

DESIGN AND DEPLOYMENT OF LOW-COST WIRELESS SENSOR NETWORKS FOR  
REAL-TIME EVENT DETECTION AND MONITORING

By

Dennis Edward Phillips

A DISSERTATION

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

Computer Science – Doctor of Philosophy

2018

## ABSTRACT

### DESIGN AND DEPLOYMENT OF LOW-COST WIRELESS SENSOR NETWORKS FOR REAL-TIME EVENT DETECTION AND MONITORING

By

Dennis Edward Phillips

As sensor network technologies become more mature, they are increasingly being applied to a wide variety of environmental monitoring applications, ranging from agricultural sensing to habitat monitoring, oceanic and volcanic monitoring. In this dissertation two wireless sensor networks (WSNs) are presented. One for monitoring residential power usage and another for producing an image of a volcano's internal structure.

The two WSNs presented address several common challenges facing modern sensor networks. The first is in-network processing and assigning the processing tasks across a heterogeneous network architecture. By efficiently utilizing in-network processing power consumption can be reduced and operational lifetime of the network can be extended. As nodes are embedded into various environments sensing accuracy is intrinsically affected by physical noise. The second challenge relates to how to deal with this noise in a way which increases sensing accuracy. The third challenge is ease of deployment. As WSNs become more common place they will be installed by non-experts.

As a key technology of home area networks in smart grids, fine-grained power usage monitoring may help conserve electricity. Smart homes outfitted with network connected appliances will provide this capability in the future. Until smart appliances have wide adaption there is a serious gap in capabilities. To fill this gap an easy to deploy monitoring system is needed. Several existing systems achieve the goal of fine-grained power monitoring by exploiting appliances' power usage signatures utilizing labor-intensive *in situ* training processes. Recent work shows that autonomous power usage monitoring can be achieved by supplementing a smart meter with distributed sensors that detect the working states of appliances. However, sensors must be carefully installed for each appliance, resulting in high installation cost. *Supero* is the first *ad hoc* sensor system that can

monitor appliance power usage without supervised training. By exploiting multi-sensor fusion and unsupervised machine learning algorithms, Supero can classify the appliance events of interest and autonomously associate measured power usage with the respective appliances. Extensive evaluation in five real homes shows that Supero can estimate the energy consumption with errors less than 7.5%. Moreover, non-professional users can quickly deploy Supero with considerable flexibility.

There are a number of active volcanos around the world with large population areas located nearby. An eruption poses a significant threat to the adjacent population. During times of increased activity being able to obtain a real-time images of the interior would allow seismologists to better understand volcanic dynamics. Volcano tomography can provide this valuable information concerning the internal structure of a volcano. The second sensor network presented in this dissertation is a seismic monitoring sensor network featuring in-network processing of the seismic signals with the capability to perform volcano tomography in real-time. The design challenges, analysis of processing/network processing times in the information processing pipeline, the system designed to meet these challenges and the results from deploying a prototype network on two volcanoes in Ecuador and Chile are presented. The study shows that it is possible to achieve in-network seismic event detection and real-time tomography using a sensor network that is 2 orders of magnitude less expensive than traditional seismic equipment.

## ACKNOWLEDGEMENTS

As with any research effort, it can not be conducted in isolation. There are two groups within Engineering at Michigan State University who deserve special recognition. The first is the Electrical and Computer Engineering Technical Services group. This group fabricated many prototype seismic sensor circuit boards as I revised the final design. The second is the Engineering Machine Shop with the Mechanical Engineering Department. Using the equipment available within the machine shop I was able to fabricate the various mounting brackets for the sensor modules. In addition, the machinists were able to cut the odd shaped connector mounting holes in the sensor enclosures. Without the efforts of these two groups the construction of the seismic sensors would not have been achieved.

I would also like to thank the Department of Computer Science and Engineering for their support. Without the assistantships, fellowships, and general support I would not have been able to achieve my goal of earning my PhD.

Sincerely, Dennis E. Phillips

## TABLE OF CONTENTS

LIST OF TABLES . . . . .	vii
LIST OF FIGURES . . . . .	viii
CHAPTER 1 INTRODUCTION . . . . .	1
CHAPTER 2 RELATED WORK . . . . .	7
CHAPTER 3 UNSUPERVISED RESIDENTIAL POWER USAGE MONITORING USING A WIRELESS SENSOR NETWORK . . . . .	11
3.1 Overview of Supero . . . . .	12
3.1.1 Design Objectives and Challenges . . . . .	12
3.1.2 Motivating Observations . . . . .	13
3.1.3 System Architecture . . . . .	14
3.2 Event Detection and Data Correlation . . . . .	15
3.2.1 Light Event Detection . . . . .	15
3.2.2 Acoustic Event Detection . . . . .	16
3.2.3 Power Event Detection . . . . .	18
3.2.4 Multi-modal Data Correlation . . . . .	18
3.3 Event Classification and Appliance Association . . . . .	18
3.4 Duty-Cycled Heating Appliances . . . . .	21
3.5 Implementation and Deployment . . . . .	22
3.5.1 Prototype System Implementation . . . . .	22
3.5.2 System Deployment and Configuration . . . . .	23
3.6 Experimental Evaluation . . . . .	25
3.6.1 Deployments and Evaluation Methodology . . . . .	25
3.6.2 Controlled Experiments in Apartment-1 . . . . .	26
3.6.2.1 Experimental Settings . . . . .	26
3.6.2.2 Energy Estimation Accuracy . . . . .	30
3.6.2.3 Impact of Distance Errors . . . . .	31
3.6.3 10-Day Experiment in Apartment-1 . . . . .	31
3.6.3.1 Experiences and Learned Lessons . . . . .	31
3.6.3.2 Evaluation Results . . . . .	33
3.6.4 Experiments in Apartment-2 . . . . .	34
3.6.5 Experiments in House-1 . . . . .	38
3.6.6 System Usability . . . . .	38
3.6.7 System Lifetime . . . . .	39
3.7 Conclusion and Future Work . . . . .	40
CHAPTER 4 A SENSOR NETWORK FOR REAL-TIME VOLCANO IMAGING . . . . .	41
4.1 Background of Volcano Tomography . . . . .	41
4.1.1 Scientific Motivation . . . . .	41

4.1.2	Tomographic Processing Pipeline . . . . .	42
4.1.3	Spatial Coverage . . . . .	44
4.2	Design Requirements . . . . .	45
4.3	System Architecture and Design . . . . .	47
4.3.1	Signal Processing and Event Detection . . . . .	48
4.3.2	Sensor Node Design . . . . .	50
4.3.3	Design Lessons Learned and Generation 4 . . . . .	54
4.3.4	Base Station Design . . . . .	57
4.3.5	Packaging . . . . .	57
4.4	System Modeling & Dynamic Task Assignment . . . . .	58
4.4.1	System Delay Modeling . . . . .	59
4.4.2	System Lifetime Modeling . . . . .	60
4.4.3	Dynamic Task Assignment . . . . .	61
4.5	Deployments . . . . .	64
4.5.1	Tungurahua Volcano Deployment . . . . .	64
4.5.2	Llaima Volcano Deployment . . . . .	65
4.5.3	Deployment Lessons Learned . . . . .	67
4.6	Evaluation and Deployment Experiences . . . . .	67
4.6.1	System Delay . . . . .	68
4.6.2	Data Fidelity . . . . .	69
4.6.3	Communication Performance . . . . .	71
4.6.4	Battery and Solar Panel Performance . . . . .	73
4.6.5	Packaging and Ease of Deployment . . . . .	75
4.7	Conclusion . . . . .	75
CHAPTER 5 CONCLUSION . . . . .		76
BIBLIOGRAPHY . . . . .		81

## LIST OF TABLES

Table 3.1: Energy breakdown for the 1-hour controlled experiment in Apartment-1. . . . .	29
Table 3.2: Energy breakdown during 7 days in Apartment-1* . . . . .	34
Table 3.3: The set of sensors detecting a light (i.e., $R_m$ ) and clustering/association results . . . . .	36
Table 3.4: Energy breakdown in House-1* . . . . .	37
Table 4.1: Key Characteristics of a Traditional Seismic Station versus a our sensor nodes . . . . .	50
Table 4.2: Application processing deadlines . . . . .	59
Table 4.3: Task execution times by tier . . . . .	59
Table 4.4: Key characteristics of the deployments . . . . .	64
Table 4.5: Shared SPI bus ADC Sample Intervals . . . . .	70

## LIST OF FIGURES

Figure 1.1: Tungurahua Volcano near Baños - Ecuador Deployment . . . . .	5
Figure 3.1: The Supero architecture. . . . .	15
Figure 3.2: EDF result on light readings sampled at 4 Hz. Vertical lines represent detections. A person passes by Light 1 at the 31 <sup>st</sup> and 53 <sup>rd</sup> seconds. . . . .	16
Figure 3.3: Acoustic signal is separated into three bands using lattice wave digital filters for feature extraction. . . . .	17
Figure 3.4: Light feature vectors of two sensors. . . . .	19
Figure 3.5: Light intensity vs. distance (cm) in log-scale. . . . .	19
Figure 3.6: Acoustic event clustering and transition detection for a 3-speed fan. (a) The number of phases is identified as three; (b) Clustering and transition detection results, where <i>Y</i> -axis is the major principle component (PC) and vertical lines represent the detected acoustic transitions. . . . .	19
Figure 3.7: Detecting stove burner. (1) Red curve: Total household power readings when a burner is working; Blue curve: The reconstructed lower envelope. (2) Standard deviation of power readings and threshold-based detection results (detection window size: 100 s). . . . .	21
Figure 3.8: Web configuration interface. . . . .	25
Figure 3.9: Apartment-1 deployment. . . . .	27
Figure 3.10: Results of the controlled experiment in Apartment-1. (1) The top chart shows the power readings labeled with ground truths of the events. (2) The bars in the second chart show the detections of the light sensors. Two black bars at around the 35 <sup>th</sup> minute are false alarms (labeled “FA” in the chart) identified by the multi-modal data correlation. Clusters are differentiated by colors and the overhead numbers are the IDs of the associated light. (3) The third chart shows the major principle component given by PCA and the detected acoustic transitions. The acoustic transitions of the same color are associated with the same appliance. (4) The bottom chart shows the clustered and associated power events of the unattended appliances. . . . .	28
Figure 3.11: PRR and power traces in 10 days. . . . .	32

Figure 3.12: Sensor placements in Apartment-2. The numbers in the squares and circles are the sensor IDs of TelosB and Iris, respectively. If a TelosB does not face upward, the arrow represents its facing direction. . . . .	35
Figure 3.13: Sensor installation examples. Sensors were placed on the ground, in the corner of walls, on the fan of a range, and on a table. . . . .	36
Figure 3.14: Battery voltage traces of TelosB and Iris. . . . .	40
Figure 4.1: Seismic tomography and real-time signal processing pipeline. The tomography estimates a velocity model consisting of the seismic wave propagation speeds in cubic blocks beneath the volcano surface. . . . .	42
Figure 4.2: Tomography results using simulated seismic data. . . . .	44
Figure 4.3: General system architecture and online remote monitoring panel. . . . .	47
Figure 4.4: Data processing components of sensor node and base station. . . . .	48
Figure 4.5: STA/LTA ratio in response to a seismic signal. . . . .	49
Figure 4.6: Seismic Node Prototypes . . . . .	51
Figure 4.7: Seismic Monitoring Nodes . . . . .	53
Figure 4.8: Recorded Ecuador Seismic Event . . . . .	54
Figure 4.9: The clock drift and correction by GPS 1PPS . . . . .	54
Figure 4.10: Llaima Volcano deployment, Chile, 2015. . . . .	65
Figure 4.11: Node locations in Llaima Volcano deployment. . . . .	66
Figure 4.12: Hypocenter execution times by number of stations. . . . .	68
Figure 4.13: Tomography execution times by number of events. . . . .	68
Figure 4.14: One-hop link quality (circles represent nodes). . . . .	71
Figure 4.15: Line of Sight Path between Node 9 and Node 3 . . . . .	72
Figure 4.16: Distribution of Satellite Round Trip Ping Times . . . . .	72
Figure 4.17: Battery daily charging cycle. . . . .	74

Figure 4.18: System Life Time Estimation . . . . . 74

# CHAPTER 1

## INTRODUCTION

Since their introduction in the late 1990's the use of wireless sensor networks (WSNs) to perform monitoring has continually grown. Initially used to monitor and record various environmental readings (temperature, humidity, etc) their use has broadened to include detection, tracking, and identification of objects and events. As the complexity of the events or objects to be detected and identified increases so must the sophistication of the techniques and algorithms. With the increase in event detection complexity, there has also been an increase in the type and complexity of features used to in the detection process. This has gone from simple environmental features (light levels, temperature, humidity, etc.) to multiple features (from both single and multiple sensors) to time sequences of features.

Along with the changes in complexity, the very nature of WSNs has changed. WSNs have moved from simple monitoring to providing information and data which is to be acted upon in realtime. This results in new challenges which must be addressed when designing a WSN. Some of these challenges are:

**Ease of deployment** - As WSNs have matured who deploys the nodes has changed. Initial WSN were installed by researchers who were also the designers. Today, WSNs are being installed by ordinary homeowners, as is the case for residential power monitoring applications, or individuals who are not the designers. This requires the individual sensor nodes to be easy to install and insensitive to where they are placed.

**Sensing accuracy** - One of the purposes is to provide actionable information as a result accuracy is important. False alarms and false positives can have a significant effect possibly resulting in the wrong or inappropriate action to be taken. Steps must be taken to reduce these to application acceptable limits.

**Heterogeneous network architecture** - The increase in computation capabilities also allows WSN designers to include nodes with different computational capabilities. This results in an opportunity

for assigning the computational task to the appropriate element that minimizes overall system power consumption while still meeting any application processing deadlines. The assignments may need to be dynamic and adapt to changes in the sensor data or communication delays.

**Operational lifetime** - One challenge which hasn't changed is power related. In many applications the sensor nodes are still battery powered. While the computing capabilities have increased significantly, battery power density has not increased at the same pace. Yet, these WSNs are expected to reliably operate for extended periods of time with little to no maintenance. As the computational capabilities of the individual nodes have increased, this allows more processing to be performed within the network. Rather than sending raw sensor data over the WSN communication channels, the network now can analyze the data, extract the relevant features, and only transmit this reduce set of features. This in-network processing can be used to reduce power and extend the operational lifetime of the WSN.

This dissertation presents two different applications of wireless sensor networks illustrating these challenges and approaches to address them. The first is a residential power monitoring application, *Supero*. Since 1978 the percentage of residential electricity has increased from 17% to 31% [49], while the cost of energy has also been on the rise. A recent study shows that in 2001 one half of the American homes with an annual income of less than 50,000 U.S. dollars annually spent, on average, 12% of the after tax income on energy [2]. Now in 2011 this percentage has increased to 21% [2]. With these increasing energy costs, home owners are increasingly interested in reducing the energy usage of their appliances. If they have a better understanding of the energy consumption of their appliances, they could more easily identify any wastage of power.

As a key technology of home area networks in smart grids, fine-grained power usage monitoring may help conserve electricity. Several existing systems achieve this goal by exploiting appliances' power usage signatures identified in labor-intensive *in situ* training processes. Recent work shows that autonomous power usage monitoring can be achieved by supplementing a smart meter with distributed sensors that detect the working states of appliances. However, sensors must be carefully installed for each appliance, resulting in high installation cost. Chapter 3 presents *Supero* – the

first *ad hoc* sensor system that can monitor appliance power usage without supervised training. By exploiting multi-sensor fusion and unsupervised machine learning algorithms, Supero can classify the appliance events of interest and autonomously associate measured power usage with the respective appliances.

Previous systems for fine-grained power usage monitoring can be broadly classified into two categories. The first category, *direct sensing*, measures per-appliance power usage by smart plugs [18] and smart switches [17]. As smart plugs are placed between the appliances and power outlets, they cannot be used for appliances hardwired to power lines, such as ceiling lights. Replacing normal wall switches with smart switches needs cumbersome hardwiring and possibly expensive modifications to walls. In light of the installation overhead, direct sensing is suitable only when permanent monitoring is desired. However, for identifying power wastage and diagnosing inefficient appliances, a swift one-off deployment for a short time period (e.g., a few weeks) is typically sufficient. The second category, *indirect sensing*, is less intrusive as it infers the working states and energy consumption of individual appliances by detecting their power usage patterns [15, 35] or ambient signals they emit during operation [14, 22]. However, these techniques require either labor-intensive *in situ* supervised training, due to their dependency on the appliance characteristics [15] and electrical wiring [35, 14], or careful sensor installation for each appliance [22], leading to high installation cost and reduced usability.

