

Natural Timestamping Using Powerline Electromagnetic Radiation

Yang Li
Advanced Digital Sciences Center,
Illinois at Singapore
1 Fusionopolis Way, #08-10 Connexis
North Tower
Singapore 138632
yang.li@adsc.com.sg

Rui Tan*
School of Computer Science and
Engineering, Nanyang Technological
University
50 Nanyang Avenue
Singapore 639798
tanrui@ntu.edu.sg

David K. Y. Yau
Information Systems Technology and
Design, Singapore University of
Technology and Design
8 Somapah Road
Singapore 487372
david_yau@sutd.edu.sg

ABSTRACT

The continuous fluctuation of electric network frequency (ENF) presents a fingerprint indicative of time, which we call *natural timestamp*. This paper studies the time accuracy of these natural timestamps obtained from powerline electromagnetic radiation (EMR), which is mainly excited by powerline voltage oscillations at the rate of the ENF. However, since the EMR signal is often weak and noisy, extracting the ENF is challenging, especially on resource-limited sensor platforms. We design an efficient EMR conditioning algorithm and evaluate the time accuracy of EMR natural timestamps on two representative classes of IoT platforms – a high-end single-board computer with a customized EMR antenna and a low-end mote with a normal conductor wire acting as EMR antenna. Extensive measurements at five sites in a city, which are away from each other for up to 24 km, show that the high-end and low-end nodes achieve median time errors of about 50 ms and 150 ms, respectively. To demonstrate the use of the EMR natural timestamps, we discuss two applications, namely time recovery and run-time clock verification.

CCS CONCEPTS

•Computer systems organization →Sensor networks; Dependable and fault-tolerant systems and networks;

KEYWORDS

Timestamping, electromagnetic radiation, sensor networks

ACM Reference format:

Yang Li, Rui Tan, and David K. Y. Yau. 2017. Natural Timestamping Using Powerline Electromagnetic Radiation. In *Proceedings of The 16th ACM/IEEE International Conference on Information Processing in Sensor Networks, Pittsburgh, PA USA, April 2017 (IPSN 2017)*, 12 pages. DOI: <http://dx.doi.org/10.1145/3055031.3055075>

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IPSN 2017, Pittsburgh, PA USA

© 2017 ACM. 978-1-4503-4890-4/17/04...\$15.00
DOI: <http://dx.doi.org/10.1145/3055031.3055075>

1 INTRODUCTION

In the era of Internet of Things (IoT), we have witnessed the proliferation of networked sensors and smart devices embedded in civil infrastructures. It is estimated that, by 2019, more than 100 million wireless sensors will be installed for smart buildings [17]. Modern industries are also embracing the vision of IoT for their production lines [11]. For all these systems, having a *common notion of time* (CNoT) among nodes is a fundamental and critical requirement. Accurate timestamps are crucial for interpreting data and associating data from different sensors; synchronized clocks enable punctual and well coordinated operations among the nodes. On the other hand, wrong timestamps and clock desynchronization may lead to system chaos and failures. Thus, resilient CNoT against faults such as hardware faults and network outage is important for diverse wireless sensor network (WSN) and IoT systems.

Atomic clocks, GPS, and clock synchronization and calibration protocols represent principal means to achieve the CNoT. For massive deployments, chip-scale atomic clocks are still too expensive (e.g., \$1,500 per unit [5]). Although GPS receivers can provide global time with μ s accuracy, they generally do not work in indoor environments. Existing clock synchronization and calibration approaches can be classified broadly into two categories. The first category consists of clock synchronization protocols (e.g., NTP [4], PTP [3], RBS [8], TPSN [12], and FTSP [21]) that exchange network or radio messages among nodes to estimate and remove their clock offsets. As these approaches generally involve measuring the delays in transmitting synchronization packets, network outage and dynamic network conditions may degrade their performance significantly [22]. The second category uses time broadcasts from timekeeping radios [7] and external periodic signals (e.g., periodic data blocks in FM radios [18], Wi-Fi beacons [16], and powerline electromagnetic radiation [23]) for clock synchronization and calibration. To capture and/or decode the external signals, these approaches need sophisticated hardware peripherals, which hinders their adoption due to the increased node cost and complexity. Moreover, the wireless signals used (e.g., FM/AM and Wi-Fi) may be blocked and interfered with.

In this paper, we study *natural timestamping*, a low-cost and novel approach to resilient CNoT. In general, a natural timestamp is a finite-length sampled signal from the environment that yields a unique form during any given sampling time. In other words, the signal form “encodes” when it is sampled. A prior work [25] shows that the electric network frequency (ENF) of an alternating current (ac) power grid contains natural timestamps that encode

time information at sub-ms accuracy. Specifically, as the ENF fluctuates around its nominal value (50 Hz or 60 Hz) [26] and the fluctuation is almost identical across the whole power grid, an ENF trace collected by a node, say *A*, can be matched against a longer ENF trace collected by another node, say *B*. The best matching position indicates a reliable common time when *A* and *B* obtained their respective ENF samples. Hence, CNoT is established between *A* and *B*.

As a power grid pervades almost all civil infrastructures in an urban geographic area, ENF natural timestamping is promising for establishing reliable time for WSN and IoT in these environments. Moreover, based on the natural timestamps, various time-critical services and protocols with much improved resilience can be developed. For instance, the ENF natural timestamps recorded by a node during an extended network outage can be decoded afterwards to recover time information of any critical sensor data and system logs. ENF natural timestamps can also be used for clock synchronization, as shown in [25]. Specifically, by exchanging ENF natural timestamps that are associated with *A*'s and *B*'s clocks, the offset between the two clocks can be identified through natural timestamp matching [25]. Since this approach does not depend on measuring packet transmission delays, it is immune to dynamic delays due to varying network conditions.

However, prior work [25] collects ENF traces from wall power outlets directly, which limits its applicability to nodes that are plugged in to the power grid only. To push the state of the art, in this paper, we systematically investigate the existence and time accuracy of the natural timestamps in wireless electromagnetic radiation (EMR) emitted from ac powerlines. The basis is that, since EMR is mainly excited by powerline ac voltage that oscillates at the same rate as the ENF, it should also carry the ENF natural timestamps. In addition, to foster adoption at low cost, we investigate simple EMR reception devices that can nevertheless capture the ENF natural timestamps effectively. Our results reveal that significant time information is carried by wireless EMR. This understanding extends the applicability of the ENF natural timestamping to nodes that are not connected directly to the grid. These nodes represent a major sector of sensor and IoT deployments.

Unfortunately, compared with strong sinusoidal ac voltage signals sensed directly at wall power outlets, the EMR signal is weak and noisy. First, as EMR has an extremely long wavelength, its signal reception at small sensors is weak. Second, although the EMR is mainly excited by the ac voltage, it is nevertheless also affected by changing electric currents in powerlines and appliances. Thus, changes in the appliances' states and operations can create continuous and significant random noises and transient disturbances to the EMR. Third, the EMR can be similarly disturbed by other environment factors such as human movements. These characteristics pose significant challenges to capturing minute ENF fluctuations of less than 0.1 Hz that constitute the natural timestamps.

In this paper, we address the above challenges by analyzing representative IoT nodes as EMR reception devices and designing efficient EMR conditioning algorithms on resource-constrained sensor platforms. The inquiry is driven by the end performance goal of time-accurate EMR natural timestamps. Specifically, to understand the effectiveness of EMR reception, we use two typical classes of IoT nodes: i) the Raspberry Pi (RPI) single-board computer [1]

equipped with a customized EMR reception peripheral, and ii) the MSP430-based Z1 mote [6] with a normal conductor wire acting as EMR antenna. For both platforms, we develop a unified real-time signal processing pipeline based on an infinite impulse response (IIR) filter to reliably extract the ENF trace from the weak and noisy EMR measurements. We deploy the EMR sensors at five sites (a commercial building, two homes, and two campus buildings) in a city, which are away from each other for up to 24 km. Extensive experiments at these five sites over long periods of time (up to one month) show that the EMR natural timestamps captured by the RPi-based node and the Z1 mote give median time errors of about 50 ms and 150 ms, and 95th-percentile time errors of about 250 ms and 780 ms, respectively. Such CNoT accuracy is sufficient for a wide range of WSN and IoT applications, e.g., building monitoring and control, utility smart metering, and city-scale power grid state estimation at the granularity of minutes [26].

To demonstrate how to use the EMR natural timestamping approach for resilient CNoT, we discuss two applications, i.e., i) recovery of time information for nodes without network access and ii) run-time verification of the integrity of clocks at network nodes, which is a practical problem in real systems facing various hardware and software faults [20]. Evaluation results on the clock integrity verification are presented.

