensive and low-power wireless sensor node platforms are not equipped with GPS time. They rely on time synchronization protocol so all the sensor nodes in the deployment can use the same time reference to timestamp their sensor data.

Two classes of time synchronization (or reconstruction) methods have found wide use in wireless sensor networks. Online time synchronization protocols [4] designate a node as a reference time for the network and propagate the reference time to the entire network to keep the clocks on all the nodes synchronized to the clock of the reference node. The main advantage of online methods is their accuracy and the main disadvantage is their energy consumption overhead.

Offline methods [6, 3], on the other hand, use time reconstruction at the base station or the server to infer the time at which sensor samples were taken. The sensor data is sampled using the local clocks. Additional information that can translate the local clocks to a reference clock is used at the base station to compute the global time after the data is collected. This method is typically more energy efficient than online algorithm but may incur larger errors than online algorithms.

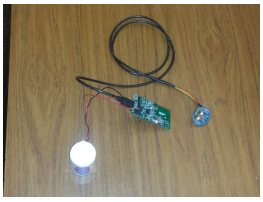

Figure 1. A Bacon sensor node connected to Chronodot

In our work, we develop techniques to improve the robustness of offline time synchronization protocols by introducing a node with real-time clock in the network. We use Phoenix [2] as a case study of how our technique can be incorporated into an offline time synchronization schema to improve its robustness. Phoenix was specifically designed to be robust to frequent node reboots and base station going offline and is known to achieve error rate of 6ppm for 99% of measured data. However, Phoenix, and many similar algorithms, consider the base station as a reliable source of accurate time and ignore base station failures.

Our approach introduces a node with a real-time clock to provide reference time to the network. This node is different from a base station. Separation of functionalities between a base station and time reference can simplify the software architecture and robustness of the timing service. The time advertised by the reference mote propagated multi-hop across the network, with appropriate adjustments on each hop. The sensor nodes then log these global reference times which provide time anchors during time reconstruction after data is sent to the base station.

Our main contribution in this work is the design of an architecture that can tolerate errors caused by base station absence, reboots, and sensor node reboots. Our second contribution is a testbed-based evaluation of our approach.

2 Time Reconstruction

The time reconstruction algorithm relies on time anchors logged by the motes. These anchors are time broadcasts received by a mote. Each node periodically broadcasts its local time to its direct neighbors. The nodes log such received time broadcasts along with their local time but attempt no calibration or synchronization online. One of these nodes is a reference node with a hardware real-time clock.

*Currently at Google

SenSys'14, November 3–5, 2014, Memphis, TN, USA.
ACM 978-1-4503-3143-2/14/11.
http://dx.doi.org/10.1145/2668332.2668365

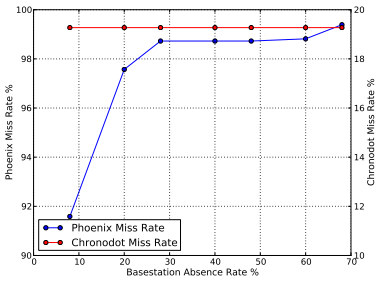

Figure 2. Unsynchronized timestamps rate over base station absence rate

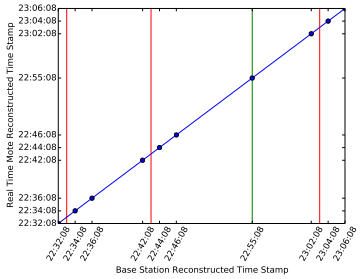

Figure 3. Time reconstruction robustness against Leaf (green) and Reference (Red) mote reboots

Base station periodically (not necessarily with constant intervals) queries entire network and downloads each node's stored records. Using node's time stamps relations, we can reconstruct time stamps by finding a path from that local time to global time. We can use a simple linear model ($RTS = \alpha * LTS + \beta$) to describe the relation between local (LTS) and remote (RTS) time stamps [5].

Node reboots are quite common among sensor nodes and linear relationship among time stamps is only valid in periods which both LTS and RTS monotonically increasing. Once we detect a reboot in the collected log, we need to recalculate α and β values. First, we need to find segments have direct relation to global segments. Second, we need to find α and β values for those segments which do not have direct link to global segments using segments we discovered at first phase.

3 Evaluation

In this section, we describe an implementation of our approach and experimental results from a small deployment.

3.1 Implementation

We use TinyOS and Bacon motes for our implementation. The Bacon mote uses TI's CC430 SoC, which combines a 900MHz radio core (almost equivalent to a CC1101) and an MSP430MCU. Rather than integrate a real-time clock onto the mote boad, we use Chronodot [1] hardware module as a source of accurate real-time clock. Chronodot is a self-contained real-time clock module with its own power supply. Thus, the clock can continue to work even when the mote fails or runs out of power. According to its datasheet, a single CR1632 can power a Chronodot for up to 8 years. We connect Chronodot to Bacon mote using a I2C interface. We wrote a I2C driver to obtain time from Chronodot. Figure 1, shows a reference node.

3.2 Experiments and Results

We deployed 27 Bacon nodes in our lab with 1 minute sampling interval and 2 minutes local time broadcasting. Base station collects data every 10 minutes. We ran the experiment under two different scenarios for 8 hours. In the first scenario, we used base station as global time reference. This setting is similar to Phoenix, or other systems, that use the data collection base station to also provide the reference time. In the second scenario, we used the reference mote (with Chronodot) to provide the global time. During the experiments, we introduced Leaf, base station, and reference node reboots.

Figure 2 shows the fraction of timestamps the time reconstruction algorithm was not able to reconstruct in the two scenarios. We find that in the second scenario, with the reference mote, there are fewer unsynchronized time than in the first scenario even though the base station continuously broadcasts its time to the network. Second, we observe that when there are base station outages, our approach can reconstruct the time more accurately than Phoenix.

Next, we try to understand if our approach is robust against reboots of the reference mote. During our experiment, we manually rebooted Chronodot equipped node and reconstructed the timestamps uploaded from one of the Leaf motes. Figure 3 shows the reconstructed times using our approach. We observe that the reconstructed timestamps follow the linear model despite the reboots of the Leaf mote and the reference mote.

4 Conclusions

In this project, we propose a robust offline time synchronization approach for WSN applications. Our system uses real time clock to reconstruct time stamps and we showed our approach is more resistant to failures compare to state of the art time synchronization algorithms. Our technique leverages the existence of extremely stable and reliable real time clock hardware modules to provide robustness and reliability in WSN time synchronization and reconstruction.

5 Acknowledgments

This work was partially supported by the National Science Foundation under grant no. IIS-1111507.

6 References

[1] Chronodot clock,http://www.adafruit.com/products/255.

[2] J. Gupchup, D. Carlson, R. Musloiu-E., A. Szalay, and A. Terzis. Phoenix: An epidemic approach to time reconstruction. In *WSNs*, volume 5970 of *LNCS*. 2010.

[3] B. R. Hamilton, X. Ma, Q. Zhao, and J. Xu. Aces: Adaptive clock estimation and synchronization using kalman filtering. MobiCom '08.

[4] F. Ren, C. Lin, and F. Liu. Self-correcting time synchronization using reference broadcast in wireless sensor network. *Wireless Communications, IEEE*, 15(4):79–85, Aug 2008.

[5] J. Sallai, B. Kus, . Ldeczi, and P. Dutta. On the scalability of routing integrated time synchronization. In *WSNs*, volume 3868 of *LNCS*. 2006.

[6] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. OSDI '06.