Supero aims to design a residential power usage monitoring system that (i) uses only inexpensive and easy-to-install sensing devices, (ii) can be deployed by non-professional users with straightforward instructions, and yet (iii) can work effectively based on a small amount of easily obtained prior information without resorting to supervised *in situ* training. Such a system must automatically detect events of interest, autonomously associate the events with the correct appliances, and finally infer the power usage of each appliance. It brings three key challenges. First, inexpensive sensors typically have limited sensing capabilities; hence, they can produce false alarms or miss important events of monitored appliances. Second, when sensors are installed in an *ad hoc* manner, multiple sensors may detect the same event, and it becomes difficult to associate the event with the appliance

that is the source of the event. Lastly, to make the system practical, we must minimize the amount of prior information that users will need to collect.

Another area which can benefit from the use of WSNs is seismic monitoring. It is estimated that approximately 500 million people live within the risk range of volcanos [7]. Unfortunately, the dynamics of volcanos is not fully understood. Volcano tomography provides valuable information concerning the internal structure of a volcano. Volcano tomography allows a seismologist to visualize the internal geological structure of a volcano [24, 25, 13]. Using the travel times from seismic event sources to multiple monitoring stations it is possible to model the propagation speed of the seismic waves as they travel through different portions of the structure. The speed is affected both by the density of the material as well as its temperature. When a volcano transitions from a dormant state to a crisis state, as indicated by a marked increase in seismic activity, the fear is that an eruption could occur. If the seismologists could observe the internal structural changes in real-time they would have a better idea if an eruption is imminent and nearby populations centers should be evacuated.

Current volcano monitoring systems developed by the seismic community cannot achieve real-time tomography due to the small number of stations deployed around active volcanos. Most existing volcano monitoring systems employ expensive broadband seismic stations. Portable traditional stations cost \$20,000US as well as being bulky and difficult to deploy. Consequently, even the most active volcanos today are monitored by fewer than 20 nodes [42] which is insufficient to perform real-time tomography. The stations typically sample seismic data at a sample rate of 50-200Hz for off-line, batch processing. With such a small number of nodes and batch processing mechanism, existing systems do not have the capability to capture the physical dynamics of volcanoes with sufficient detail. This limits our ability to study volcanic activities and the internal physical dynamics of volcanoes. Therefore, an online seismic monitoring system could lead to substantial scientific discoveries on the geology and physics of active volcanoes and similar applications. This requires a large-scale network with real-time sensing and online in-network processing capabilities.

There have been many low-power sensor systems developed [51, 42], but mainly focus on data



Figure 1.1: Tungurahua Volcano near Baños - Ecuador Deployment

collection and didn't address the challenge of the data intensive computations needed to perform real-time volcano tomography. With the high sampling rate required for seismic sensing, it is virtually impossible to collect and transmit live data from a large-scale dense sensor network, due to severe limitations of energy and network bandwidth with current, battery-powered sensor nodes.

Designing a sensor platform for real-time volcano tomography faces several challenges. System hardware and software needs to be carefully designed in tandem to ensure sufficient compute power is available while being power-efficient, minimizing network bandwidth utilized and staying within cost budget. Due to the significant data intensity, the system likely consists of heterogeneous platforms processing the information at multiple levels, such as sensor nodes, local cluster head, and remote server. The complex trade-offs between multiple system performance metrics such as delay, energy consumption, and sensing accuracy must be carefully analyzed.

A seismic sensor network capable of performing volcano tomography was developed. This system performs real-time seismic event detection and identifies the seismic wavefront arrival times at sensor nodes. To achieve desirable event detection accuracy while keeping computation

overhead low, we develop a new seismic event detection framework that extends and integrates two earthquake detection algorithms and a data fusion scheme. Moreover, to leverage the system heterogeneity, we design a novel compute task assignment scheme based on analytical models of system delay and lifetime. Processing tasks can be assigned to the right system tiers such that all real-time and energy requirements are met. Through two field deployments of the system on Tungurahua Volcano, Ecuador and Llaima Volcano, Chile, in 2012 and 2015, respectively, it is shown that the system can survive the harsh environmental conditions found in volcanic regions.

The system design and deployment experience provides important insight into developing sensor networks in other application domains that require sophisticated, high-fidelity processing of intensive, highly dynamic and complex physical information with the network. In summary, the main contributions include 1) design and implement a new system for seismic monitoring consisting of a low cost sensor node (< \$500US) with mesh data network communication, base station supporting data intensive computations and satellite link communication, and remote server performing real-time network monitoring and tomography computations. 2) conduct an empirical delay analysis of the tomographic processing pipeline and propose a task assignment scheme specifically targeted at computationally intensive real-time applications, 3) describe the experiences of two field deployments to volcanic regions, and 4) systematically evaluate our system using results from these deployments.

## CHAPTER 2

### RELATED WORK

Research as it relates to the WSN applications presented in this dissertation center around three areas. The first area is the overall approach to sensing, i.e. direct or indirect sensing. For example, the temperature of an object could be measured by directly attaching a temperature probe or indirectly from a distance using an infrared sensor. This is also the case when trying to determine appliance power usage. A second area of research involved sensor systems which are very data-intensive. In the past, WSNs samples sensors at rates ranging from one reading per second to several minutes between readings. Data-intensive systems sample sensors at much higher rates ranging from 100 Hz to several kilohertz. The third area of research relates to assign computation tasks to different hardware elements based on computation capabilities, power consumption and communication delays.

The following discusses representative indirect sensing approaches for appliance power usage monitoring, and identifies their differences from Supero. Early work in this area [15, 8, 10] utilizes per-appliance power operating characteristics, measured at power panels, to disaggregate the total energy consumption. These approaches need either *in situ* training [15, 10] or a comprehensive database of *a priori* power characteristics of appliances [15, 8]. Jiang et al. [19] present the experience of monitoring the power usage of a laboratory using smart plugs [18] and light sensors. In [20], binary sensors are used to help deploy power meters to estimate energy breakdowns in a building. Both of the studies [19, 20] exploit the tree topology of the subject power supply system. Patel et al. [35] detect and classify electrical events based on transient noises generated by the appliances. Their transient signatures are heavily influenced by the electrical wiring, which results in the need of *in situ* training. In [14, 45], appliances are recognized based on their electromagnetic interference and acoustic signals. Similarly, their work requires labor-intensive *in situ* training. A typical training process involves switching on/off appliances, and collecting and labeling signals. Recently, Ho et al. [16] use a thermal camera to detect the on/off states of appliances and infer the

per-appliance energy consumption. The thermal camera can be hard to install and can raise privacy concerns in residential environments.

ViridiScope [22] is a fine-grained power usage monitoring system closest in design to Supero. It features an autonomous regression framework that can calculate per-appliance energy consumption based on the appliances' working states and the total household power trace. ViridiScope detects the working states by carefully installed sensors. For instance, light and magnetic sensors are placed in close proximity to or attached to each appliance, and must not be triggered by other appliances to ensure correct power estimation. Such an installation of the sensors is hard for difficult-to-access appliances such as ceiling lights. In Supero, due to the use of unsupervised learning and novel sensor fusion/association techniques, sensors are not dedicated to specific appliances, and so can be deployed in an *ad hoc* manner, leading to significantly lower installation costs. ViridiScope uses two acoustic sensors to monitor a fridge compressor and reject ambient noises. In Chapter 3, we propose a systematic approach for monitoring a range of acoustic appliances, which jointly processes the data from all acoustic sensors to detect the appliances' working states.

The task partitioning presented in ORBIT [29] dispatches the execution of sensing and processing tasks in a smart-phone-based multi-tier architecture to achieve *data-intensive* applications requirements. Signal processing timing profiles can exhibit significantly variations in real scenarios. To address this, ORBIT measures the statistical timing profiles at runtime, and periodically refines the partitioning results. ORBIT maximizes the battery lifetime subject to the application-specific latency constraints. Moreover, in order to support fine-grained task partitioning across the tiers, the developer specifies the application's task structure as well as real-time requirements via either Java annotations or an XML-based application model provided by ORBIT. ORBIT also provides a messaging interface to support unified data passing mechanism between heterogenous tiers and between different application components.

There have been a number of wireless sensor network (WSN) deployments over the past decade. In the following, this focuses on discussing sensor systems that target data-intensive applications, including those monitoring industrial equipment[23], structural health [33, 52, 21, 44], earthquakes

[39, 11] and volcanos [51, 42].

Krishnamurthy, et. al. [23] installed a WSN in a semiconductor plant and an operating oil tanker to collect data used to predict equipment failure. Each node sampled multiple vibration sensor at 19.2. kHz. Data was collected at regular intervals for processing and analysis by a backend system. Only DC component removal was performed in-network.

Xu, et. al. [52] and Paek, et. al. [33] developed WSNs for structural monitoring using Mica motes interfaced to a vibration sensor card sampled at either 100 or 200 Hz. In both cases the focus was primarily on reliable network communication. Kim, et. al. [21] installed a 64-node network on the Golden Gate Bridge to perform structural health monitoring utilizing vibration sensors were sampled at 1kHz. All analysis was performed off-line. Stoianov, et. al. [44] used a WSN to monitor the health of underground pipeline in Boston, MA. The nodes in this system computed pressure sensor reading summary statistics and transmitted these statistics to a gateway.

Rosi, et. al. [39] deployed a 16 node landslide monitoring WSN over a 500 square meter area. The Micaz nodes communicated to a laptop used to relay information via cellular network to a remote server for storage and display of the sensor readings. A data compression algorithm based on the motion state was used to reduce network bandwidth. No other in-network processing was performed. Faulkner, et. al. [11] used a community based sensor network to detect rare seismic events. Cell phones and motion sensors attached to PC were used to detect seismic events sending an event message to a cloud fusion center.

Werner-Allen, et. al. [51] and Song, et. al. [42] deployed small (< 20 node) WSNs on volcanos. Both systems employed event detection on the nodes and transmitted the data to a base station for storage and offline processing. Werner-Allen's stations did not sample and store sensor readings during the data communication process resulting in lost events and data. Song's stations were large and needed to be transported to the deployment site using a helicopter. Unlike their systems which focused on network communication and data collection, our system targets in-network seismic signal processing and real-time compute-intensive tomography.

Various task offloading schemes for smartphones have been developed recently. Spectra [12]

allows programmers to specify task partitioning plans given application-specific service requirements. Chroma [3] aims to reduce the burden on manually defining the detailed partitioning plans. Medusa [38] features a distributed runtime system to coordinate the execution of tasks between smartphones and cloud. Turducken [43] adopts a hierarchical power management architecture, in which a laptop can offload lightweight tasks to tethered PDAs and sensors. While Turducken provides a tiered hardware architecture for partitioning, it relies on the application developer to design a partitioned application across the tiers to achieve energy efficiency.

The MAUI system [5] enables a fine-grained offloading mechanism to prolong the smartphone's battery lifetime. However, MAUI relies on the properties of the Microsoft .NET managed code environment to identify the functions that can be executed remotely. When a function is executed remotely, MAUI assumes the energy associated with its local execution is saved. In contrast, ORBIT does not rely on any language specific environment and its measurement-based power profiles account for many realistic power characteristics such as CPU sleep, wake up and tail time.

The Wishbone system [31] also features a task dispatch scheme. Wishbone uses the CPU and network timing profiles only to find the optimal task partition. Moreover, Wishbone depends on the timing profiles based on sample data under the assumption that the sample data can represent actual runtime data. However, signal processing timing profiles can exhibit significantly variations in real scenarios due to the variations in data complexity. Moreover, Wishbone formulates the partitioning problem as a 0/1 integer programming problem and thus supports two tiers only.

## CHAPTER 3

### UNSUPERVISED RESIDENTIAL POWER USAGE MONITORING USING A WIRELESS SENSOR NETWORK

This chapter presents the design and implementation of *Supero* – a system for unsupervised power monitoring.

To detect appliance operating modes *Supero* utilizes an indirect sensing approach. With this approach sensors are not directly attached to each appliance. This also allows each sensor to potentially monitor more than one appliance at a time. Using some a priori information provided by a home owner *Supero* is able to estimate the energy consumption with errors less than 7.5%

*Supero* utilizes a smart meter to measure real-time total household power consumption and inexpensive light and acoustic sensors that are deployed in an *ad hoc* manner to detect interesting events of appliances. It uses multi-sensor fusion to correlate data collected by power, light, and acoustic sensors and reduce possible sensing errors. By using advanced unsupervised clustering algorithms, *Supero* analyzes the signal signatures of different appliances and identifies the events generated by the same appliance. Moreover, *Supero* autonomously associates the classified events with the appliances through an optimization framework that accounts for environmental factors such as light signal propagation. Given a small amount of easily obtained prior information such as sensor-appliance distances and rated powers of a small subset of the appliances, our unsupervised algorithms work together to disaggregate the total household energy consumption into usage by the individual appliances.

To the best of our knowledge, *Supero* is the first practical *ad hoc* sensor system that can accurately monitor appliance power usage without supervised training. *Supero* aims at swift one-off deployments for power usage diagnosis over short time periods (e.g., a few days to weeks). As such, there should be little concern about user privacy or any negative visual impact of the sensor installation.

*Supero* was prototyped using a network of TelosB/Iris motes [28, 27] and a smart meter, and

evaluated it in five real homes of different sizes and with different characteristics of electricity consumption. A 10-day evaluation in an apartment shows that Supero can estimate the energy consumption with errors less than 7.5%. The results also demonstrate that Supero can be quickly deployed by non-professionals with considerable flexibility.

The remainder of this chapter is organized as follows. Section 3.1 presents the overview of Supero. Section 3.2 presents the event detection and multi-sensor fusion algorithms. Section 3.3 presents the unsupervised event clustering and autonomous appliance association approaches. Section 3.4 discusses estimating the power consumption of a class of high-power heating appliances. Section 3.5 and Section 3.6 present our system implementation and evaluation results, respectively. Section 3.7 concludes this chapter.

## **3.1 Overview of Supero**

### **3.1.1 Design Objectives and Challenges**

The goal of Supero is to produce fine-grained electricity usage reports over specific time durations in a household. A report includes the energy consumption of particular appliances, as well as when they were turned on/off. Supero is designed to meet the following three objectives. First, it should be possible to deploy the sensors in an *ad hoc* and *non-intrusive* manner. A non-professional should be able to deploy battery-powered wireless sensors with intuitive instructions such as “place a light sensor with unobstructed view to the light” and “place an acoustic sensor on top of the microwave.” Second, we aim to reduce needed efforts for system configuration by avoiding the use of labor-intensive training and extensive user inputs. Third, Supero should be able to operator for a long enough time period (e.g., a few weeks) without changing the sensors’ batteries, such that the generated report is meaningful and informative enough for identifying wasteful energy usage and diagnosing efficiency problems in appliances.

Four major challenges are brought by the above design objectives. First, in an *ad hoc* deployment, a sensor may pick up signals emitted by multiple appliances, which can make it difficult to pinpoint the appliance that is consuming power. For instance, a light sensor can sense light emitted

by various sources, and an acoustic sensor in the kitchen can hear sounds from the exhaust fan, disposer, microwave, etc. Second, without careful installation, sensors typically suffer from sensing errors caused by ambient noises and human activities. For instance, light sensors can report false alarms when nearby window blinds are opened, and acoustic sensors may pick up sounds such as human conversations that are unrelated to power consumption. Third, without *in situ* system training, unsupervised learning often requires more prior information than supervised learning. In Supero, we strive to reduce the burden on users to obtain the prior information required, while maintaining good monitoring accuracy. Finally, to extend the system lifetime, wireless sensors should adopt lightweight sensing algorithms and minimize the data transmissions, which however raises challenges for accurate monitoring of appliance working states.

### 3.1.2 Motivating Observations

To meet the aforementioned objectives, Supero utilizes a household power meter and a small number of inexpensive light and acoustic sensors that are deployed in the home in an *ad hoc* manner. Based on an unsupervised approach, it does not require any *in situ* system training. Rather, it leverages a small amount of prior information that can be easily obtained by non-professional users. We now discuss several important observations that motivate our approach.

**Real-time total household power metering** Nowadays, the real-time total household power consumption can be easily measured by installing a commercial off-the-shelf smart meter (e.g., TED [6] and AlertMe [1]) on the main circuit panel. These meters are inexpensive and most of them can be easily installed without hardwiring with the power lines. Moreover, as the coverage of smart grid increases, the real-time total household power readings are increasingly available to the homeowners, without resorting to a personal smart meter.