The rest of this paper is organized as follows. Section 2 reviews the problem's background and related work. Section 3 presents key hardware and algorithm designs for EMR natural timestamping. Section 4 presents a low-cost implementation of the proposed technique, and discusses benchmark results that guide the settings of several important parameters of the natural timestamping algorithm. Section 5 presents the results of extensive experiments conducted at the five sites. Section 6 discusses applications of EMR natural timestamping. Section 7 concludes.

2 BACKGROUND

2.1 ENF Fluctuations and Powerline EMR

In an ac power grid, the ENF is regulated at a nominal value (50 or 60 Hz) by a grid-wide centralized control system [26]. ENF has the following two properties from power engineering [26]. First, at any given time instant, the ENF is almost identical across all locations within a geographical region (e.g., building, factory, or city). Second, as the load of a power grid is a random process, the ENF regulated by the centralized control system is also a random process. We note that distributed generation is increasingly connected to the power grid. The ac electricity locally generated must be synchronized precisely both in frequency and phase with the power grid. Thus, the local power injection back to the power grid will not affect the above two properties. Fig. 1 shows the per-second ENF measurements during one hour collected by two high-precision voltage sensors connected to the wall power outlets at two locations 12 km apart in our city. We can see that i) the ENF fluctuates by less than 0.1 Hz, and ii) the fluctuations are identical at the two locations. Thus, it is possible to match a short segment of ENF trace collected at a location against a long ENF trace collected at the other location. The best match point indicates a common time instant when the two nodes obtain their respective ENF samples.

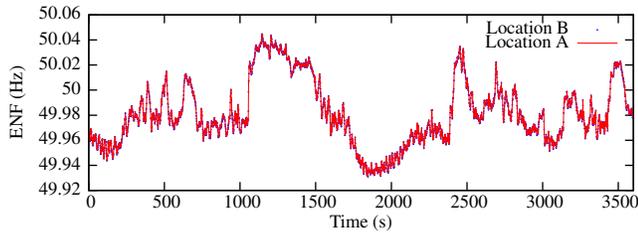


Figure 1: Per-second ENF measured at two locations 12 km apart in a city. Mean error: 0.1 mHz; RMSE: 0.2 mHz.

Ac powerlines and electrical appliances can emit EMR that oscillates at the same rate as the ENF. Fig. 2(a) shows the ac voltage signal collected by a high-precision voltage sensor from a wall power outlet and the EMR signal recorded by another sensor. The two sensors are located in the same building. We can see that the EMR signal is not an ideal sinusoid. Moreover, the EMR signal has a phase lag compared with the ac voltage signal collected from the power grid directly. Fig. 2(b) shows the ENF traces estimated from the cycle lengths of the ac voltage and the EMR signal, respectively. The EMR-based ENF fluctuates around the actual ENF by up to 2 Hz, significantly exceeding the peak fluctuation of the actual ENF on the order of 0.1 Hz, as shown in Fig. 1. The key question we ask in this paper is whether the noisy EMR data contains ENF natural timestamps at sufficient time accuracy.

2.2 Related Work

EMR can generate interference in the circuits of audio and video recorders. Fluorescent lights, which flicker at a frequency twice that of ENF, can also generate ENF-related visual interference to video cameras. Thus, multimedia recordings often contain information related to ENF fluctuations. This property has been used in forensic applications [10, 13, 14, 24]. For instance, it is possible to authenticate the recording time of an audio/video clip by matching an ENF trace extracted from the clip against a historical ENF database recorded directly from the power grid. Different from these forensic studies that focus on extracting the *passively* recovered ENF traces from multimedia files, we focus on developing a lightweight approach for wireless sensors to *actively* capture EMR and push the accuracy limit, using low-cost hardware, of the global time information contained in the captured EMR data. Based on our results, various applications besides forensics can be developed, such as those discussed in Section 6 of this paper.

EMR and fluorescent light flickering have been exploited for clock calibration for WSNs and mobile devices. Rowe *et al.* [23] design a mote peripheral to receive EMR and calibrate the clocks of motes based on the detected ac cycles. Li *et al.* [19] leverage fluorescent light flickering to calibrate the clocks of nodes equipped with light sensors. Their approach [19] requires fluorescent lighting, which may not be available everywhere. Our work is fundamentally different from these two prior studies [19, 23]. The approaches in [19, 23] leverage the periodicity of EMR or fluorescent flickering to achieve *clock calibration*, which ensures that different clocks advance at the same speed. In contrast, our natural timestamping approach exploits the global time information encoded in

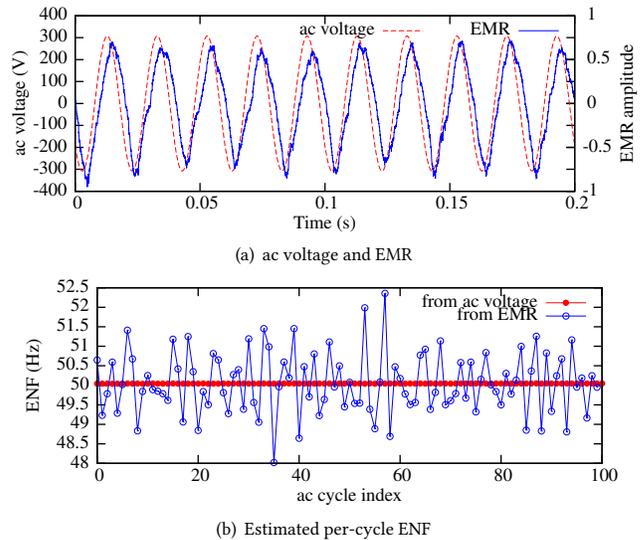


Figure 2: Ac voltage and EMR data.

the grid-wide *imperfections* of the EMR's periodicity. During the natural timestamping phase, no packets are transmitted. The natural timestamps of two nodes are exchanged only when they need to establish CNoT.

Besides the above ENF-based clock calibration [19, 23], AM/FM radios [7, 18] and Wi-Fi beacons [16] have been exploited for clock synchronization and calibration. (Different from clock calibration, *clock synchronization* regulates the clocks to have the same value.) The approaches based on AM/FM radios [7, 18] need sophisticated radio receiver circuits to extract the time broadcasts. In contrast, our approach requires simple EMR reception devices only, such as a normal conductor wire. The approach based on Wi-Fi beacons is applicable to a local area. In contrast, our approach can be used across a geographic region served by the same power grid (e.g., a city area as evaluated in this paper).

Natural timestamping has been studied for a few other sensing modalities including seismic [20] and sunlight [15]. Lukac *et al.* [20] use microseisms received by seismic sensors and a microseism model to recover the timestamps of the data captured by the sensors with faulty clocks. However, their approach is developed specifically for seismic sensors. Gupchup *et al.* [15] exploit the cyclic pattern of sunlight to recover timestamps. However, their approach can achieve minutes accuracy only. Moreover, their approach is not applicable for indoor environments without sunlight.

As discussed in Section 1, this paper extends the applicability of the ENF natural timestamping from grid-connected nodes, as demonstrated in prior work [25], to nodes that are not connected directly to the grid. The approach in [25] uses a sequence of the cycle lengths of ac voltage as the natural timestamp. In this paper, we use the average frequency over multiple EMR cycles as the natural timestamp. This different design is due to the noisy nature of the EMR as shown in Section 2.1. Moreover, to further reduce the impact of noises, we employ an EMR conditioning algorithm that can be efficiently executed on resource-constrained platforms.

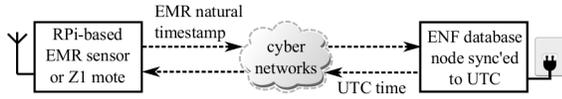


Figure 3: Illustration of an EMR natural timestamping system.

3 EMR NATURAL TIMESTAMPING

This section presents the hardware and algorithms for validating the EMR natural timestamping.

3.1 Hardware

We use three types of natural timestamping hardware devices in this paper, i.e., Raspberry Pi (RPI) [1] based EMR sensors, Zolertia Z1 motes [6], and an ENF database node. The RPI-based node has a dedicated EMR sensor and samples the EMR signal at up to 44.1 kHz. The Z1 mote uses a normal conductor wire as the EMR antenna and samples at sub-kHz such that the EMR data can be processed in real time given the Z1’s limited processing capability. Thus, the RPI-based sensor and Z1 mote represent high-end and low-end sensors, respectively. The results obtained on the two platforms offer important insights into how hardware and processing capability may impact the effectiveness of natural timestamping. The ENF database node, synchronized to global time, uses a high-precision ENF sensor to record ground-truth ENF data directly from a wall power outlet. As we will discuss later (Section 3.3), the ground-truth ENF data is used to decode the EMR natural timestamps captured by the RPI-based sensors and Z1 motes into global time. Fig. 3 illustrates a simple EMR natural timestamping system, in which an EMR sensor transmits a natural timestamp to the database node and the database node decodes it and sends back the global time at which the natural timestamp was captured. The database node can serve multiple EMR sensors.

3.1.1 RPI-based EMR sensor. Fig. 4(a) shows the schematics of the proposed EMR sensor. It shares a similar design as the one in [23]. Specifically, it includes an analog front-end (AFE) and an LC antenna consisting of a 470 mH coil inductor and a 22 μ F capacitor. The inductor generates currents in the EMR’s changing magnetic field. Through impedance matching by the capacitor, the LC circuit’s theoretic resonant frequency is 49.49 Hz, which is very close to the nominal ENF of 50 Hz in our region. Thus, we expect good EMR reception. We also tested other combinations of candidate inductors and capacitors that produce theoretic resonant frequencies close to 50 Hz. However, compared with the 470 mH and 22 μ F combination, the other combinations gave inferior EMR reception, potentially due to their larger resistance that leads to actual resonant frequencies different from 50 Hz.

The output of the LC circuit is processed by the AFE to adapt to a sampling board. Specifically, the signal is amplified by a two-stage amplification circuit. The first stage is by an instrumentation amplifier INA126P that gives an open-loop gain of 4,450. The second stage, based on an operational amplifier LM124J, gives a gain of about 21. Two high-pass filters are applied after the two amplifiers respectively to remove the direct current (dc) component. The output of the AFE swings within $[-1\text{ V}, 1\text{ V}]$.

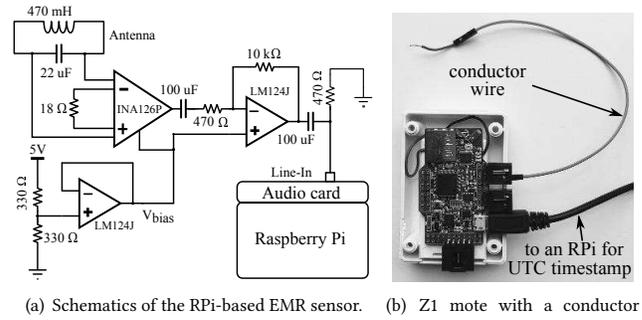


Figure 4: EMR capture devices.

We use a Cirrus Logic audio card mounted on the RPI to sample the output of the AFE. This audio card provides a 24-bit resolution and up to a 192 kHz sampling rate, both of which are much higher than that of other off-the-shelf analog-to-digital converter (ADC) boards for RPI. The output of the AFE is wired to the line-in port of the audio card. The RPI runs Raspbian OS. In our experiments, we configure the audio card to operate at 16-bit resolution, and 44.1 kHz or 8 kHz sampling rate. As the audio card sampling introduces a 4% CPU utilization only, we do not observe significant jitters in the sampling. The RPI uses NTP [4] to synchronize its clock with a local stratum-1 time server equipped with a GPS receiver. From NTP’s status report, the RPI’s synchronization error is around one millisecond. With the synchronized clock, the RPI adds Coordinated Universal Time (UTC) timestamps to the EMR measurements. The UTC timestamps are used as ground truth to evaluate the error of the time information derived from the ENF fluctuations. The whole setup is powered by a high-capacity dc power bank to make sure that the node senses EMR only, rather than ac voltage leaked from an ac/dc power supply unit.

3.1.2 Z1 mote. The Z1 mote [6] is equipped with an MSP430 microcontroller (MCU). It has two Phidgets sensor ports connected to several ADC pins of the MCU. The Z1 mote runs TinyOS 2.1.2. As shown in Fig. 4(b), we connect a normal conductor wire to a pin in a Phidgets port and use the MCU’s ADC to sample the electric potential generated by the EMR on the wire. Such a configuration minimizes the use of extra hardware. In addition, the power consumption of capturing EMR is merely the MCU’s power in sampling and processing the signal. In applications of EMR natural timestamping, the EMR data is typically collected for a short duration infrequently or on an on-demand basis. These applications will introduce little power consumption. The simplicity of the EMR reception device and the low power footprint can foster the adoption of EMR natural timestamping in WSN and IoT systems.

To facilitate evaluation, we connect the Z1 mote to an RPI via a USB cable for power supply and data storage. The RPI is powered by a dc power bank and synchronized with the aforementioned local stratum-1 time server. Once an ENF measurement is available, the Z1 mote sends it over the USB cable to the RPI for ground-truth UTC timestamping. Note that in real applications, the RPI is not needed.

3.1.3 ENF database node. The ENF database node consists of an ENF capture device connected to a wall power outlet and an RPi equipped with an Adafruit GPS receiver for UTC synchronization. The ENF capture device uses an ac/ac transformer to step down the 220 V ac voltage and a signal processing circuit to condition the signal and generate interrupts upon zero crossings of the ac voltage to an MCU. The MCU uses a hardware timer running at 8.4 MHz to measure the time periods between consecutive interrupts, i.e., the cycle lengths of the ac voltage. The MCU streams the measurements to the RPi for UTC timestamping. The timestamped ac cycle length measurements are further streamed to a server for long-term storage. The ac cycle length measurements can be easily translated to ENF measurements. Our ENF capture device provides a resolution of 0.3 mHz for measuring the ENF.

From our measurements using multiple database nodes distributed in our city, the recorded ENFs across the city are almost identical with a mean error of 0.1 mHz. Thus, for our experiments conducted in the city, a single ENF database node suffices. We leave the problem of how to deploy ENF database nodes for an IoT system distributed in a large-scale (e.g., state-scale) power grid to future work.

3.2 ENF Extraction

This section presents the algorithm for extracting the ENF trace from the EMR measurements. This algorithm will be used by the RPi-based EMR sensor and the Z1 mote. We adopt a time-domain approach to detect the zero crossings of the near-sinusoidal EMR signal. From our result, this approach can achieve a precision of a few mHz in estimating an ENF value based on EMR measurements over one second. Alternatively, a frequency-domain approach can apply the fast Fourier transform (FFT) to the EMR measurements and then find the principal frequency as the ENF. However, as FFT needs x seconds of data to generate an ENF estimate with a resolution of $1/x$ Hz, it will need more than 16 minutes of EMR data to achieve a 1 mHz resolution. Thus, the FFT approach cannot track the ENF fluctuations at sub-second time resolutions that are critical for natural timestamping.

With the detected zero crossings, we use a sliding window approach to compute the ENF trace. Specifically, for each window consisting of K ac cycles, an ENF measurement is computed as $f = \frac{K}{\sum_{i=1}^K t_i}$, where t_i is the time duration of the i th ac cycle that is timed by the RPi's or Z1's local clock. The window slides by one ac cycle each time. A larger setting of K reduces the noise of f due to the averaging, but also reduces the sensitivity to ENF changes. In Section 4, we will evaluate its setting to achieve the best overall accuracy of the natural timestamping. Robust detection of zero crossings is a key to the success of the time-domain approach for extracting ENF. However, this task is non-trivial, due to primarily considerable noises in the EMR signal. In the rest of this section, we firstly discuss the challenges of this task and then propose a band-pass filtering approach as solution.

3.2.1 Challenges of zero crossing detection. EMR signal often contains both high- and low-frequency noises. A hardware band-pass filter with a steep passband boundary is often complicated. Moreover, the Z1 mote has no circuits to condition the EMR signal. Thus, it is important to remove the noises in software.