**Sensing modalities** According to a survey of U.S. Department of Energy [48], the average distribution of electricity consumption in household is: heating 24%, lights 24%, air conditioners 20%, refrigerators 15%, dryers 9%, and electronics 9%. As most heating appliances con-

sume substantially more power than other appliances, their consumption trace often can be identified from the real-time total household power readings. Most lights, air conditioners, refrigerators and dryers emit light and acoustic signals. As a result, on average, more than 90% power consumption of a typical household can be captured by a combination of a smart meter and a set of light and acoustic sensors.

**Useful prior information** To avoid expensive *in situ* system training, Supero leverages unsupervised learning techniques and a small amount of prior knowledge including rough sensor-appliance distances and the rated powers of a small subset of appliances. As the light/acoustic signal decays with the distance from the source appliance, the distances between sensors and appliances provide important hints for associating the detected events to the right appliances. Moreover, although the rated power of an appliance often has small discrepancy with the actual power consumption, it helps identifying the consumption trace of a small number of difficult-to-detect appliances from the household power readings. Rated powers are often available from the labels on the appliances or the user manuals. Moreover, there exist a few publicly available databases (e.g., [46]), which provide rated power based on the appliance brand and model.

### 3.1.3 System Architecture

Supero consists of a number of wireless sensors distributed in the home being monitored, a smart meter, and a base station for receiving information from the sensors and the smart meter. In this work, we only consider light and acoustic sensors while other sensing modalities such as infrared can be easily incorporated by Supero.

Fig. 3.1 illustrates the *two-tiered* architecture of Supero. In the first tier, sensors sample signals and detect events that are possibly caused by switching appliances on/off. On the detection of an event, a sensor extracts various features of the event and sends an event message to the base station. Further details of the first tier will be presented in Section 3.2. The base station provides a graphic configuration interface that allows user to input prior information such as sensor-appliance

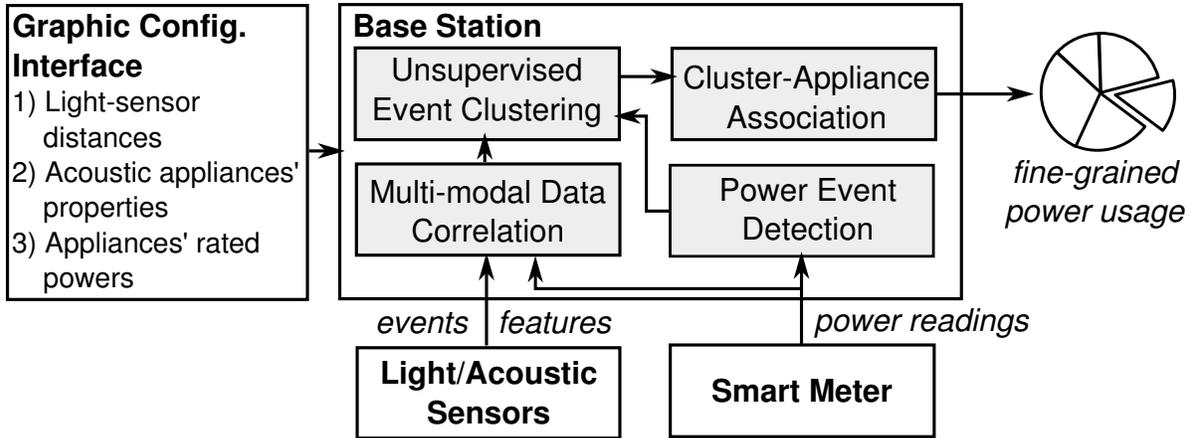


Figure 3.1: The Supero architecture.

distances and appliances' rated powers. When Supero is requested to generate a power usage report, the base station executes the following second-tier algorithms based on the collected data and the prior information input by the user:

**Multi-modal data correlation** The base station correlates sensor events and power readings to differentiate between true appliance events and false alarms unrelated to power consumption. (Section 3.2.4)

**Unsupervised event clustering and event-appliance association** Leveraging unsupervised clustering algorithms, we can classify the events generated by an appliance into the same cluster, and estimate the power consumption of the appliance by correlating the events with measurements by the smart meter. Supero associates the classified events with their appliances based on features of the events and the prior information. It then calculates the energy consumption of each appliance. (Section 3.3)

## 3.2 Event Detection and Data Correlation

### 3.2.1 Light Event Detection

Light sensors detect the state changes of lights from changes in the light readings. We adopt an *exponential difference filter* (EDF) to light intensity samples to detect light events. The EDF is

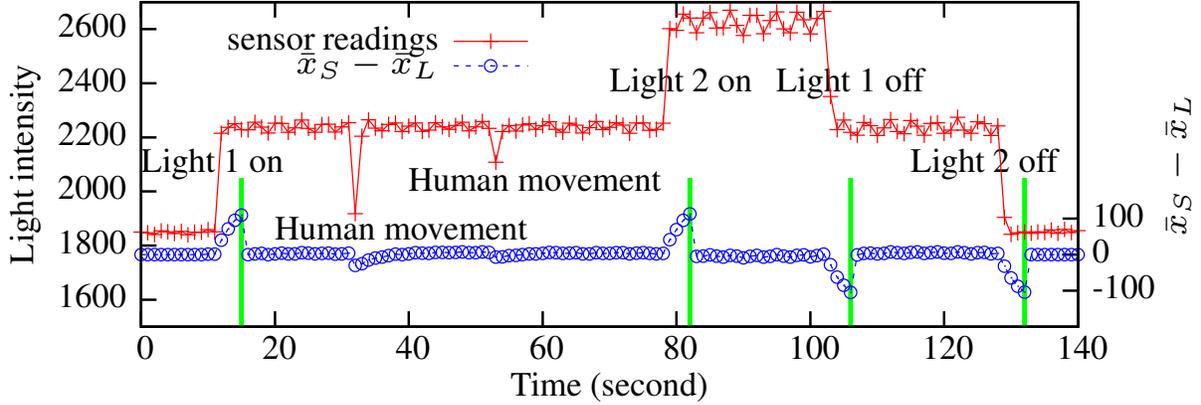


Figure 3.2: EDF result on light readings sampled at 4 Hz. Vertical lines represent detections. A person passes by Light 1 at the 31<sup>st</sup> and 53<sup>rd</sup> seconds.

lightweight and resilient to sensing noise and natural ambient light changes. Specifically, using two settings for the coefficient of the exponential moving average (EMA), the sensor computes the *short-term* and *long-term* EMAs, denoted by  $\bar{x}_S$  and  $\bar{x}_L$ , respectively, over the periodic light samples (4 Hz in our implementation). Note that a historical light reading has higher weight in  $\bar{x}_L$  than in  $\bar{x}_S$ . If  $|\bar{x}_S - \bar{x}_L|$  keeps higher than a threshold for a certain number of readings, the sensor reports a *light event* message which includes the current reading as well as the two averages. Moreover, it sets  $\bar{x}_L = \bar{x}_S$  to adapt  $\bar{x}_L$  quickly to the most recent readings. The coefficients and thresholds used in EDF are carefully tuned in offline experiments such that the EDF is resilient to normal human movements. Fig. 3.2 shows the operation of the EDF on the sensor readings when two lights are turned on/off and a person moves around. It can be seen that the light events can be accurately detected and the human movements do not trigger false alarms. Light sensors may still pick up events unrelated to power consumption (which we refer to as *non-power events*), such as those caused by human movements and the opening/closing of window blinds, which will be identified by a multi-modal data correlation technique given in Section 3.2.4 and then discarded.

### 3.2.2 Acoustic Event Detection

A challenge in acoustic sensing is that a high sampling rate is often required to extract event features. Supero adopts a duty-cycled and adaptive sampling scheme to reduce the energy consumed in the

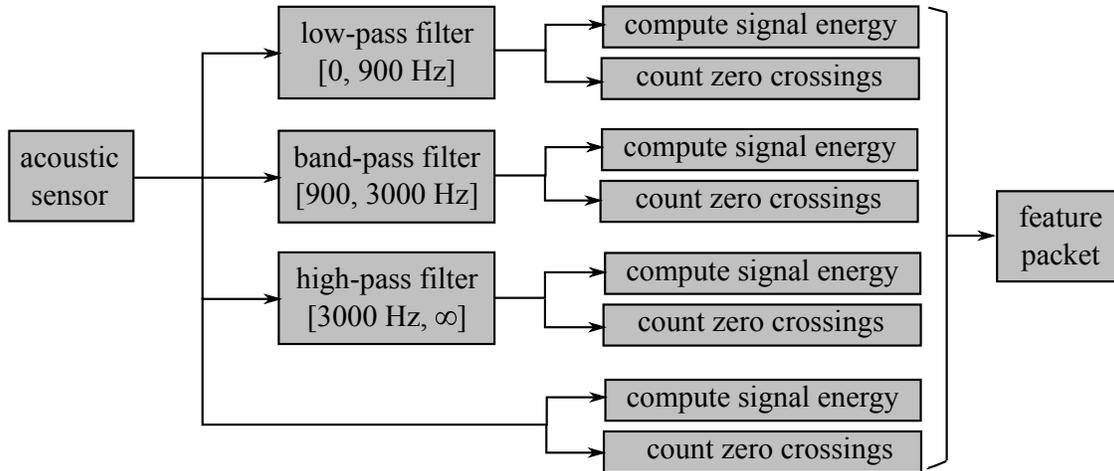


Figure 3.3: Acoustic signal is separated into three bands using lattice wave digital filters for feature extraction.

sampling and computation. For each second, an acoustic sensor is active for 0.08 seconds only. Initially, it samples the signal at 1 kHz when it is active. If the signal energy exceeds a threshold  $\eta_A$ , the sensor switches to a high sampling rate of 12.5 kHz to capture more details of the potential event. As shown in Fig. 3.3, we use three lattice wave digital software filters to decompose the signal into low-pass, band-pass, and high-pass components. The passbands of the three filters are  $[0, 900 \text{ Hz}]$ ,  $[900 \text{ Hz}, 3000 \text{ Hz}]$ , and  $[3000 \text{ Hz}, \infty)$ , respectively. The signal energy and zero-crossing counts of the signals in the whole band and the three subbands are computed as acoustic features and transmitted to the base station. The sensor remains in the fast sampling mode as long as the signal energy is above  $\eta_A$ . We set a low threshold  $\eta_A$  conservatively such that the acoustic sensors will not miss any sounds generated by an appliance. Note that different from a *light event*, that refers to the switching on/off of a light, an *acoustic event* refers to the sound heard by a sensor. Therefore, the sensor will continuously report acoustic events while the sound persists. We refer to the switching or phase change of an acoustic appliance as an *acoustic transition*. Owing to intrinsic complexity of the acoustic modality, acoustic transitions are detected by advanced learning algorithms running on the base station, as we will discuss in Section 3.3.

### 3.2.3 Power Event Detection

As the total power consumption is critical for identifying appliance events and estimating per-appliance consumption, real-time power readings by the smart meter are transmitted to the base station for storage. Moreover, the base station applies EDF to detect rapid increases and drops in the power measured. The thresholds in the EDF are tuned in offline experiments such that power changes as small as 50 W can be always detected. In this analysis, we assume that the appliances are not duty-cycled at high rates, except those explicitly specified. In Section 3.4, we develop an approach for monitoring high-power duty-cycled appliances (e.g., stove burner) and discuss how to integrate the approach with Supero.

### 3.2.4 Multi-modal Data Correlation

Because of their limited sensing capability and the complexity of home environments, the sensors can easily raise false alarms or miss important on/off events of appliances. For instance, opening/closing a window blind can trigger the nearby light sensors, and human conversations may trigger the acoustic sensors. To deal with these sensing errors, we present a two-tiered fusion approach to correlate the light/power events and acoustic transitions reported by different sensors. The first tier uses a short moving window to correlate the events/transitions reported by multiple sensors of the same modality. The events/transitions falling into the same window are regarded as generated by the same source. This is equivalent to an OR-rule for decision fusion and can greatly reduce the overall miss rate. The second tier correlates the results of the first tier with readings by the smart meter to remove false alarms. Specifically, if the change in power on an event/transition is smaller than a conservatively low threshold (e.g., 5 W), the event/transition will be discarded. The evaluation in Section 3.6 shows that this approach is effective in removing sensor false alarms.

## 3.3 Event Classification and Appliance Association

A novel feature of Supero is that it automatically classifies the detected events and associates them with the right appliances, without any *in situ* system training. Supero uses a two step approach

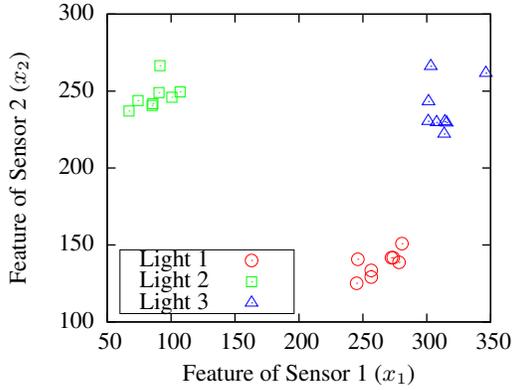


Figure 3.4: Light feature vectors of two sensors.

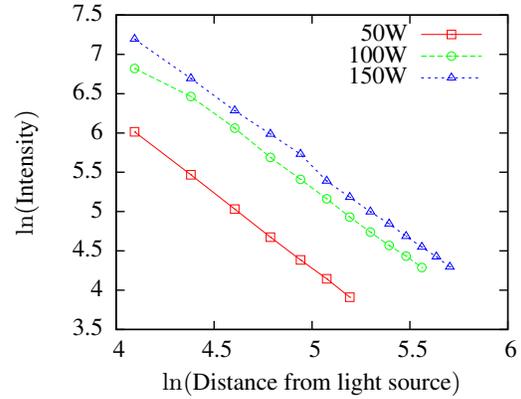


Figure 3.5: Light intensity vs. distance (cm) in log-scale.

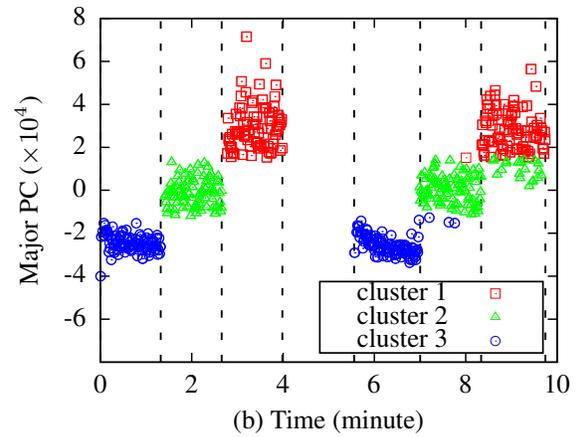
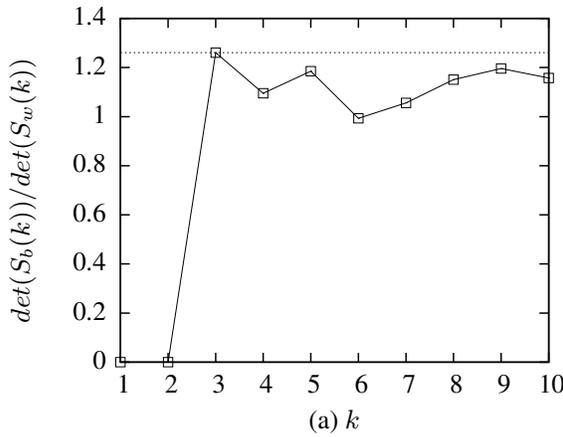


Figure 3.6: Acoustic event clustering and transition detection for a 3-speed fan. (a) The number of phases is identified as three; (b) Clustering and transition detection results, where Y-axis is the major principle component (PC) and vertical lines represent the detected acoustic transitions.

to this:

**Event Clustering** Events are clustered using the features extracted by the sensor nodes. For lights, light intensity as measured by each light sensor is used. Fig. 3.4 shows the feature vectors measured by two light sensors when three standing lights nearby the sensors were turned on and off. We can clearly see that the feature vectors are clustered together.