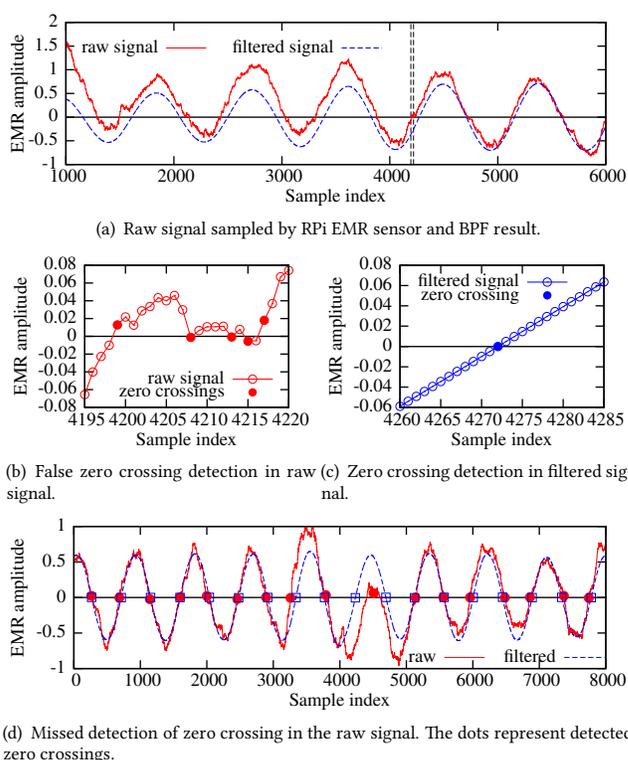


Figure 5: Challenges of zero crossing detection in noisy EMR signal, and effectiveness of BPF.

We use real signals captured by an RPi-based EMR sensor to illustrate the challenges. The solid curve in Fig. 5(a) shows the EMR signal obtained by the EMR sensor shown in Fig. 4(a). We can see that the near-sinusoidal signal has many high-frequency spurs that can lead to false zero crossing detection. For instance, Fig. 5(b) shows a zoomed-in view of the time duration within the two vertical lines in Fig. 5(a). The signal staggers around zero, leading to repeated zero crossing detection if no special signal processing step is used to deal with this issue. A *dead-zone* approach could suppress the repeated detection. Specifically, it will not report successive zero crossings within a dead zone of several milliseconds after a detection. However, this approach will lead to missed detection in the presence of low-frequency noises, such as those caused by nearby human movements. In Fig. 5(d), from the 3000th to the 6000th samples, the EMR sensor experiences a transient low-frequency disturbance and the signal barely crosses the zero line. Because of the dead zone, the detector misses a zero crossing.

The first two subfigures of Fig. 6 show the results of the dead-zone approach. When the dead zone is set to be 3 ms, the approach generates repeated zero crossings due to high-frequency noises. Thus, the estimated ENF can be a few Hz higher than the ground-truth ENF. When the dead zone is set to be 5 ms, although the approach generates fewer repeated zero crossings, it will miss more actual zero crossings. As a result, the approach yields many wrong ENF measurements that are several Hz lower than the ground truth.

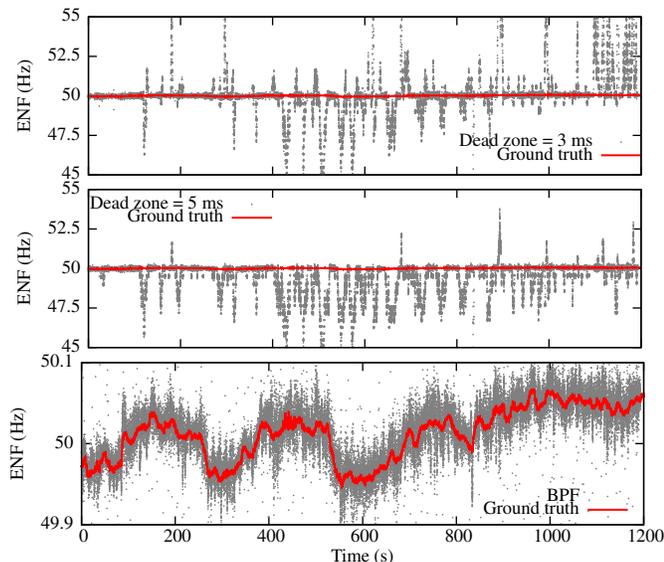


Figure 6: Comparison between the dead-zone approach and our BPF-based approach (sliding window size $K = 50$).

The above results illustrate that the dead-zone approach cannot effectively eliminate false positives and negatives simultaneously. Although the dead zone can be fine tuned and more heuristic rules can be added to reduce false positives/negatives, the performance of the tuning and piecemeal heuristics tend to be data dependent. A sharp approach that directly addresses the root cause of the high-/low-frequency noises is desirable.

In [23], a phase-locked loop (PLL) is used to deal with the zero crossing jitters and misses due to low-frequency disturbances. However, the objective of the PLL is to generate an extremely stable periodic signal driven by EMR to calibrate the mote’s clock, which is different from our objective of tracking ENF fluctuations at fine time resolutions. Although the parameters of the PLL can be tuned to reduce its convergence time and improve tracking performance, the stability of the PLL will be worsened due to its closed-loop nature.

3.2.2 Band-pass filtering. The results in Section 3.2.1 clearly suggest the necessity of conditioning the EMR signal to remove the high-/low-frequency noises simultaneously. To this end, we apply a 5th-order/6-tap infinite impulse response (IIR) band-pass filter (BPF) that gives steep boundaries of the passband. We use Scipy’s IIR design tool to choose the filter parameters to give a passband of [45 Hz, 55 Hz]. Then, we implement the filter in C and nesC for the RPi-based EMR sensor and the Z1 mote, respectively. As shown in Fig. 5(a) and 5(c), the filtered signals represented by the dashed curves are smooth and have clear zero crossings. In Fig. 5(d), during the transient time period of the low-frequency disturbance, the EMR signal is restored by the BPF. The bottom subfigure of Fig. 6 shows the ENF measurements by our BPF-based approach. We can see that our approach can well track the ground-truth ENF measured by the database node. The mean error is 0.7 mHz and the root mean square error (RMSE) is 44 mHz.

The above results show the effectiveness of BPF in achieving robust zero crossing detection. However, the implementation of the IIR filter on resource-constrained platforms such as Z1 mote is difficult, due to its compute complexity. The implementation challenges will be discussed in Section 4.2.

3.3 Natural Timestamp and Decoding

From the bottom subfigure of Fig. 6, the EMR sensors can track the fluctuations of ENF. This implies that it is possible to *time-align* an ENF trace captured by an EMR sensor against an ENF trace captured by the database node. In other words, the database node can “decode” the time during which the ENF trace was captured by the EMR sensor. Thus, we call a finite-length ENF trace captured by an EMR sensor a *natural timestamp*. The process of identifying the UTC time when the natural timestamp was captured is called *decoding*. This section presents the decoding algorithm. Extensive evaluation of the accuracy will be presented in Section 4 and Section 5.

A natural timestamp, denoted by \mathbf{f} , is a vector of n consecutive ENF measurements by an EMR node. Let a vector \mathbf{g} denote a trace of m consecutive ground-truth ENF measurements by the database node. The measurements in \mathbf{g} are timestamped with UTC. Moreover, the following condition needs to be satisfied for natural timestamp decoding:

Decoding condition C1: The time duration of measuring \mathbf{f} is within the time duration of measuring \mathbf{g} .

The above condition can be verified using an approach presented in Section 6.2. The decoding is to match \mathbf{f} with a ground-truth natural timestamp within a window of size n in \mathbf{g} using a dissimilarity metric, e.g., RMSE. Specifically, by sliding the window within \mathbf{g} , the UTC timestamp of the first element of the window that yields the minimum dissimilarity is identified as the UTC time for the first element of \mathbf{f} . Formally, the index of the window that yields the minimum dissimilarity is given by

$$i^* = \arg \min_{i \in [1, m-n+1]} \text{dissimilarity}(\mathbf{f}, \mathbf{g}[i : i+n-1]), \quad (1)$$

where $\mathbf{g}[i : i+n-1]$ represents a vector consisting of the i th to $(i+n-1)$ th elements of \mathbf{g} . Then, the decoding algorithm outputs the UTC timestamp of $\mathbf{g}[i^*]$ for $\mathbf{f}[0]$. Thus, the time when \mathbf{f} is captured is known.

As discussed in Section 3.1, both the RPi-based EMR sensor and the Z1 mote add UTC timestamps to the captured EMR measurements for evaluation purpose. For a natural timestamp, the difference between the decoded UTC time and the UTC time recorded by the sensor is the *decoding error*. To understand the overall accuracy, we decode many natural timestamps from a long ENF trace captured by an EMR sensor and present various statistics of the decoding errors.

4 IMPLEMENTATION & BENCHMARK

This section presents a set of benchmark experiments to guide the settings of several important parameters of the natural timestamping. Extensive performance evaluation of the natural timestamping will be presented in Section 5.

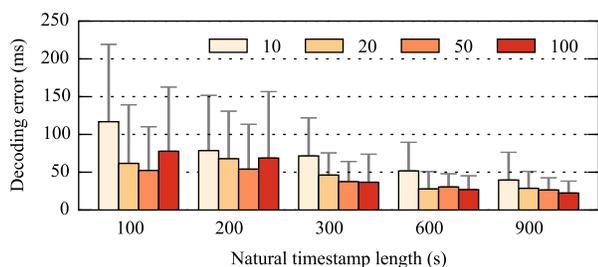


Figure 7: Mean decoding error vs. natural timestamp length under different K . Error bar represents standard deviation.

4.1 Timestamping Parameter Settings

Our natural timestamping approach presented in Section 3.2 has three key parameters: the EMR signal sampling rate, the window length K for estimating an ENF value, and the natural timestamp length n . In this section, we extensively evaluate the impact of their settings on the natural timestamping performance. We use the RPi-based EMR sensor to collect a one-hour EMR data trace at a sampling rate of 44.1 kHz. The natural timestamps generated from this data trace are decoded using a one-hour ground-truth ENF trace collected by the database node. The RPi-based EMR sensor is deployed in an office pantry with several electrical appliances; the ENF database node is deployed in a workspace in the same office. The decoding results under different parameter settings are presented below.

First, we jointly evaluate the settings of K and n . Fig. 7 shows the decoding error versus the natural timestamp length n under different settings of K . To facilitate comparison, we translate the natural timestamp length n to the time duration of the natural timestamp and use it as the X -axis of Fig. 7. Under the setting $K = 50$ (i.e., each ENF value is computed from the EMR measurements in one second), the average decoding errors are generally lower than or comparable to those under other K settings. Thus, in the rest of this paper, we adopt $K = 50$ unless otherwise specified. Under this setting, when the natural timestamp length is 300 seconds, the average decoding error is 37 ms only.

Second, we jointly evaluate the settings of sampling rate and natural timestamp length. To simulate different sampling rates, we downsample the original EMR trace to 7.4 kHz, 3.7 kHz, and 0.7 kHz. Note that the 0.7 kHz and 3.7 kHz are the limits for the Z1 to perform real-time and offline ENF extraction, respectively (cf. Section 4.2). Fig. 8 shows box plots of the decoding errors versus the timestamp length under various sampling rates. The sampling rates of 7.4 kHz and 3.7 kHz give similar decoding errors. Compared with the results in Fig. 7, the decoding errors in Fig. 8 are generally higher due primarily to the reduced sampling rates, which is consistent with intuition. Nevertheless, with a sampling rate of 3.7 kHz and a timestamp length of 300 s, the average decoding error is 44 ms, which is similar to the result in Fig. 7. If the sampling rate is 0.7 kHz, the average decoding error is below 500 ms when the timestamp length is larger than 300 s. Thus, we

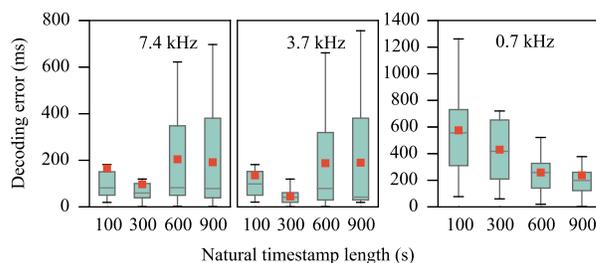


Figure 8: Impact of sampling rate and timestamp length on the RPi-based EMR sensor. Grey line represents median; red square dot represents mean; box represents the (25%,75%) range; upper/lower bar represents maximum/minimum.

can see that the node’s processing capability may limit the performance of natural timestamping due to limited achievable sampling rate.

In the first two plots of Fig. 8, when the natural timestamp length is larger than 300 s, the decoding error increases with the natural timestamp length. In the third plot of Fig. 8, the decoding error decreases with the natural timestamp length. This is because the EMR sampling rate and timestamp length jointly affect the decoding errors. When the EMR sampling rate is high, the sensor can capture many random noises in EMR. Thus, a longer timestamp will likely include more noises and potentially increase the decoding errors. This understanding is consistent with the observation from the first two plots, where the sampling rate is relatively high (7.4 kHz and 3.7 kHz). When the sampling rate is relatively low (e.g., 0.7 kHz in the third plot), a longer timestamp generally results in smaller decoding errors.

4.2 Z1 Implementation and Benchmark

This section presents preliminary benchmark experiments conducted on a Z1 mote deployed in an office pantry to understand its natural timestamping capability. The first set of experiments specifically evaluates the fidelity of the EMR signal captured by the Z1’s simple EMR reception device. The Z1 mote transmits the EMR measurements directly to the attached RPi through the USB cable. The Z1 mote adopts the maximum sampling rate of 3.7 kHz that can be supported by the UART interface. Fig. 9 shows a one-hour ENF trace computed from the EMR measurements captured by the mote. We can see that the ENF trace consists of discrete values, with a resolution of about 13 mHz. This is because of the limited sampling rate.¹ Nevertheless, the Z1 mote’s ENF trace well tracks the ground-truth ENF trace. Fig. 10(a) shows a box plot of the decoding errors under various natural timestamp lengths. By comparing Fig. 10(a) and Fig. 8, the Z1 mote generally yields larger decoding errors. This is expected, since the Z1 mote uses a normal conductor wire for capturing the EMR. Nevertheless, the Z1 mote still gives sub-second average decoding errors.

¹With a sampling rate of 3.7 kHz, a zero-crossing detection error of one EMR sample leads to an ENF estimation error of 660 mHz. The averaging in each sliding window reduces the error to 13 mHz.

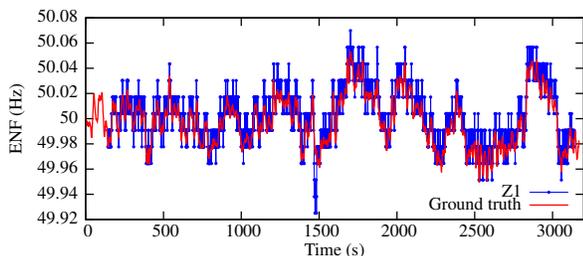


Figure 9: ENF trace computed based on EMR measurements captured by Z1 mote. Sampling rate is 3.7 kHz.

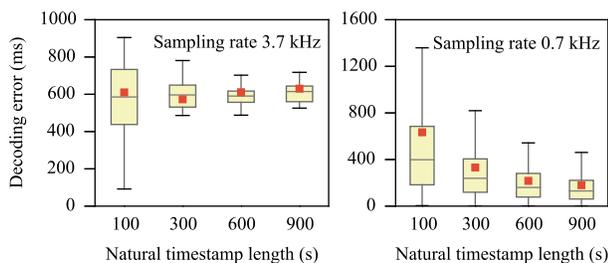


Figure 10: Impact of sampling rate and timestamp length on the timestamping performance of Z1 mote.

The second set of experiments evaluates the impact of the Z1’s limited processing capability on the natural timestamping performance. We have implemented the IIR BPF algorithm using nesC on the Z1. By extracting the ENF trace in real time, the Z1 does not need to store the massive EMR data. However, without special care in implementing the IIR filter, its high compute complexity may greatly limit the mote’s ability to achieve a satisfactory sampling rate. The filter needs many floating-point multiplications that consume a significant number of MCU cycles. Our test shows that, without code optimization, the Z1 mote can sustain a sampling rate of 90 Hz only. To address this, we apply the Horner algorithm [2], which implements floating-point multiplication with integer operations. As a result, the Z1 mote can sustain real-time ENF extraction at a sampling rate of 0.7 kHz. Fig. 11 shows a one-hour ENF trace estimated by the Z1 mote based on the EMR signals sampled at 0.7 kHz. We can see that the ENF estimates are at discrete levels with a low resolution, which is caused by the limited sampling rate as discussed previously. It is interesting to understand whether the low-resolution natural timestamps contain global time information of satisfactory accuracy. Fig. 10(b) shows a box plot of the decoding errors. The decoding errors in Fig. 10(a) and Fig. 10(b) are comparable. For some natural timestamp length settings (e.g., 900 s), the decoding error in Fig. 10(b) is even lower than that in Fig. 10(a). As the Z1 mote uses a simple conductor wire as the EMR antenna, its measurements contain larger noises. Thus, as discussed in Section 4.1, a higher sampling rate in Fig. 10(a) may result in the inclusion of more noises and increase the decoding errors.

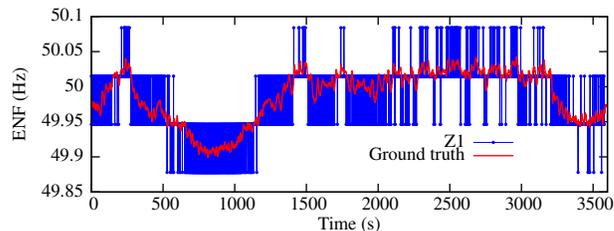


Figure 11: ENF trace estimated by the Z1 mote based on EMR signals sampled at 0.7 kHz.