For appliances, the overall sound intensity and the intensity in three specific frequency bands are used. To reduce then number of acoustic features, principal component analysis is used and those features contributing to 99% of the variance are used. A challenge when clustering

features is determining the number of clusters to use. For light events, we know the number of monitored lights. This is used as the number of clusters which should be given to the k-means clustering algorithm. For acoustic events this is more difficult. Some appliances have multiple modes with each mode consuming a different amount of power such as a three speed floor fan. In this case each acoustic cluster may represent a different mode or appliance. The number of actually used modes of an appliance depends on the habit of the user and is therefore unpredictable. Thus, it is not known the number of clusters that will be present. Supero estimates the number of clusters by comparing the between-cluster and within-cluster variance matrices for varying number of clusters. As a simple example, Fig. 3.6 shows a case study using one acoustic sensor only to detect the phase changes of a 3-speed fan. As shown in Fig. 3.6(a), optimum number is identified as 3 based on the acoustic event features shown in Fig. 3.6(b). Transitions between modes can be identified as transitions between clusters over time as shown by the vertical lines in Fig. 3.6(b). In either case, the mean power change associated with each event in a given cluster is used as the power consumption for that device or mode.

**Appliance Association** To associate, light events with the actual two pieces of information are use: a) the prior information of how far a specific light is from each light sensor and b) the decay of light intensity follows the power law as shown in Fig. 3.5. Using this the association algorithm is described in detail in [36] is applied to associate a light with each light event cluster. For associating appliances with acoustic events, the change in sound intensity is used. The assumption is made that the sensor registering the greatest change is the sensor closest to the appliance generating the acoustic event. A heuristic association algorithm described in [36] is used. There may be appliances that generate power events but do not generate light or acoustic events. These are referred to as unattended appliances. The approach to handling unattended appliances is also described in [36].

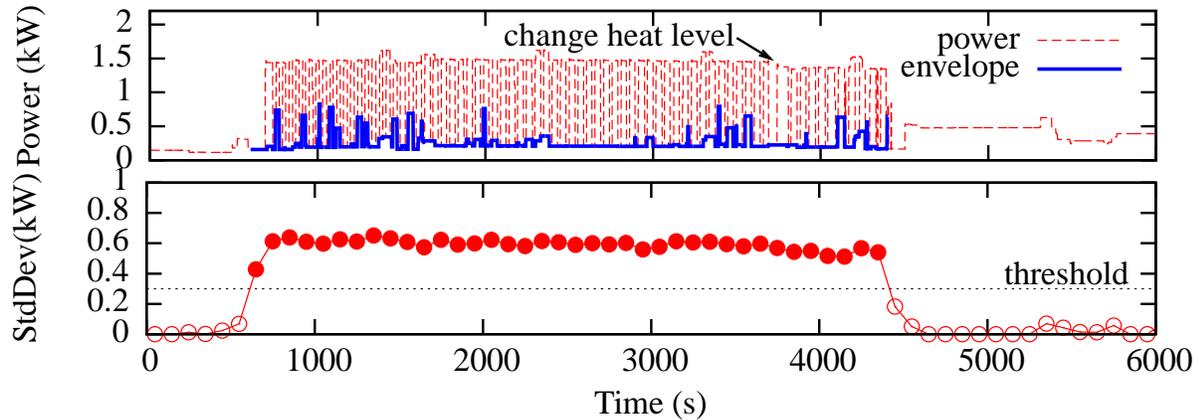


Figure 3.7: Detecting stove burner. (1) Red curve: Total household power readings when a burner is working; Blue curve: The reconstructed lower envelope. (2) Standard deviation of power readings and threshold-based detection results (detection window size: 100 s).

### 3.4 Duty-Cycled Heating Appliances

As discussed in Section 3.1.2, heating appliances such as stove burner and oven are major electricity consumer in homes. Most modern heating appliances duty-cycle to achieve the desired heat level. For instance, the top part of Fig. 3.7 shows the total household power readings when a GE JB710ST2SS burner is working. As the cycle can be short (e.g., several seconds), the EDF-based detector discussed in Section 3.2.3 may have poor performance. In this section, we propose a new approach to detect the duty-cycling pattern from the total power readings and calculate the related energy consumption.

Duty-cycled appliance rapidly switches between on and off, causing large variation in power readings. Thus, we detect the duty-cycling pattern based on the standard deviation of the windowed power readings. By denoting  $P$  and  $\gamma \in (0, 1)$  as the power and duty cycle of the appliance, the standard deviation of the power readings can be derived as  $P\sqrt{\gamma - \gamma^2}$ . We choose a threshold of  $P\sqrt{0.05 - 0.05^2}$  by conservatively assuming that the duty cycle is greater than 5%. When  $P$  is unknown, we can choose a default value of 1.5 kW for  $P$  because most duty-cycled heating appliances have a rated power around 1.5 kW [46]. As a result, the default threshold is 0.327 kW. To suppress the false alarms caused by other high-power non-duty-cycled appliances, we further

require that the zero crossing count of the mean-removed power readings in a window is at least 2. The bottom part of Fig. 3.7 shows the standard deviation of the power readings in the top part and the detection result. We can see that the time duration that the burner is working can be accurately detected. For the power readings in a window that has a positive detection, we apply the  $k$ -means algorithm with  $k = 2$  and then interpolate the power readings in the cluster with a smaller average to reconstruct the lower envelope of power consumption (i.e., the background power), as shown in the top part of Fig. 3.7. With the lower envelope, it is easy to calculate the energy consumption of the duty-cycled appliance. In typical U.S. homes, stove burner and oven are the major duty-cycled heating appliances and they are often the components of a range. Supero does not differentiate the duty-cycled heating appliances and attributes all energy consumption to the range. To address multiple simultaneously working duty-cycled appliances, the number of clusters, i.e.,  $k$ , can be first determined by the technique presented in Section 3.3.

The rapid duty-cycling can cause significant errors to the EDF-based power event detection (cf. Section 3.2.3) and the second tier of the multi-modal data correlation (cf. Section 3.2.4). Hence, when a duty-cycled appliance is detected, Supero disables these two components and the power changes of the light/acoustic events in this period are set to be missing values. Although such a design can cause errors to other appliances, it is worthwhile to give priority to the high-power duty-cycled appliances since they usually dominate the total power consumption of a household.

## 3.5 Implementation and Deployment

### 3.5.1 Prototype System Implementation

**Sensors and smart meter.** The sensors are implemented using TelosB [28] and Iris [27] motes. TelosB only has light sensor while Iris has both light and acoustic sensors. According to our lab tests, the light sensors on TelosB and Iris have satisfactory isotropic sensitivity in a considerably large range of incoming angles, which can mitigate the impact of sensor orientation on the accuracy of the power-law-based association algorithm. The signal sampling and event detection algorithms described in Section 3.2 are implemented in TinyOS 2.1. The parameters used in these algorithms

are carefully tuned offline and then fixed for different deployments. The sensors communicate directly with the base station. Such a single-hop topology suffices for our deployments in three apartments and two multi-story houses. TED5000 [6] is used to measure the total household power consumption.

**Base station.** The base station is a TelosB mote connected to a laptop computer. A daemon service on the computer retrieves real-time power readings from the TED5000 and stores the received event messages. The data correlation, clustering, and association algorithms are implemented in GNU Octave. The energy consumption of an appliance is computed by integrating estimated power over time. Note that this simple energy calculation method can be easily replaced by the regression-based method developed in [22] to improve robustness.

**Groundtruth Kill-A-Watt meters.** In order to evaluate the accuracy of Supero, we built 14 power meters based on the P3 Kill-A-Watt (KAW) Model P4400 [32] to provide groundtruth power usage data of individual appliances. We connect two ADC channels of a Senshoc mote to two pins on the internal circuit board of a KAW to sample the voltage and current signals. Senshoc is a TelosB-compatible mote implementation with significantly reduced cost [26]. The Senshoc mote computes and transmits the real-time power usage data to the base station for storage. Each modified KAW is carefully calibrated to output accurate power readings.

### 3.5.2 System Deployment and Configuration

This section discusses the sensor deployment and initial configuration of Supero.<sup>1</sup>

**Sensor deployment strategies.** A necessary condition for correct clustering and association is that every light/acoustic appliance can be detected, which is referred to as the *coverage requirement*. A conservative deployment strategy is to place a sensor close to each appliance. The number of sensors can be reduced by incrementally placing sensors close to appliances, starting with those that emit dim light/acoustic signals, until the coverage requirement is met. In our implementation,

---

<sup>1</sup>An online video illustrating the system deployment and configuration can be found at <https://youtu.be/4sSZaaV0Kv4>

the coverage is checked by switching appliances on and check the sensors' LEDs that blink to indicate detection. Note that this coverage check is different from supervised training processes (e.g., [35]) that are typically conducted after system deployment and involve labelling the events with the source appliances. After coverage requirement is met, a few extra sensors may be deployed in regions without any sensors to provide redundancy and improve robustness. The effectiveness of the above conservative and incremental deployment strategies will be evaluated in Section 3.6.4.

**User inputs.** First, Supero needs a list of the monitored appliances, which are categorized as lights, acoustic, or unattended appliances. Supero also needs to know whether an appliance has multiple working states although the exact number of the working states is optional. Second, for the light modality, Supero requires roughly estimated line-of-sight distances between the sensors and lights. Third, for the acoustic modality, Supero needs to know whether an acoustic appliance has a primary sensor or not. All the non-primarily monitored acoustic appliances need to be sorted by their powers. Such a ranking is usually straightforward to obtain, e.g., based on common sense. Finally, Supero requires the rated powers of the unattended appliances, which can be obtained from the labels on the appliances or from a database of appliance rated powers. Supero only needs to be reconfigured occasionally, e.g., when sensors/appliances are relocated.

**Configuration interface.** We have developed a web configuration interface using JavaServer Pages served by the base station computer to help the user input all the required information. For instance, Fig. 3.8(a) shows the configuration for the acoustic sensing, where the user can input the acoustic sensor IDs, appliance names, and other required information. In addition, we leverage TPCDB [46], which is an online collaboratively edited database of appliance powers, to help the user input the required rated powers. Currently, TPCDB comprises the information of more than 500 appliances. Fig. 3.8(b) shows our interface of querying TPCDB through its web service API, where the user can find the rated power by appliance type, manufacturer and model. The case studies presented in Section 3.6.6 shows that this interface can be easily used by non-professionals.

**Part 2: Acoustic Sensing**

**(a) Acoustic sensors**

ID	Op.
11	<a href="#">Del</a>
12	<a href="#">Del</a>
	<a href="#">Del all</a>

New acoustic sensor ID

**(b) Acoustic appliances**

Name	Primary sensor	Multiphase	Power order	Op.
bath fan	None ▾	No ▾	1 ▾	<a href="#">Del</a>
hair dryer	None ▾	No ▾	2 ▾	<a href="#">Del</a>
tower fan	11 ▾	Yes ▾	N/A ▾	<a href="#">Del</a>
				<a href="#">Del all</a>

New acoustic appliance name

(a) Acoustic configuration

**Appliance Power Database**  
(Data source: [www.tpcdb.com](http://www.tpcdb.com))

Type  Manufacturer

Model	Power
VE276Q	52.17 W
MS246H	33.0 W
24T1E	75.0 W
MS238H	33.0 W
MT276H	70.0 W
22T1E	75.0 W

**Other online databases:**

- [California Energy Commission: AED](#)
- [Energy Star](#)
- [Natural Resources Canada: enerGuide](#)

(b) Rated power database

Figure 3.8: Web configuration interface.

### 3.6 Experimental Evaluation

#### 3.6.1 Deployments and Evaluation Methodology

We deployed and evaluated Supero in five real households. We first deployed Supero in a 40 m<sup>2</sup> single-bedroom apartment (Apartment-1). As most of the appliances in Apartment-1 can be monitored by groundtruth KAW meters, this deployment allows us to extensively evaluate the accuracy of Supero. We then evaluate the sensor deployment strategies (cf. Section 3.5.2) in an 80 m<sup>2</sup> apartment (Apartment-2). In addition, we deployed Supero in a one-story three-bedroom ranch house (House-1) to evaluate the portability of Supero to larger homes. Lastly, we recruited two homeowner volunteers to deploy Supero in their homes, an apartment (Apartment-3) and a two-story house (House-2). The Apartment-3 and House-2 deployments evaluate if non-professionals can deploy Supero easily.

We compare Supero with two baseline approaches. The first baseline approach (referred to as *Oracle*) uses appliances' groundtruth states and then applies the regression-based energy calculation

method in ViridiScope [22]. In the second baseline approach (referred to as *Baseline*), the state of each appliance is detected by the sensor closest to the appliance and then the regression is applied. The results of *Baseline* will help us understand the challenges brought by an *ad hoc* sensor deployment.

## **3.6.2 Controlled Experiments in Apartment-1**

### **3.6.2.1 Experimental Settings**

The electrical appliances in Apartment-1 include 5 standing lights, a fridge, a water boiler, a 3-speed tower fan, a rice cooker, a bath fan, a hair dryer, 3 laptop computers, and a WiFi router. The apartment uses a natural gas range and a steam-based central heating unit that do not draw electrical power. The deployment consists of 4 TelosB and 5 Iris motes. The Iris motes only detect acoustic events. The laptops and router cannot be easily detected by sensors. However, as the router's rated power is known and it is always on, Supero can estimate its energy consumption. The residual energy consumption is thus mainly attributed to the laptops. The rice cooker, water boiler, and fridge are treated as unattended appliances, because they do not emit light or stable acoustic signals. The water boiler and fridge are also monitored by acoustic sensors. Fig. 3.9 shows the floor plan and sensor positions. The sensors are placed on the floor, a nearby table, chairs, and a toilet. The positions of the sensors are not carefully chosen except for the tower fan, fridge, and water boiler. Sensors are deployed close to these quiet appliances. As the bathroom has complex sound patterns, two acoustic sensors are deployed and both of them can hear all the appliances and the sound of water flow in the bathroom.

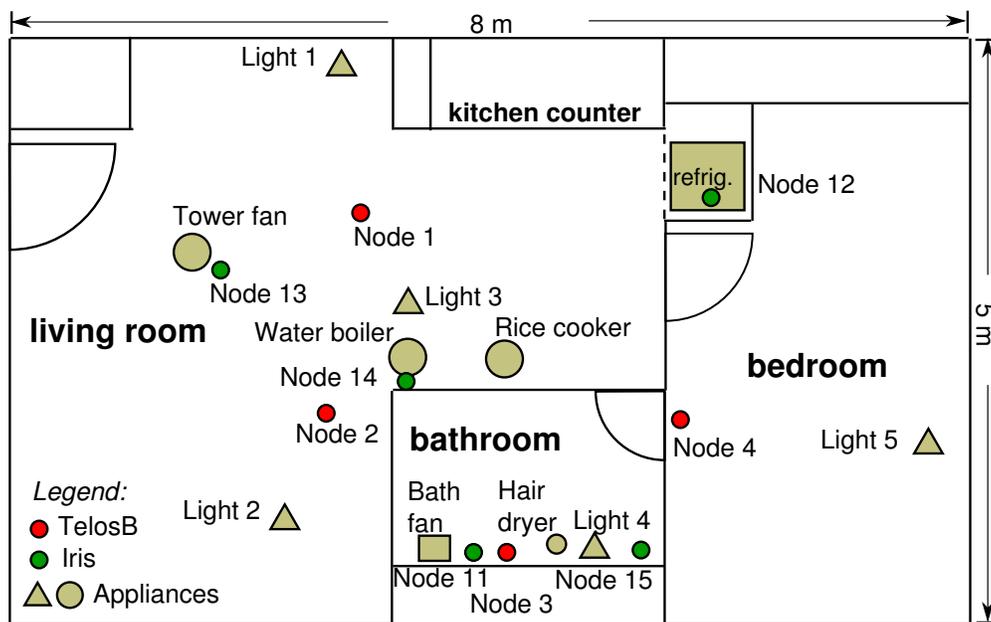


Figure 3.9: Apartment-1 deployment.

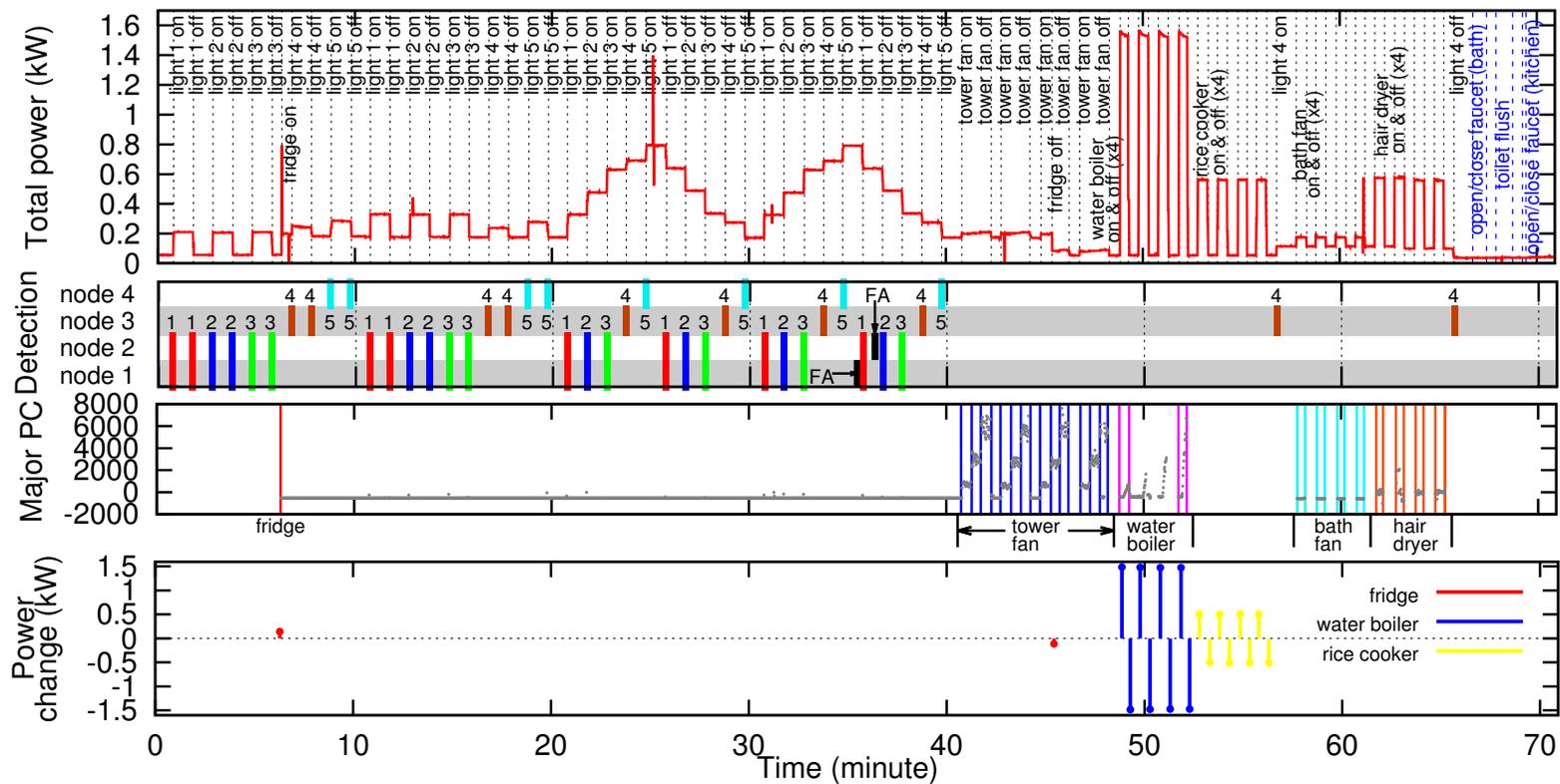


Figure 3.10: Results of the controlled experiment in Apartment-1. (1) The top chart shows the power readings labeled with ground truths of the events. (2) The bars in the second chart show the detections of the light sensors. Two black bars at around the 35<sup>th</sup> minute are false alarms (labeled “FA” in the chart) identified by the multi-modal data correlation. Clusters are differentiated by colors and the overhead numbers are the IDs of the associated light. (3) The third chart shows the major principle component given by PCA and the detected acoustic transitions. The acoustic transitions of the same color are associated with the same appliance. (4) The bottom chart shows the clustered and associated power events of the unattended appliances.

Table 3.1: Energy breakdown for the 1-hour controlled experiment in Apartment-1.

Appliance		KAW		Supero			Oracle			Baseline		
Name	Rating (W)	Power (W)	Energy (kW·h)	Power (W)	Energy (kW·h)	Error* (%)	Power (W)	Energy (kW·h)	Error* (%)	Power (W)	Energy (kW·h)	Error* (%)
Light 1	150	152	0.0307	154	0.0309	0.7	152	0.0305	0.7	153	0.0310	1.0
Light 2	150	148	0.0298	150	0.0300	0.7	150	0.0300	0.7	151	0.0305	2.3
Light 3	150	151	0.0300	153	0.0304	1.3	153	0.0306	2.0	152	0.0307	2.3
Light 4	50	60	0.0211	61	0.0210	0.5	60	0.0210	0.5	62	0.0219	3.8
Light 5	100	102	0.0207	103	0.0205	0.5	100	0.0200	3.4	102	0.0206	0.5
Water boiler	1500	1472-1524	0.0490	1479	0.0456	6.9	1481	0.0481	1.8	232	0.0289	41.0
Tower fan	N/A	23-40	0.0031	N/A	0.0029	5.3	{23, 28, 35}	0.0028	9.7	30	0.0045	45.1
Rice cooker	500	498	0.0163	508	0.0168	3.1	507	0.0168	3.1	508	0.0163	0.0
Hair dryer	N/A	442	0.0158	462	0.0150	5.1	459	0.0150	5.1	5	0.0018	88.6
Fridge	N/A <sup>†</sup>	117-146	0.0784	129	0.0841	7.3	122	0.0795	1.4	119	0.0848	8.2
Bath fan	N/A <sup>‡</sup>	N/A	N/A	60	0.0020	N/A	61	0.0020	N/A	55	0.0048	N/A
Router	12	12.5	0.0147	12	0.0142	3.4	13	0.0154	4.8	13	0.0154	4.8
3 Laptops	N/A	37-63	0.0468	36	0.0430	8.1	31	0.4840	3.4	53	0.0472	0.9
Average error				3.6			3.1			16.5		

\*Error is the relative error of energy, in percentage, with respect to the KAW measurements.

<sup>‡</sup>Bath fan is hardwired to the power line and hence no KAW is applied for it.

<sup>†</sup>Fridge's rated power is not available. However, its power events can be correctly associated when a rated power of 80 W to 400 W is given to Supero.

### 3.6.2.2 Energy Estimation Accuracy

This section presents the results of a *controlled* experiment, in which we intentionally turned on and off the appliances. It allows us to understand the micro-scale performance of Supero. Fig. 3.10 shows the groundtruth information, power readings, event detection and clustering results. Both of the two light false alarms are identified by the multi-modal event correlation. No light event is missed. All the light events are correctly clustered and associated. For the acoustic modality, the non-power sounds such as toilet flush and run of tap water can be identified by the multi-modal data correlation. From the third chart in Fig. 3.10, Supero fails to detect the off event of the fridge and four events of the water boiler. The miss detections of the water boiler are caused by the delay of sound. However, as discussed in Section 3.3, by jointly treating the fridge and water boiler as acoustic and unattended appliances, these misses can be successfully recovered by the events detected from the power readings. Other detected acoustic transitions including the phase changes of the 3-speed tower fan can be correctly associated.

Table 3.1 shows the groundtruth measurements by KAWs and the estimation results of the various approaches. Both Supero and *Oracle* can accurately estimate the power and energy of each appliance. The average errors of energy consumption estimate are lower than 4%. For a few appliances, Supero outperforms *Oracle*. This can be caused by small errors in the groundtruth measurements by KAWs and the adoption of different energy calculation methods in Supero and *Oracle*. As Lights 1, 2, and 3 have no nearby sensors, *Baseline* uses the groundtruth states of Lights 1, 2 and 3. For other appliances, *Baseline* uses the closest sensor to detect the state of an appliance. As *Baseline* does not perform data correlation and event clustering, it generates excessive false alarms. For instance, as the hair dryer is very noisy, all the acoustic sensors raise detections when the hair dryer is on, which causes false alarms for all the other acoustic appliances. Hence, *Baseline* yields wrong power and energy estimates for several appliances. In fact, it is quite difficult to deploy dedicated acoustic sensors as they can be easily triggered by any noisy appliances. Acoustic data from multiple sensors must be jointly processed to produce correct detections.

### 3.6.2.3 Impact of Distance Errors

This section evaluates the robustness of the association algorithm in Section 3.3 with respect to the errors in the light-sensor distances. The distances given to Supero are distorted as follows. First, we proportionally increase all the distances. As the association algorithm can find a best fit scaling factor  $\beta$ , the association remains correct even if we multiply the distances by 10. Second, we add a random bias to a particular distance in each test. The result shows that if the bias is within 70% of the true distance, the association remains correct. Finally, when we exclude Node 2 from the evaluation, the results remain the same as long as the order of the distances from Node 1 to Light 1 and Light 3 is consistent with reality, i.e., Light 1 is farther from Node 1 than Light 3. These results demonstrate that Supero is robust to the errors in the light-sensor distances.

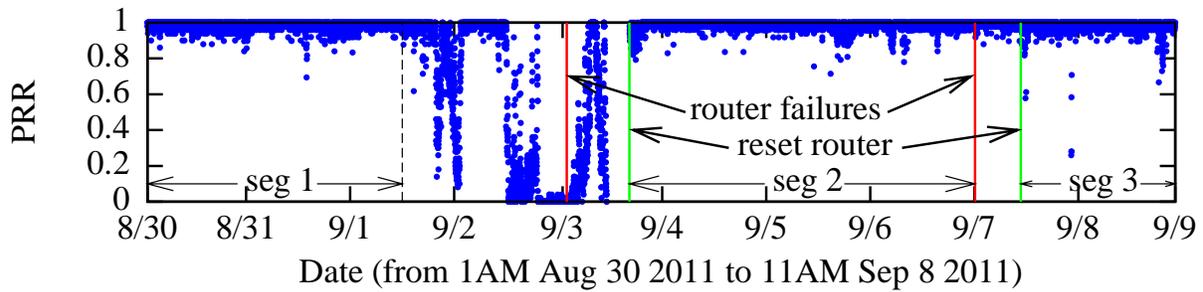
### 3.6.3 10-Day Experiment in Apartment-1

We conducted a 10-day *uncontrolled* experiment, during which two residents led normal lives in their apartment. In this section, we first discuss our experiences and learned lessons, and then present the evaluation results.

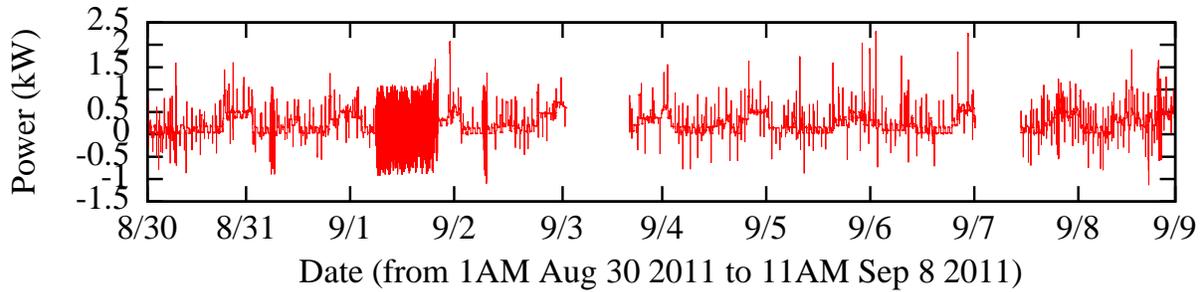
#### 3.6.3.1 Experiences and Learned Lessons

We experienced the following three issues during the 10-day experiment.

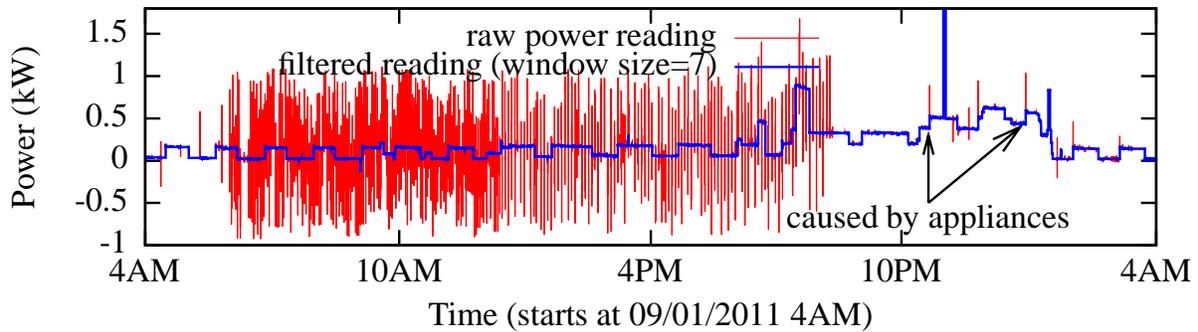
**Power spikes.** Power spike is a typical dynamics in power lines, which can be caused by bad weather conditions and turning on/off appliances in the tested home and even neighbor homes. Power spikes may cause errors in the appliance power estimation. In the controlled experiment, we can see a few power spikes in the top chart of Fig. 3.10 when an appliance changes state. As we apply a guard region for computing the power change as discussed in Section 3.2.4, the power spikes do not affect the results. However, in the 10-day experiment, we observe excessive power spikes as shown in Fig. 3.11(b) that can affect the calculation of power changes for the detected events. We suspect that the power spikes observed on September 1 were caused by the thunderstorms during the period



(a) PRR of a KAW



(b) Power trace



(c) 12 hours of power trace on September 1, 2011

Figure 3.11: PRR and power traces in 10 days.

of experiment. A zoomed-in view of the power trace on that day is shown in Fig. 3.11(c). Almost all power spikes can be removed by a median filter with a window size of 7 seconds. We also apply the median filter with the same setting to the power traces collected in other experiments.

**Router failures.** The probe of TED5000 installed on the power panel sends real-time readings through power lines to the TED5000 gateway, which was attached on a power outlet and wired to the WiFi router to deliver readings to the base station computer. However, the router failed twice during the 10 days, leading to disruptions to the collection of power readings. We had to reset the router manually to restart the data collection. We suspect that the failures were caused by bugs in

the router. As power readings are critical information to Supero, it is crucial to adopt a high-quality and stable router. Moreover, when the base station fails to receive power readings for a while, it can raise an alarm sound to remind the user to reset the router.

**Communication performance.** The quality of wireless links between the base station and sensors can affect the performance of Supero. Each Supero sensor only sends a packet when an event is detected while each KAW meter continuously transmits groundtruth power usage to the base station by the attached Senshoc mote equipped with a CC2420 radio. Therefore, we use the data traces of KAWs to examine the packet reception ratio (PRR). Fig. 3.11(a) shows the PRR of a KAW during the 10 days. We can see that the communication performance significantly degraded and fluctuated between the evening of September 1 and the noon of September 3. As the residents watched online videos over WiFi during this period, we suspect that the poor link quality was caused by the interference from WiFi. We also examined the traces of other KAWs. Similarly, their link quality degraded during this outage period. We were able to repeat this phenomenon in an extra experiment using Senshoc motes and two laptops that transferred a large file over WiFi. Although the channel of Senshoc was set to 11, which is well separated from channel 6 used by WiFi, the PRR of Senshoc still significantly degraded. However, we did not observe significant degradation of PRR when experimenting with TelosB and Iris motes. Hence, we suspect that the performance degradation is caused by the imperfect antenna design of Senshoc.

Nevertheless, after the 10-day experiment, we have enabled packet acknowledgment and added retransmission mechanism to enhance the reliability of communication. Due to the router failures and lost groundtruth information from KAWs, we only use three data segments (“seg 1”, “seg 2” and “seg 3” shown in Fig. 3.11(a)). The total length of the three segments is more than 7 days. The three data segments are concatenated and then fed to the clustering and association algorithms.