Table 1: Various sites for performance evaluation.

| Site | Description | Distance from A.0 |
|------|----------------------------|-------------------|
| A | office building | |
| A.0 | database node | |
| A.1 | work space | 9 m |
| A.2 | pantry | 14 m |
| B | living room of Apartment I | 10 km |
| C | study room of Apartment II | 12 km |
| D | office at Campus I | 19 km |
| E | office at Campus II | 24 km |

The above benchmark results show that the Z1 mote can achieve satisfactory natural timestamping performance.

5 PERFORMANCE EVALUATION

This section presents an extensive set of experiments at five deployment sites conducted over long periods of time (up to one month) to validate the EMR natural timestamping.

5.1 Deployments and Settings

We choose five sites, as summarized in Table 1, to evaluate the natural timestamping performance. For each site, we deploy a RPi-based EMR sensor and a Z1 mote. Note that we deploy a database node at Site A. The distances given in Table 1 are roughly estimated Euclidean distances between each deployment site and the database node. The Sites A.1 and A.2 are two spaces in Site A. At Site A.1, the nodes are placed on an occupied working desk with a desktop computer, a laptop, and various chargers. Site A.2 is a pantry with various appliances including a fridge and a microwave. The nodes are placed in a corner of the pantry. Site B is the living room of an apartment. The nodes are placed on a TV stand. Site C is the study room of a different apartment. The nodes are placed on a computer desk. Site D is an office on a campus. Site E is an open office space on a different campus. For evaluation purpose, the nodes at Sites A.1, A.2, and E are synchronized with local stratum-1 time servers equipped with GPS receivers; at the other sites, due to difficulty of deploying GPS receivers, the nodes are synchronized with Internet time servers in our city.

For the RPi-based sensor and the Z1 mote, the sampling rates are 8 kHz and 0.7 kHz, and the natural timestamp lengths are 300 s and 600 s, respectively. A one-hour ENF trace that is recorded by

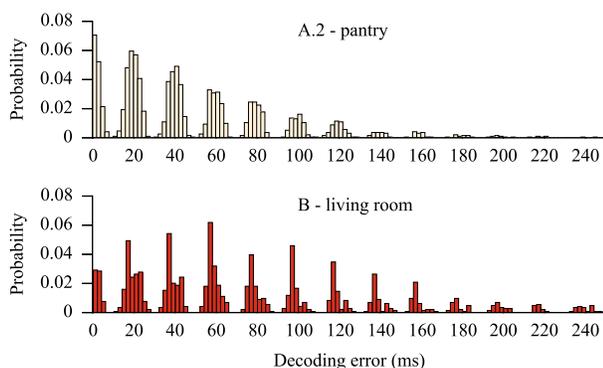


Figure 12: Distribution of decoding errors of RPi-based EMR sensor at Site A.2 and Site B. Histogram bin width is 2 ms.

Table 2: Decoding error statistics at various sites.

| Site | RPi-based EMR sensor | | Z1 mote | |
|------|----------------------|----------------------------------|-------------|----------------------------------|
| | median (ms) | 95 th percentile (ms) | median (ms) | 95 th percentile (ms) |
| A.1 | 59 | 204 | 150 | 786 |
| A.2 | 40 | 128 | 146 | 499 |
| B | 82 | 357 | 165 | 580 |
| C | 56 | 138 | 142 | 523 |
| D | 37 | 136 | 160 | 542 |
| E | 59 | 240 | 220 | 780 |

the database node and satisfies the decoding condition C1 is used for natural timestamp decoding.

5.2 Evaluation Results

5.2.1 Performance evaluation at various sites. Fig. 12 shows the distribution of the decoding errors of the RPi-based EMR sensor at Site A.2 and Site B over eight days. As discussed in Section 3.2, we slide the window by an ac cycle each time to compute an ENF value. Thus, ideally, the decoding error should also be a multiple of the ac cycle. From Fig. 12, the decoding error clusters are indeed centered at multiples of the ac cycle 20 ms. As the EMR signal is also affected by the environment, we can see that the decoding errors are also distributed around the multiples of 20 ms. At Site A.2, a majority of the decoding errors are below 2 ms and all the decoding errors are below 298 ms. Site B gives a different decoding error distribution, with the highest bar at around 60 ms. This is because the deployment position at Site B gives a weaker EMR signal strength.

Table 2 summarizes the medians and 95th percentiles of the RPi-based EMR sensor and the Z1 mote at the different deployment sites. For the RPi-based EMR sensor, the median decoding error ranges from 37 ms to 82 ms and the 95th percentile is below 357 ms. Due to the simple EMR antenna and limited sampling rate, the Z1 mote yields higher decoding errors. The median is from 142 ms to 220 ms and the 95th percentile is below 786 ms. The results show

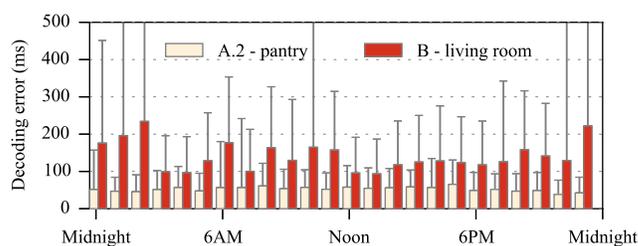


Figure 13: Decoding errors in one month. Each error bar is computed based on the decoding errors in the same time slot of the 30 days. Error bar represents standard deviation.

that the EMR natural timestamping works well in various environments, when the distance between an EMR sensor and the database node is up to 24 km. This set of experiments validates the applicability of the EMR natural timestamping for city-scale WSN and IoT systems.

5.2.2 One-month evaluation. We deploy two RPi-based EMR sensors at Site A.2 and Site B for one month. Fig. 13 shows the error bars of the decoding errors in different time slots at the two deployment sites. At the two sites, the average decoding errors are around 50 ms and 150 ms, respectively. Larger decoding errors are observed when human activities and electrical appliance usage increase. The one-month deployments at the two sites show that the timestamping approach can work reliably over long periods of time.

5.2.3 Impact of deployment position. The EMR signal attenuates with physical distance and it can be affected by the operations of nearby electrical appliances. In this set of experiments, we deploy the RPi-based EMR sensor at different spots at Site B, which is a real home environment. The triangles in Fig. 14 represent the deployment spots. At each spot, the sensor collects the EMR trace for one hour. The rectangle labels in Fig. 14 show the median decoding errors at different deployment spots. We can see that the decoding errors vary with the deployment spots, from 62 ms to 338 ms. From more extensive measurements in the apartment, the RPi-based sensor can always receive a salient EMR signal for natural timestamping.

We also conduct a set of experiments to evaluate the impact of distance from powerlines on the decoding errors. As shown in Fig. 14, we increase the distance of the EMR sensor from a TV stand in the living room up to 1.8 m. We choose the TV stand because it contains a collection of power cords that supplies a TV, a table fan, and a Wi-Fi router. The rectangular labels at the right bottom of Fig. 14 show the median decoding errors at different distances from the TV stand. As the EMR signal strength attenuates with distance, the zero-crossing detection will be more likely affected by environmental disturbances. Thus, from Fig. 14, we can see that the decoding error increases with the distance. In the center of the living room, which is likely the farthest position from the powerlines in the home, the median decoding error is 280 ms.

We also evaluate the performance of the Z1 mote in the apartment. Fig. 15 shows the median decoding error versus the strength

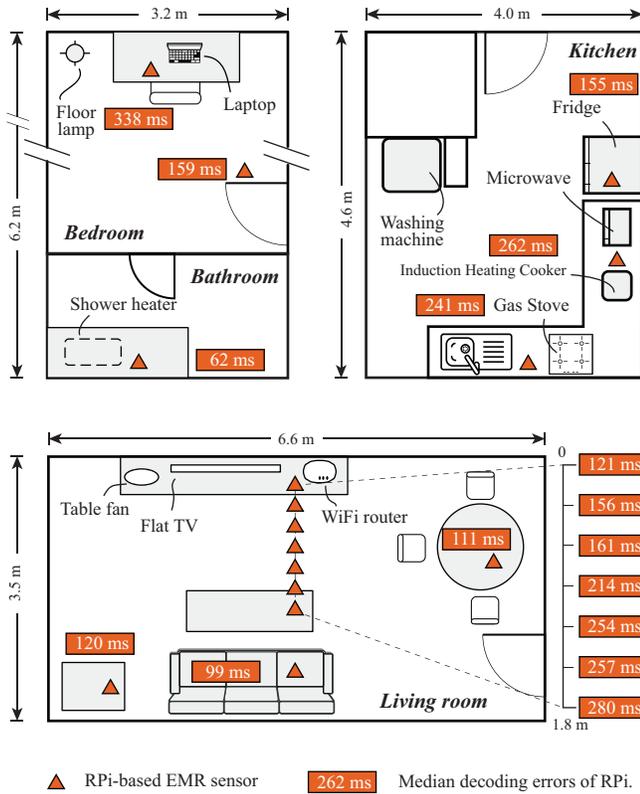


Figure 14: Deployment spots at Site B. An RPi-based EMR sensor is deployed at each spot for one hour.