### **3.6.3.2 Evaluation Results**

Table 3.2 shows the results based on the data of 7 days. During this period, 713 false alarms out of a total of 859 light events were raised by the light sensors, in which 703 of the false alarms

Table 3.2: Energy breakdown during 7 days in Apartment-1\*

Appliance	KAW	Supero			Oracle			Baseline		
Name	$E$ (kW·h)	$P$ (W)	$E$ (kW·h)	Error (%)	$P$ (W)	$E$ (kW·h)	Error (%)	$P$ (W)	$E$ (kW·h)	Error (%)
Light 1	4.14	154	4.17	0.5	152	4.11	0.9	152	4.11	0.9
Light 2	4.96	150	4.96	0.1	149	4.92	0.8	149	4.92	0.8
Light 3	6.15	155	6.24	1.4	155	6.25	1.7	155	6.25	1.7
Light 4	1.45	62	1.45	0.1	62	1.45	0.1	63	1.48	1.7
Light 5	0.39	105	0.39	0.2	105	0.39	0.7	110	0.41	5.5
Water boiler	0.48	1493	0.48	0.5	1491	0.48	1.6	0	0	100
Tower fan	0.15	30	0.21	50	26	0.17	17.9	24	0.24	66.2
Rice cooker	1.00	499	0.98	2.2	513	1.01	1.2	511	1.01	0.8
Hair dryer	0.09	467	0.07	19.2	467	0.09	0.4	3	0.02	73.2
Fridge	12.22	143	11.8	3.7	127	11.8	3.2	127	11.8	3.2
Bath fan	N/A	50	0.12	N/A	57	0.17	N/A	0	0	N/A
Router	2.12	12	2.03	4.3	18	3.04	43.3	18	3.04	43.3
Average error				7.5			6.5			27.0

\*Error is relative error of energy with respect to KAW measurements.

are identified by the multi-modal data correlation. All the remaining false alarms are identified as outliers by the event clustering algorithm (cf. Section 3.3). In addition to the acoustic transitions generated by the fridge, 60 acoustic transitions were detected. We see that Supero can accurately estimate the energy consumption of lights. The tower fan was turned on and off twice and all its transitions were detected. However, two bath fan transitions were incorrectly associated with the tower fan, because Node 13 (i.e., the primary sensor for the tower fan) heard loud noises in the living room at the same time. The two false associations introduce errors in the energy estimates of the tower fan and hair dryer. As shown in Table 3.2, the average error of Supero is only 7.5%. The average error of *Oracle* is 6.5%. Therefore, the performance of Supero is close to that of *Oracle*. *Baseline* still fails to estimate the energy consumption of several appliances due to excessive false alarms, leading to an average error of 27%.

### 3.6.4 Experiments in Apartment-2

This section evaluates the performance of Supero under different sensor placements. We deployed 6 TelosB and 11 Iris motes in the doorway, living room, and kitchen of Apartment-2, as shown in Fig. 3.12. As the two doorway lights are in series, they are regarded as one light. As shown in Fig. 3.13, sensors were placed or attached on the ground, walls, appliances, and furniture. Note that

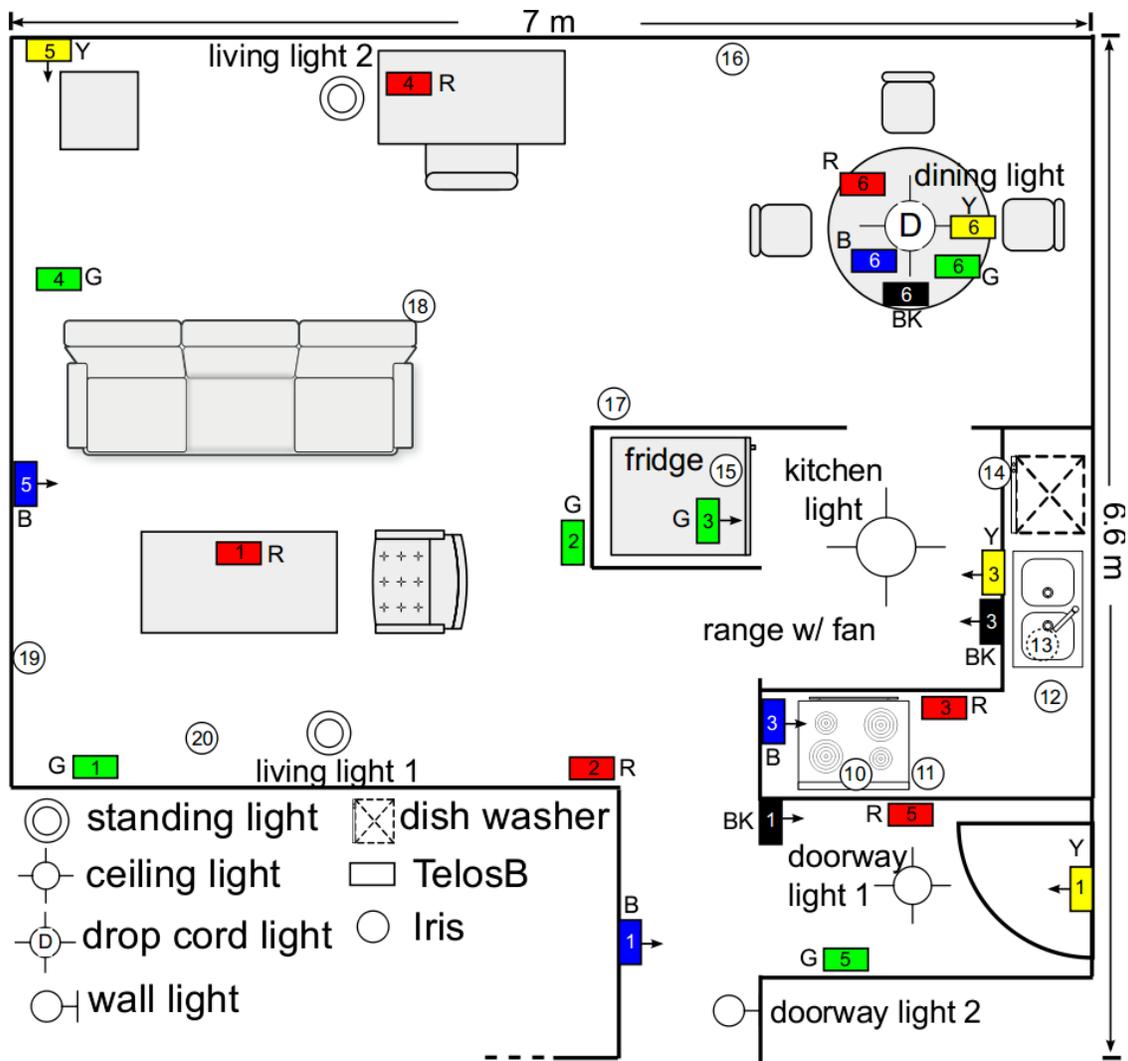


Figure 3.12: Sensor placements in Apartment-2. The numbers in the squares and circles are the sensor IDs of TelosB and Iris, respectively. If a TelosB does not face upward, the arrow represents its facing direction.

the positions of sensors were chosen by common sense without careful planning. We also varied the positions of sensors in several trials and similar results were observed, as shown later in this section. We first evaluate the light modality. We conducted five sensor placement trials to monitor 6 lights including incandescent bulbs and fluorescent lamps. Different colors of the TelosB notes in Fig. 3.12 represent different placements, which are also labeled with the initials of color names, i.e., ‘R’, ‘G’, ‘B’, ‘Y’ and ‘BK’. In the red and green placements, a sensor was placed close to each appliance. The blue and yellow placements follow the incremental strategy to reduce the number

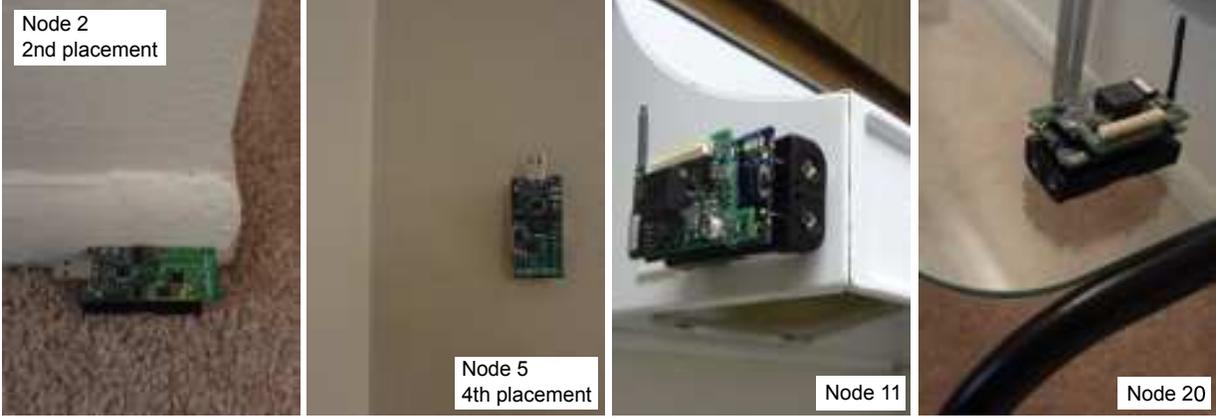


Figure 3.13: Sensor installation examples. Sensors were placed on the ground, in the corner of walls, on the fan of a range, and on a table.

Table 3.3: The set of sensors detecting a light (i.e.,  $R_m$ ) and clustering/association results

Light	Red	Green	Blue	Yellow	Black
Dining	{6}	{6}	{6}	{6}	{6}
Kitchen	{3}	{3}	{3}	{3}	{3}
Doorway	{5}	{5}	{1}	{1}	{1}
Living 1	{1,2,4}	{1,2,4}	{5}	{5,3}	{3}
Living 2	{1,2,4,6}	{1,2,4,6}	{5,6}	{5,6}	{6}
Result	✓	✓	✓	✓	X

of sensors from 6 to 4. In the black placement, no sensor was deployed in the living area. All the placements ensure the coverage requirement. We conducted a controlled experiment to evaluate each placement. Table 3.3 shows the set of sensors that can detect the same light. The clustering and association results of the red to yellow placements are correct. In the black placement, although all the events can be detected, they cannot be correctly clustered. For instance, although Node 6 can detect the near dining light (13 W) and the farther “living light 2” (150 W), the changes in light intensity from them are similar, leading to incorrect clustering.

To further demonstrate the flexibility of sensor deployment, we deployed 11 Iris motes and select four different subsets of them as sensor placements, which are  $S_1 = \{\text{All Iris motes}\}$ ,  $S_2 = \{10, 12, 14, 15, 16, 18, 20\}$ ,  $S_3 = \{10, 12, 14, 19\}$ , and  $S_4 = \{10, 14\}$ . All the subsets satisfy the coverage requirement. However, they represent very different deployment strategies.  $S_1$  and  $S_2$  use redundant sensors and hence are conservative.  $S_3$  follows the incremental deployment strategy.

Table 3.4: Energy breakdown in House-1\*

Appliance Name	Groundtruth		Supero		
	$P$ (W)	$E$ (kW·h)	$P$ (W)	$E$ (kW·h)	Error (%)
Entry light	32	.0079	33	.0081	2.3
Hall light	38	.0112	38	.0109	1.9
Kitchen light	24	.0059	23	.0056	5.8
Dining light	76	.0149	77	.0113	24.6
Living light	43	.0041	41	.0040	3.1
Master bed light	33	.0065	31	.0061	6.0
Master bath light	22	.0054	21	.0052	3.6
Master bath fan	47	.0069	47	.0068	2.3
Guest bed light	29	.0071	29	.0056	21.2
Guest bath light	20	.0070	20	.0070	0.6
Guest bath fan	41	.0097	40	.0097	0.0
Stove burner	1356	.4603	1379	.4675	1.6
Water dispenser	N/A	N/A	140	.0518	N/A
Average error					6.1

\*Error is relative error of energy with respect to KAW measurements.

As there is no sensor in the living area,  $S_4$  does not follow any proposed deployment strategy. The acoustic appliances covered in the experiment include an exhaust fan over the range, a waste disposer in the sink, a dish washer, and a vacuum cleaner. During the experiment, we used the vacuum cleaner in both the dining and living areas. The exhaust fan has two speeds and Node 10 is designated as the primary sensor for the fan. For the other appliances, the order (rather than the actual values) of their power consumption is provided to Supero. The event detection and association results for  $S_1$ ,  $S_2$ , and  $S_3$  are correct. For  $S_4$ , although all the acoustic events can be successfully detected, some of them cannot be correctly associated. For instance, when the vacuum cleaner ran in the living area, Node 10 received the highest signal energy, which is inconsistent with its designation as the primary sensor for the exhaust fan.

The results in this section show that both the conservative and incremental deployment strategies can effectively ensure the sensing results. Moreover, the data correlation and the unsupervised clustering/association algorithms adopted by Supero allow the sensors to be deployed in an *ad hoc* manner with considerable flexibility.

### 3.6.5 Experiments in House-1

House-1 is a one-story three-bedroom ranch house with a living space of about 150 m<sup>2</sup>. Compared with Apartment-1, it has more lights of various types (incandescent bulbs and standard/compact fluorescent lamps). The deployment consists of 7 TelosB and 3 Iris motes. The Iris motes detect both light and acoustic events. We conducted a controlled experiment for more than 5 hours. Groundtruth information was manually recorded and then rectified by checking the total power readings. In the experiment, each light sensor could detect multiple lights, and 40 false alarms out of totally 127 light events were raised by the light sensors, where 38 of the false alarms were identified by multi-modal data correlation. The remaining two false alarms were identified as outliers by the clustering algorithm. Table 3.4 shows the results. For one of the dining light events, a sensor monitoring the light missed the event, which resulted in a misclassification and error in estimating the energy of the dining light. From the background cluster of unattended power events, we observed that an unknown appliance with a power of 140 W was turned on for one minute about every 10 minutes. The appliance turns out to be a hot water dispenser at a sink. Moreover, the dispenser caused a missed detection of a guest bed light event, as the dispenser and the light were once turned on/off at the same time. The average error of Supero is 6.1%.

### 3.6.6 System Usability

We now present two case studies on how easily Supero can be deployed and configured by non-professionals. We recruited two homeowner volunteers to deploy Supero in their homes including a single-bedroom apartment (Apartment-3) and a two-story house with basement (House-2). We first introduced Supero and explained the deployment strategies to the volunteers, which took less than one hour. They then installed the sensors and configured the system using our web interface without any further instructions from us. For safety reasons, they did not install the TED5000.2

---

<sup>2</sup>The TED5000 probe needs to be hardwired to electrical service wires to get powered and connected to the gateway. Contactless power sensors [34], which are more friendly to non-technical end users, can be used instead.

In Apartment-3, the volunteer deployed 5 TelosB and 3 Iris motes to monitor all the appliances including 5 lights, a fridge, a microwave, and a fan. The deployment and configuration took only about half an hour. In House-2, the volunteer took about one hour to survey the appliances and another hour to install the sensors. He finally deployed 12 TelosB and 10 Iris motes to monitor 12 lights, an exhaust fan in the kitchen, a waste disposer, a dish washer, a fridge, a microwave, and three fans in three bathrooms respectively. The base station on the first floor could reliably receive data packets from sensors distributed on the two floors and basement. After the system deployments, we conducted controlled experiments to evaluate the deployments and configurations. We generated total power readings according to gathered groundtruth to run the algorithms. The event detection, clustering, and association results of the controlled experiments are correct in both deployments. These two case studies show that the non-professional users were able to quickly deploy Supero and ensure correct sensing results. We also find that both users preferred the conservative deployment strategy discussed in Section 3.5.2.

### **3.6.7 System Lifetime**

This section evaluates the lifetime of the battery-powered Supero sensors. In this experiment, we force the CPUs of the motes to stay active even though they would operate in low duty cycles (e.g.,  $\leq 5\%$  for Iris) in Supero. The radios are turned on only when there are packets to transmit. The TelosB motes report their battery voltages to the base station every minute. Fig. 3.14(a) plots the battery voltages of two TelosB motes with Alkaline and Lithium batteries, respectively, over time. The projected lifetime with Alkaline batteries is 79 days by conservatively setting the minimum operating voltage (MOV) to be 2.2 V although it is 2.1 V in datasheet [28]. With the high-capacity Lithium batteries, there is no observable voltage drop in one month. For the tested Iris mote, we enforce it to always work in the fast sampling mode. It piggybacks voltage reading to the acoustic feature packet. Fig. 3.14(b) plots the battery voltage of the Iris with Alkaline batteries. The tested Iris kept working from the 4<sup>th</sup> to the 9<sup>th</sup> day. Regression analysis shows that the projected lifetime is 40 days by conservatively setting the MOV of Iris to be 2.2 V, since the MOVs of the RF230

radio chip and ATmega1281 8MHz MCU on Iris are 2.1 V and 1.8 V. We note that the lifetime can be further extended by simply using Lithium batteries and duty-cycling the CPU of motes.