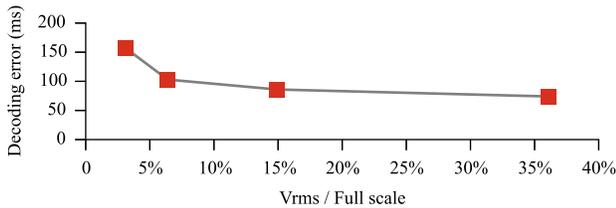


Figure 15: Z1's decoding error vs. EMR signal strength.

of the EMR signal received by the Z1 mote, where the signal strength is measured by the ratio of the root mean square of the raw EMR signal to the full scale of the Z1's ADC. When the signal strength is 3% only, the median decoding error remains at about 150 ms. When the signal strength reduces to 1.2% and below, the decoding gives large errors. From these results, the wireless sensor platforms with very simple EMR reception devices, like the conductor wire used by the Z1 mote in this paper, are suitable for deployments close to powerlines and industrial systems that often produce considerably strong EMR.

5.2.4 *Storage volume constraint.* As discussed in Section 3.2, we adopt a sliding window approach to compute the ENF trace. Thus,

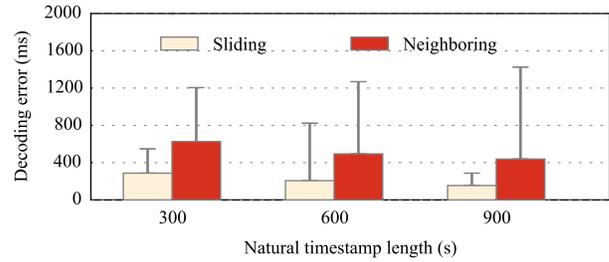


Figure 16: Decoding errors of Z1 under the sliding and neighboring approaches. Error bar represents standard deviation.

an ENF value is generated every 20 ms. As a result, a natural timestamp of 600 s will need 60 kByte storage, which may impose too much storage overhead on resource-constrained nodes. Alternatively, we can adopt a neighboring window approach to compute the ENF trace. When the window size is set to be one second, a natural timestamp of 600 s uses 1.2 kByte storage only. However, the time resolution of the decoding will increase from 20 ms to 1 s. Fig. 16 shows the decoding errors for the natural timestamps generated by the sliding and neighboring window approaches in a one-day experiment. The neighboring window approach yields doubled decoding errors, compared with the sliding window approach. This experiment shows the trade-off between the storage overhead and the natural timestamping accuracy.

6 APPLICATIONS

Based on understanding on the time accuracy of EMR natural timestamps, this section discusses two applications.

6.1 Time Recovery

In the following two scenarios, EMR natural timestamping is a useful method to recover the global time of critical sensor data and system logs.

Network-free data loggers. In various one-off deployments of data loggers found in diagnostic applications, it can be tedious to set up a communication network for logistic reasons. With EMR natural timestamping, the sensor data timestamps based on its local clock can be rectified using EMR natural timestamps in offline data analysis. Specifically, at run time, the logger captures an EMR natural timestamp periodically, and adds a local clock timestamp. In the offline data analysis, the offset of the local clock can be identified by decoding the EMR natural timestamps. As a result, the sensor data timestamps can be rectified using the identified clock offsets. The period of capturing an EMR natural timestamp can be set to make the local clock drift in the period to be commensurate with the inaccuracy of EMR natural timestamping. Typical crystal oscillators found in MCUs and personal computers have drift rates of 30 to 50 ppm [16]. Thus, the period can be set to be one hour such that the clock drift is below 180 ms

within this period, commensurate with the decoding errors in Table 2. As a natural timestamp takes five minutes of sampling time, it introduces little overhead.

System outages. Various system outages (e.g., loss of communication network and failure of centralized time servers) may significantly degrade the synchronism of nodes in the system. Using the same approach described for network-free data loggers, the timestamps of the critical system logs can be rectified in a post-outage analysis.

We note that network unavailability presents the key challenge to accurate time keeping and timestamping in the above two scenarios. The nodes' local clocks based on crystal keep drifting from the global time. Thus, the on-chip logging approach cannot guarantee bounded timestamping errors. Other state-of-the-art approaches include [10, 13, 14, 20, 24]. However, the approach in [20] is developed specifically for seismic sensors. The forensic approaches in [10, 13, 14, 24] are for multimedia recorders, which are generally unsuitable for resource-constrained nodes. Moreover, they can achieve seconds accuracy only. In contrast, EMR is widely available in civil infrastructures and can be easily captured using simple devices. Moreover, our EMR-based approach provides sub-second accuracy.

6.2 Run-Time Clock Verification

A network node may lose clock synchronization because of various unexpected reasons, e.g., hardware clock faults, transient power failures, operating system faults, blocked NTP traffic due to firewall misconfiguration, etc. This has been a practical concern for real systems. For instance, in a two-year deployment of 100 seismic sensors, each equipped with a GPS receiver for clock synchronization, up to 7% of the collected data has time offsets ranging from tens of seconds to 3,000 seconds [20]. EMR natural timestamping provides a promising tool for verifying the integrity of network nodes' clocks at run time. Specifically, a node can reply to a clock verification request with a natural timestamp together with its clock value when the natural timestamp is sampled. Once the natural timestamp is decoded, the clock value can be verified. Our approach gives high confidence in the verification, because it is not affected by network transmission delays.

As discussed in Section 3.3, we use a finite-length ENF trace collected by the database node to decode a natural timestamp. A practical issue is how to determine whether the decoding condition C1 presented in Section 3.3 is satisfied or not, given that the clock value from the node in question may be wrong. We now present an approach to verify this condition. Specifically, from the ENF database, we choose an ENF trace of sufficient length and centered at the claimed clock value of the natural timestamp. We slide a window within the natural timestamp to generate a total of J sub-natural timestamps, as illustrated in Fig. 17. Then, we decode each sub-natural timestamp against the chosen ENF database trace. Let i_j^* denote the decoding result for the j th sub-natural timestamp using the algorithm in Eq. (1). For the j th sub-natural timestamp where $j \in [1, J]$, we define $(i_j^* - j)$ as its *decoding offset*. If the decoding condition C1 is true and there are no errors in decoding each sub-natural timestamp, the decoding offsets of all the sub-natural timestamps should be a constant. If the decoding condition C1 is

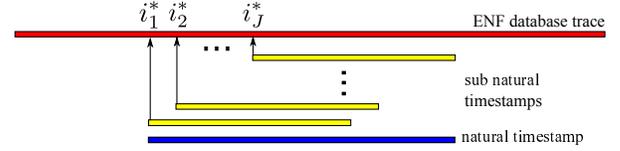


Figure 17: Divide a natural timestamp to sub-natural timestamps to verify the decoding condition C1.

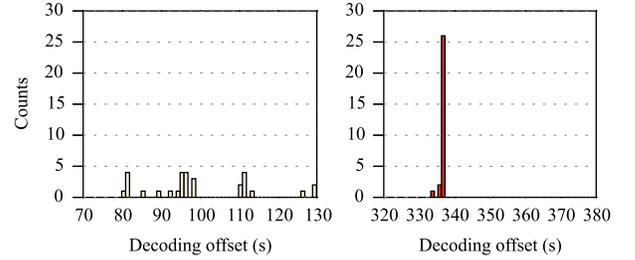


Figure 18: PDF of decoding offsets when the decoding condition is false (left figure) and true (right figure), respectively.

false due to say wrong clock value of the node in question, the decoding offsets will be likely spread out. To address errors in decoding the sub-natural timestamps, we construct a discrete probability density function (PDF) of the decoding offsets. If a significant peak can be found in the PDF, the decoding condition C1 is considered true. For instance, Fig. 18 shows the discrete PDF when the decoding condition C1 is false and true, respectively. In our experiments, if the highest peak in the PDF is at least three times higher than the second highest peak, the decoding condition C1 is considered true.

We evaluate the true/false positive rates, accuracy, precision, and recall of the above approach in determining the decoding condition C1. Specifically, in each test for evaluating the true/false positive rate, we randomly choose a natural timestamp captured by an EMR sensor and a 660-second ENF database trace that satisfies/violates the decoding condition C1 from a one-week dataset. A true positive is a positive test result when the decoding condition C1 is true; a false positive is a positive test result when the decoding condition C1 is false. We can define true negative and false negative accordingly. We follow convention [9] in calculating the true/false positive rates, accuracy, precision, and recall. The results are shown in Table 3. From the table, for the RPi-based EMR sensor, when the natural timestamp length is 200 s or longer, our approach is extremely accurate in determining the decoding condition C1. For the Z1 mote, our approach can also achieve a high accuracy.

If C1 is verified to be false, the node in question does not pass the clock integrity verification; otherwise, we can decode the natural timestamp to obtain the clock offset of the node and determine whether its clock is wrong or not.

Table 3: Measured true positive rate, false positive rate, accuracy, precision, and recall in verifying the decoding condition C1.

| Natural timestamp length (s) | RPI-based EMR sensor | | | | | Z1 mote | | | | |
|------------------------------|----------------------|----------------|----------|-----------|--------|---------------|----------------|----------|-----------|--------|
| | True positive | False positive | Accuracy | Precision | Recall | True positive | False positive | Accuracy | Precision | Recall |
| 60 | 83.3% | 1.0% | 91.1% | 98.8% | 83.3% | 17.4% | 0.6% | 58.4% | 96.6% | 17.4% |
| 100 | 97.1% | 0.0% | 98.6% | 100.0% | 97.1% | 37.1% | 0.9% | 68.1% | 97.6% | 37.1% |
| 200 | 100.0% | 0.0% | 100.0% | 100.0% | 100.0% | 69.3% | 0.0% | 84.7% | 100.0% | 69.3% |
| 300 | 100.0% | 0.0% | 100.0% | 100.0% | 100.0% | 85.7% | 0.6% | 92.6% | 99.3% | 85.7% |
| 600 | 100.0% | 0.0% | 100.0% | 100.0% | 100.0% | 97.3% | 0.3% | 98.5% | 99.7% | 97.3% |

7 CONCLUSION AND FUTURE WORK

This paper answered in the affirmative the question of whether powerline EMR will produce continuous ENF fluctuations that give an accurate and robust fingerprint indicative of time. We called this fingerprint a natural timestamp. We designed a signal processing pipeline for extracting reliably an ENF trace from often weak and noisy EMR wireless signals. We implemented the proposed system on two common sensor platforms that contrast in cost and available nodal resources. By extensive measurements at five deployment sites in a city, we validated that the EMR provides natural timestamps with median time errors down to 50 ms. Based on the results, we discussed two practical applications of EMR natural timestamping, namely time recovery and run-time clock verification.

The work in [25] uses ENF natural timestamps to solve a long-standing threat of packet delay attack [22] that manipulates the transmission times of synchronization packets to degrade the performance of NTP-like synchronization protocols. The approach in [25] can likewise use the EMR natural timestamps to counter the threat for nodes that are not connected to the power grid directly. However, the security of the EMR signal itself needs further investigation, e.g., whether it is feasible to remotely jam and/or interfere with the EMR signals received by the sensors. If so, how robust the EMR-based clock synchronization is under different levels of malicious jamming and interference. These are interesting subjects for future work.

ACKNOWLEDGMENTS

The authors wish to thank our shepherd Dr. Bodhi Priyantha and the anonymous reviewers for providing valuable feedback on this work. This work was funded under the Energy Innovation Research Programme (EIRP, Award No. NRF2014EWTEIRP002-026), administrated by the Energy Market Authority (EMA). The EIRP is a competitive grant call initiative driven by the Energy Innovation Programme Office, and funded by the National Research Foundation (NRF).

REFERENCES

- [1] 2017. (2017). <https://www.raspberrypi.org>.
- [2] 2017. Efficient Multiplication and Division Using MSP430. (2017). <http://www.ti.com/lit/an/slaa329/slaa329.pdf>.
- [3] 2017. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2008* (2017). DOI: <http://dx.doi.org/10.1109/IEEEESTD.2008.4579760>
- [4] 2017. NTP: The Network Time Protocol. (2017). <http://www.ntp.org>.
- [5] 2017. Symmetricom Announces General Availability of Industry's First Commercially-available Chip Scale Atomic Clock. (2017). <http://bit.ly/2ctnwjB>.
- [6] 2017. Z1. (2017). <http://zolertia.io/z1>.
- [7] Yin Chen, Qiang Wang, M Chang, and A Terzis. 2011. Ultra-low power time synchronization using passive radio receivers. In *The 10th International Conference on Information Processing in Sensor Networks (IPSN)*.
- [8] Jeremy Elson, Lewis Girod, and Deborah Estrin. 2002. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review* 36, SI (2002), 147–163.
- [9] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.
- [10] Niklas Fechner and Matthias Kirchner. 2014. The humming hum: Background noise as a carrier of ENF artifacts in mobile device audio recordings. In *The 8th International Conference on IT Security Incident Management and IT Forensics*.
- [11] World Economic Forum. 2015. Industrial Internet of Things: Unleashing the Potential of Connected Products and Services. (2015). <http://bit.ly/185DE8E>.
- [12] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava. 2003. Timing-sync protocol for sensor networks. In *The 1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*.
- [13] Ravi Garg, Avinash L Varna, and Min Wu. 2011. Seeing ENF: natural time stamp for digital video via optical sensing and signal processing. In *The 19th ACM International Conference on Multimedia (MM)*.
- [14] Catalin Grigoras. 2005. Digital audio recording analysis—the electric network frequency criterion. *International Journal of Speed Language and the Law* 12, 1 (2005), 63–76.
- [15] Jayant Gupchup, Răzvan Musăloiu-e, Alex Szalay, and Andreas Terzis. 2009. Sundial: Using sunlight to reconstruct global timestamps. In *European Conference on Wireless Sensor Networks*. Springer, 183–198.
- [16] T. Hao, R. Zhou, G. Xing, and M. Mutka. 2011. WizSync: Exploiting Wi-Fi Infrastructure for Clock Synchronization in Wireless Sensor Networks. In *The 32nd IEEE Real-Time Systems Symposium (RTSS)*.
- [17] Mareca Hatler, Darryl Gurganious, and Charlie Chi. 2013. *Smart Building Wireless Sensor Networks – A Market Dynamics Report*. Technical Report. ON World Inc. <http://onworld.com/smbldgs/>.
- [18] Liqun Li, Guoliang Xing, Limin Sun, Wei Huangfu, Ruogu Zhou, and Hongsong Zhu. 2011. Exploiting FM radio data system for adaptive clock calibration in sensor networks. In *The 9th International Conference on Mobile Systems, Applications, and Services (MobiSys)*.
- [19] Zhenjiang Li, Wenwei Chen, Cheng Li, Mo Li, Xiang-Yang Li, and Yunhao Liu. 2012. Flight: Clock calibration using fluorescent lighting. In *The 18th Annual International Conference on Mobile Computing and Networking (MobiCom)*.
- [20] Martin Lukac, Paul Davis, Robert Clayton, and Deborah Estrin. 2009. Recovering temporal integrity with data driven time synchronization. In *The 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*.
- [21] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. 2004. The flooding time synchronization protocol. In *The 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys)*.
- [22] T. Mizrahi. 2017. Security Requirements of Time Protocols in Packet Switched Networks. (2017). <https://tools.ietf.org/html/rfc7384>.
- [23] Anthony Rowe, Vikram Gupta, and Raganathan (Raj) Rajkumar. 2009. Low-power clock synchronization using electromagnetic energy radiating from AC power lines. In *The 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*.
- [24] Hui Su, Adi Hajj-Ahmad, Min Wu, and Douglas W Oard. 2014. Exploring the use of ENF for multimedia synchronization. In *International Conference on Acoustics, Speed and Signal Processing (ICASSP)*.
- [25] Sreejaya Viswanathan, Rui Tan, and David K. Y. Yau. 2016. Exploiting Power Grid for Accurate and Secure Clock Synchronization in Industrial IoT. In *The 37th IEEE Real-Time Systems Symposium (RTSS)*.
- [26] Allen J Wood and Bruce F Wollenberg. 2012. *Power generation, operation, and control*. John Wiley & Sons.