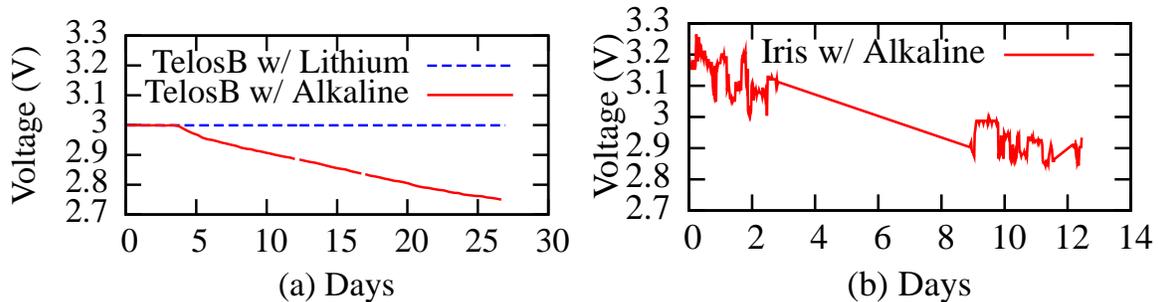


Figure 3.14: Battery voltage traces of TelosB and Iris.

### 3.7 Conclusion and Future Work

This chapter presents Supero – a sensor system for unsupervised residential power usage monitoring. In Supero, the multi-sensor fusion can effectively reduce sensing errors in complex household environments. By using unsupervised event clustering algorithms and a novel appliance association framework, Supero can autonomously estimate the power and energy usage of each monitored appliance. Extensive evaluation in five real homes shows that Supero can be deployed with considerable flexibility and provide accurate monitoring results.

Complementary to Supero, a few direct meters (e.g., the Zigbee-enabled KAW) can be applied to handle certain other appliances that have highly complex light/acoustic signal characteristics (e.g., TV) and power consumption profiles (e.g., furnace). In our future work, we will explore the use of other sensing modalities (e.g., infrared, seismic, and magnetic) to monitor these complex appliances. We will explore privacy-preserving strategies to prevent information leakage due to the wireless communications in Supero. Moreover, we plan to develop an easy-to-understand user manual to help non-professionals set up the sensor deployment, e.g., by video examples.

## CHAPTER 4

### A SENSOR NETWORK FOR REAL-TIME VOLCANO IMAGING

As sensor network technologies become more mature, they are increasingly being applied to a wide variety of environmental monitoring applications, ranging from agricultural sensing to habitat monitoring, oceanic and volcanic monitoring. This chapter presents some of the challenges facing a designer of a large scale seismic sensor network, the system designed to meet these challenges and the results from deploying a prototype network on active volcanos in Equador and Chile. Using low-cost hardware it is possible to record seismic activity within the local region of a volcano.

The remainder of this chapter is organized as follows. Section 4.1 presents the overview of real-time volcano tomography. Section 4.2 presents the design requirements the system must meet as well as the design alternatives considered. Section 4.3 presents the the system architecture and design. Section 4.4 discusses modelling system operation and the process used to assign the various computational elements to different nodes. Section 4.5 describes the system deployments in Ecuador and Chile. Section 4.6 describes the deployment experiences and results from these deployments while Section 4.7 concludes the chapter.

#### **4.1 Background of Volcano Tomography**

##### **4.1.1 Scientific Motivation**

Real-time volcano tomography creates a new paradigm for studying seismic activities and provides a deeper scientific understanding of the complex, time-varying dynamics of volcano. To study volcano dynamics, a three-dimensional (3-D) velocity model corresponding to the internal structure, as shown in Fig. 4.1, is to be constructed. The process of creating such a model from seismic event data is called *seismic tomography* [24, 25]. By observing the model changes, seismologists can predict how soon the volcano will erupt and issue early warning to evacuate surrounding areas.

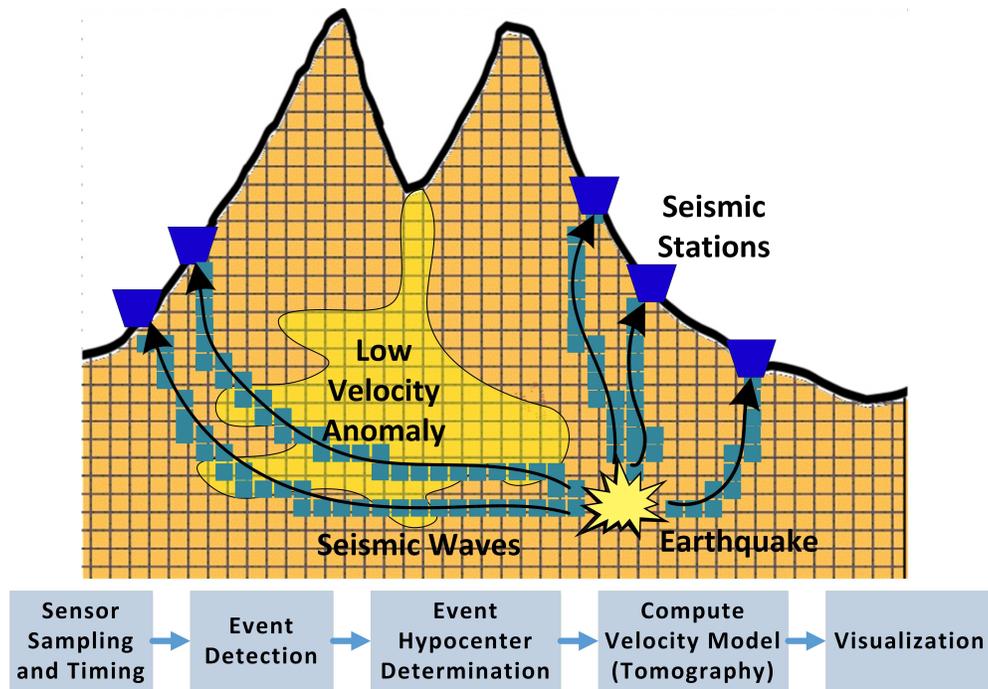


Figure 4.1: Seismic tomography and real-time signal processing pipeline. The tomography estimates a velocity model consisting of the seismic wave propagation speeds in cubic blocks beneath the volcano surface.

#### 4.1.2 Tomographic Processing Pipeline

This section introduces the steps in the tomographic pipeline in Fig. 4.1. Tomography begins with obtaining readings from seismic sensors. The seismic signal frequencies in a volcanic earthquake range from sub hertz to 40 Hz. Volcano monitoring systems often adopt a sampling rate of 100 Hz, which can fully capture the seismic events. However, such a high sample rate makes it difficult to collect raw data in real time from a large-scale (e.g., hundreds of nodes) WSN due to limitations of energy and bandwidth at current, battery-powered nodes. This requires the nodes to perform in-network event detection and only send short event messages through the network.

Seismic event detection detects the occurrence of a seismic event and determines the arrival times of primary waves (i.e., P-phases) at the sensor nodes. The STA/LTA [9] and the ARAIC (Autoregression with Akaike Information Criterion) [41] algorithms are widely used for seismic event detection. The STA/LTA computes both a long-term average and a short-term average for the signal, and makes a detection if the ratio of the two averages exceeds a predefined threshold.

The event magnitude, the window sizes of the averages, and the threshold setting affect STA/LTA's sensitivity to events and thus the accuracy in estimating events' onset times. The ARAIC constructs two autoregression models based on the seismic signals before and after each time instant and yields a time instant that maximizes the dissimilarity between the two autoregression models as the P-phase arrival time. Thus, ARAIC requires no thresholds and is more accurate than STA/LTA, but at the cost of increased computation overhead. For instance, processing a 16-second seismic signal requires computations on 6,500 floating point variables.

Once a seismic event is detected, the P-phases from multiple sensors are used to estimate the earthquake hypocenter including its source location and origin time. Geiger's method [50], a widely used hypocenter estimation approach, applies the Gauss-Newton nonlinear optimization method through a series of iterative linearization steps.

Volcano tomography [24, 25] involves tracing the travel paths of the seismic signals (ray paths) from the event source location to the monitoring stations and computing the travel times. Note that, as the material density beneath the volcano surface is not evenly distributed, the ray paths are curves, as illustrated in Fig. 4.1. The velocity model is represented as a vector of slowness perturbations, which is computed from the data collected by many sensors for many events and is thus compute-intensive.

For example, a 1992 study of Mount St. Helens [24] utilized seismic data consisting of 5,454 local seismic events recorded by 39 stations over 10 years. This produced the 35,475 rays used to model a 27.5 km by 21 km target area. Increasing station density by a factor of 10 to 20 would result in a corresponding reduction of the time required to obtain sufficient ray coverage. Thus, a viable option is to deploy a large number of nodes, i.e., 250 to 1,000 nodes to cover a typical volcano. This is economically infeasible for traditional platforms. It motivates us to develop easy-to-deploy and inexpensive volcano sensor platforms with a target cost budget of approximately 500 US\$ per node.

To investigate the impact of increasing station density we created a model of the internal structure of the Chilean Llaima volcano. Within this structure we modelled a volume to represent

a magma through which seismic waves would travel at a different velocity. Using this model, we explored the impact of the number of monitoring stations and events on the computed model accuracy as compared to the original model. The following section describes the results of this evaluation.

### 4.1.3 Spatial Coverage

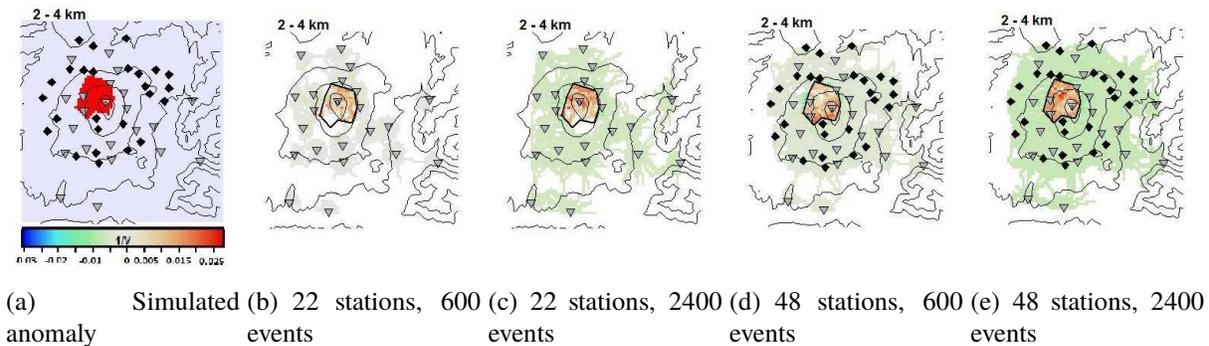


Figure 4.2: Tomography results using simulated seismic data.

Sufficient 3-D ray coverage is a key requirement of volcano tomography. That is, each cubic block in the 3-D model should have multiple rays passing through it. The 3-D ray coverage also affects the resolution of the model. Ray coverage can be increased either by including more events or by increasing station density. Wide geographic coverage is also required to ensure the 3-D model covers the entire internal structure of the volcano. The 1992 study of Mount St. Helens [24] utilized seismic data consisting of 5,454 local seismic events recorded by 39 stations over 10 years. This produced 35,475 rays used to model a 27.5 km by 21 km target area. Increasing station density by a factor of 10 to 20 would result in reduction of the time required to obtain sufficient ray coverage.

To investigate the impact of the numbers of sensors and events on the model resolution, a seismic simulation model of the Llaima Volcano was developed. The model consisted of grid 80 x 80 covering the volcano with 7 layers corresponding to the velocity model at the depths of 0, 2, 4, 5.5, 8, 10.5, 13 km. To model a seismic anomaly, corresponding to a feature that might represent magma, polygons were placed in each layer of the model and assigned a velocity factor

from the corresponding layer. Fig. 4.2(a) shows the polygon at the 2 to 4 km depth. Using this model, we generated simulated events at random locations. For each of these events we computed the travel time to each monitoring station. Gaussian noise was added to this travel time and used as the event arrival time. The monitoring station locations, events' hypocenters, and event travel times were used to compute a velocity model. We conducted simulations with 600, 1200, and 2400 events using 22 and 48 monitoring stations. Due to space limitations, only the 2-4 km layer is discussed here. The other layers showed similar results. As both the numbers of stations and events increase, there are rays covering wider areas of the model. This is illustrated in the figures by areas changing from white to a color representative of the inverse velocity (i.e., slowness). Compared with Fig. 4.2(c), Fig. 4.2(d) contains a larger area having color, which suggests that increasing the station density has a greater impact on the model coverage than increasing the number of events. As the number of events increases (Fig. 4.2(d) compared to Fig. 4.2(e)), the slowness in the region of the anomaly is more accurately modeled, which is indicated by the region's color more closely matching the color in the simulated ground truth in Fig. 4.2(a).

In the above simulation, 20,000 to 50,000 rays were required to compute the tomography. To obtain a sufficient ray coverage, we require either a large number of seismic events collected over a long time period or a large number of monitoring stations spread over a large area. When a volcano is in crisis, it is desirable to update the model one to four times per hour. Such a high temporal resolution requires a large number of rays collected in a short time period. Thus, to obtain this number of rays in a short time period a large number of nodes, i.e., 250 to 1000 nodes must be deployed to cover a typical volcano. This is economically infeasible for traditional platforms. This motivated us to develop easy-to-deploy and inexpensive volcano monitoring platforms with a target cost of approximately 500 USD per station.

## 4.2 Design Requirements

We aim to meet the following design requirements:

**Delay and data fidelity requirements:** Data-intensive sensing applications typically require

continuous data sampling at high rates. For seismic applications, this rate is typically 100 Hz. It is also important for the sampling rate to be constant with no breaks in the data to minimize required post processing. The detected events will be correlated across several nodes thus requiring the data samples to have time stamps accurate to the sub-millisecond.

Nodes must be able to process the signal fast enough to neither fall behind nor overflow memory buffers. Thus, the choice of event detection algorithms impacts the required node computational capabilities. Notification of a seismic event should be reported in seconds rather than minutes including the delays of hypocenter computation and data communications. Tomography computations needed to be completed in 15 to 30 minutes to ensure timely model updates.

A common problem with the STA/LTA event detection algorithm [47, 30] is false alarms. Systems sensitive enough to detect weak seismic events can be triggered by man-made events such as vehicle movement, vibrations caused by wind moving nearby trees, etc. These events might be detected by sensors close to the source but not by sensors further away. The system should incorporate false alarm suppression methods.

**Communication requirement:** To achieve a high degree of coverage, nodes should be deployed with 200 m to 400 m spacing over an area of several square kilometers having large variations in terrain. Cases can arise where nodes might be up to one kilometer apart with an intervening ravine. The area surrounding a volcano is typically remote with limited access, making it difficult to access the nodes once deployed. As a result, special attention must be paid to communication performance and proper planning. It is also desirable to remotely monitor node health and collect event detection results, which necessitate remote network connection. However, the remote volcanic areas often have limited network coverage with satellite communication as the only connection. Satellite links have limited bandwidth (less than 500 kbps) and high latencies (one to three seconds of round trip time). These delays and limitations must be accounted for to ensure timely reporting of detected events.

**Power requirement:** A volcano monitoring system must operate unattended for several months when a volcano becomes active, with sensor nodes sampling seismic signals continuously. In such

a case, common power conservation techniques such as sleep scheduling are often inapplicable. Thus, it is desirable to combine power conservation with power harvesting to prolong lifetime while achieving sufficient spatiotemporal coverage of seismic activities for volcano tomography.

**Packaging requirement:** Volcanic regions have harsh environment conditions for sensor nodes. Volcanic ash is extremely fine and can find its way into even the smallest places. In addition, areas may receive heavy rain storms lasting for long periods of time. Due to the remoteness of the deployment site, the nodes should be small and light enough that one person could comfortably carry three to four units.

### 4.3 System Architecture and Design

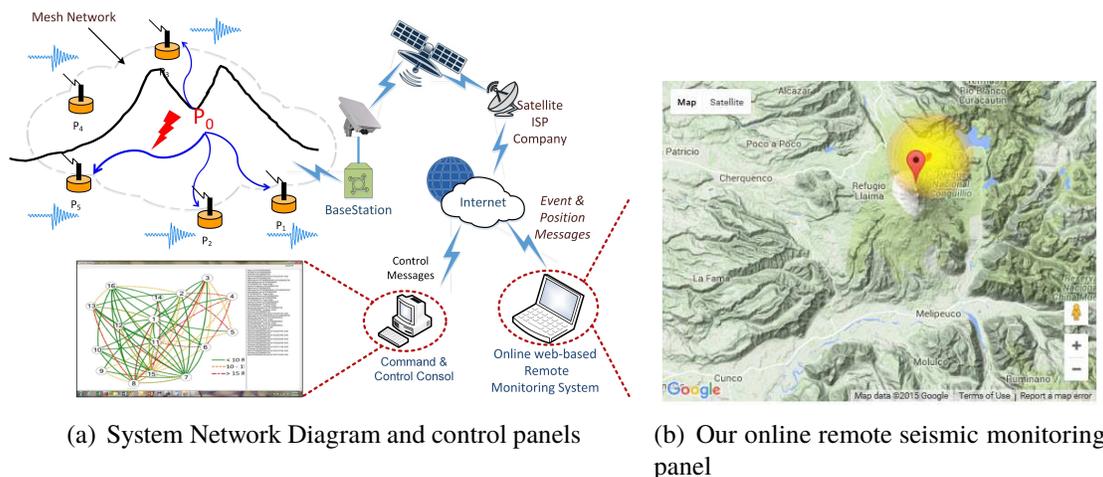


Figure 4.3: General system architecture and online remote monitoring panel.

To meet the requirements outlined in Sec. 4.2, we have developed a multi-tiered system shown in Fig. 4.3(a). It consists of sensor nodes communicating over a mesh network to a base station. The sensor nodes handle sensor sampling, time-stamping and event detection. Event information and position information is exchanged between stations over the mesh network. The base station computes the seismic event hypocenter and forwards the associated event information to a remote server. The base station also provides a remote command and control link for the sensor network. The remote server provides long-term event storage, event visualization, tomographic inversion and velocity model visualization using a web interface shown in Fig. 4.3(b).

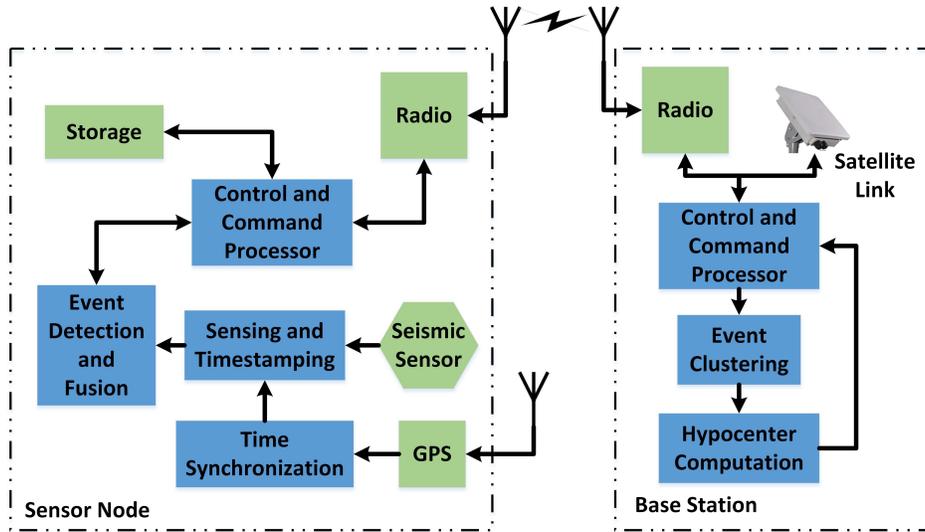


Figure 4.4: Data processing components of sensor node and base station.

### 4.3.1 Signal Processing and Event Detection

Each sensor node performs a number of processing functions as shown in Fig. 4.4. These functions are divided between interrupt-level and the main processing loop. Sampling and time synchronization both occur at interrupt-level while all other processing occurs in the main processing loop. The seismic sensor is sampled at 100 Hz. These samples are stored locally on a SD card. Once stored, the signal is processed through a digital bandpass filter to remove the direct current (DC) component and eliminate high frequency noises.

Given the severe limitations on the energy and compute resources of the system, a key challenge in the design of seismic event detection algorithms is to achieve a trade-off between detection accuracy and compute overhead. In addition, it is desirable to adopt existing, well established algorithms in the seismic community, such as STA/LTA and ARAIC, because many existing post-processing geophysical tools are designed to work with these algorithms.

The Short Term Average/Long Term Average (STA/LTA) algorithm [47, 30] is commonly used to detect seismic activity. STA/LTA uses the ratio of the short term average to the long term average to detect events. The long term average establishes the noise floor. When the ratio exceeds a threshold value an event is declared. The associated time is used as a rough estimate of the event

starting time. The various STA/LTA parameters are used to tune the detector’s sensitivity and have a secondary effect of impacting the time associated with the event. The STA/LTA also requires the seismic signal be properly filtered and the DC component removed. Another event detection algorithm is the ARAIC. This algorithm is more computationally intensive than STA/LTA but provides a more accurate event arrival time. If this algorithm is to be used the system must have sufficient computational power to perform the required computations without missing samples.

Keeping these requirements in mind, we developed a new seismic event detection framework consisting of the following two steps. First, the preliminary event detection (CoursePick in Fig. 4.5) is performed using STA/LTA. If the preliminary detection result is negative (no event), the node waits for the next sample; otherwise, the node collects an additional 8 seconds of samples. These samples along with the 8 seconds of samples preceding the event are passed to the ARAIC algorithm [40] to determine the event start time (FinePick in the Fig. 4.5). This event detection approach achieves a desirable trade-off between accuracy and energy consumption by taking advantage of complementary characteristics of two existing algorithms: STA/LTA is highly efficient but sensitive to settings and event magnitude, while ARAIC is more robust but computationally expensive.

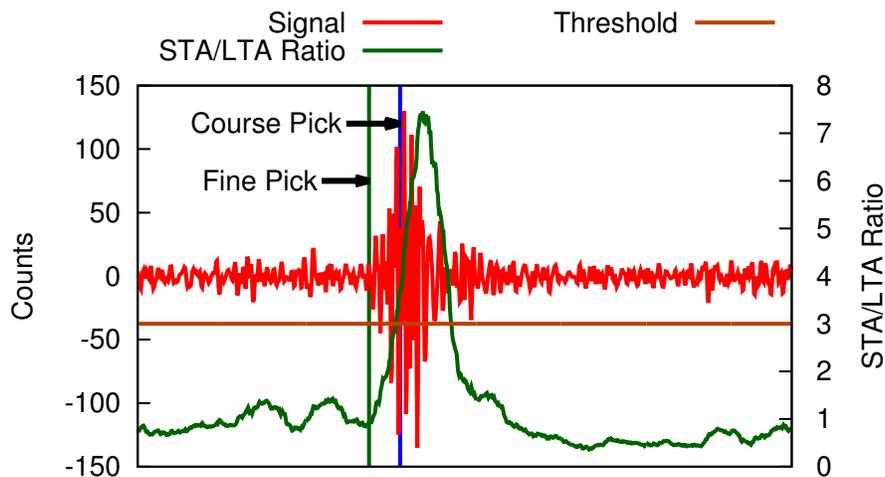


Figure 4.5: STA/LTA ratio in response to a seismic signal.

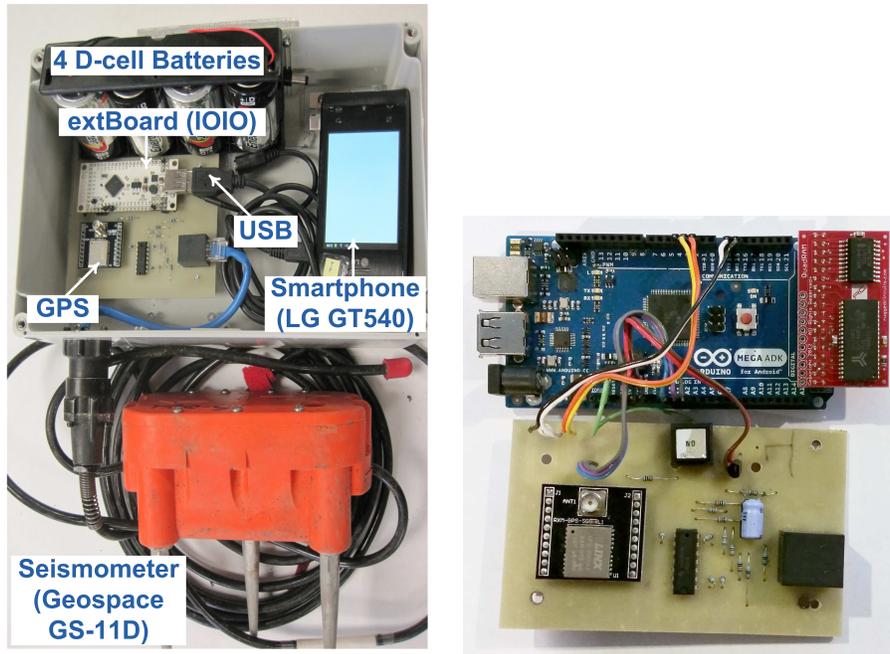
Table 4.1: Key Characteristics of a Traditional Seismic Station versus a our sensor nodes

	Traditional Station	Gen 1	Gen 2	Gen 3	Gen 4
<b>Seismic Sensor</b>					
Model	GURALP CMG-40T	PASSCAL L-28-3D	PASSCAL L-28-3D	Sercel L-28LB	Sercel L-28LB
Type	3-axis broadband	3-axis long period	3-axis long period	1-axis long period	1-axis long period
Sensitivity	1000 V/m/s	40 V/m/s	40 V/m/s	40 V/m/s	40 V/m/s
<b>Digitizer</b>					
Model	Reftek 130	Microchip	Microchip	TI	TI
ADC	24 bit delta sigma	10 bit	10 bit	24 bit delta sigma	24 bit delta sigma
Storage (GB)	8	N/A	N/A	32	32
Event Triggering	STA/LTA	STA/LTA	STA/LTA	STA/LTA, ARAIC	STA/LTA, ARAIC
<b>Power Source</b>					
Battery	55 Ah Lead Acid	4 D-Cells	4 D-Cells	7 Ah SLA	7 Ah SLA
Solar Panel	50W	N/A	N/A	20W	20W
Size	36 in x 36 in	N/A	N/A	12 in x 24 in	12 in x 24 in
Total Weight	90 lb	10 lb	10 lb	10 lb	10 lb
Total Cost	~\$22,000 US	~\$500 US	~\$500 US	~\$500 US	~\$500 US

### 4.3.2 Sensor Node Design

Several alternative sensor node designs were considered. Traditional wireless sensor network platforms, such as the TelosB series [28], while very energy efficient, have limited computational and communication capabilities. Nodes in this class can perform STA/LTA event detection but do not have sufficient memory or speed to perform the more complex ARAIC algorithm or the hypocenter computations. When operation in a continuous sampling mode, their energy consumption increases significantly. The Zigbee radios have sufficient throughput (250 kbps) to send event messages but range is limited to 20 to 100 meters, insufficient for the spacing needed by this application. After considering these options, it was decided the current class of low energy motes were not appropriate for this application. Through a series of rapid prototypes we investigated alternative hardware designs. The hardware differences are summarized in Table 4.1 and discussed in the following sections along with the lessons learned from each prototype. Two deployments allowed us to investigate and better understand the environmental conditions the nodes would be subjected to. These are discussed in Section 4.5

**Generation 1/2:** An initial question which arose early in the design process was “Can the high



(a) Generation 1 - Deployment: Tungurahua, Ecuador, July 2012 (b) Generation 2 - Arduino Test Node

Figure 4.6: Seismic Node Prototypes

sampling requirement be met with low cost hardware?” To evaluate this, a prototype (Fig. 4.6(a)) was developed and deployed on the Ecuadorian Tungurahua Volcano. This prototype consisted of an Android phone with an attached external board. The external board contained a 12 bit digitizer, signal amplifier and GPS receiver controlled by software running on the phone. During the 5-day deployment period this prototype was able to successfully record a seismic event 20km away (Fig. 4.8).

While capable of recording nearby seismic events, this prototype also identified several issues. First, a 12 bit ADC did not have sufficient dynamic range to capture all events of interest to seismologists. Second, the high sample rate did not allow the phone to enter its low power sleep mode resulting poor battery performance. Finally, the hardware was not fast enough to capture the GPS one pulse per second digital output and could not synchronize its clock with the GPS.

To resolve these issues, the Arduino family of processors were used to investigate different approaches to clock synchronization and to determine the amount of clock drift over long periods

of time. The generation 1 sensor board was attached to an Arduino Mega ADK processor (Fig. 4.6(b)). It was discovered that without any synchronization the internal clock of the Arduino Mega ADK system drifted 708 microseconds every second. The Arduino kernel was modified to allow adjusting the clock to compensate for the drift. Using the GPS 1pps signal along with this modified kernel it was possible to calculate the drift and adjust the tick interval reducing the drift to 21 microseconds per second. Unfortunately, the Arduino Mega ADK, even with an extended memory board attached, did not have sufficient built-in memory or speed to execute the needed event detection algorithms.

**Generation 3:** For the third generation of the node hardware we employed the Arduino Due process and a custom board containing the specialized sensing components (Fig. 4.7(a)). Fig. 4.7(a) shows the sensor node we designed for this work. This design provides similar quantization resolution at 2% of the cost and 10% the weight of a traditional portable seismic monitoring station. Table 4.1 lists the key characteristics of traditional portable stations versus our nodes.

The Arduino Due was chosen due to its computational power and the desire to use an existing processor design where possible. It contains an 84 MHz AMTEL 32-bit processor which is able to digitally filter a sample using a 204th order FIR filter and compute the STA/LTA ratio in 0.15 milliseconds. It can perform an ARAIC computation on a 16 second signal (1600 samples) in 3.9 seconds. Proper buffering allows the node to handle 100 samples per second without losing samples.

The analog-to-digital converter (ADC) chosen was a TI ADS1281. This is a 24-bit ADC designed specifically for seismic applications and is capable of sampling at up to 4000 Hz. To amplify and condition the seismic sensor signal, a TI THS4521 differential amplifier with a gain of 100 feeds the ADC. The ADC's internal clock is used to control the sampling rate. When a conversion has been completed an interrupt fires an interrupt routine accessing the ADC and placing the reading along with the current time into a circular buffer. Samples are removed from this buffer and analyzed in the main processing loop.

A Digi International XBee-Pro 900HP 900 MHz RF module provides network communications.

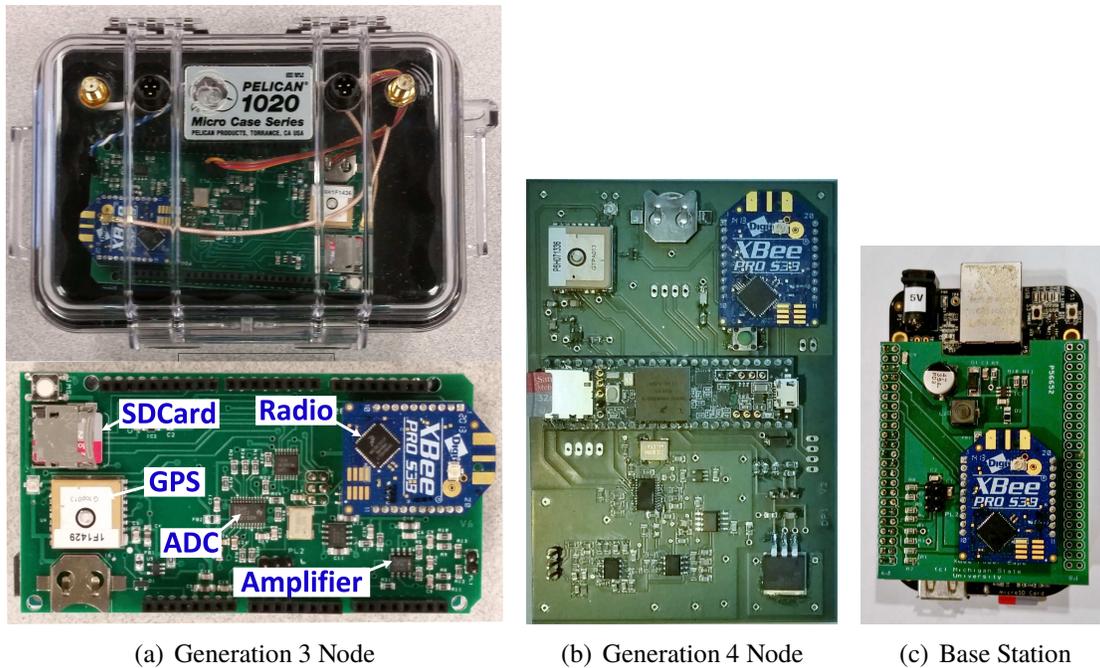


Figure 4.7: Seismic Monitoring Nodes

This module support DigiMesh network topology with a RF data rate of 200Kbps. 900 MHz was chosen over 2.4GHz due to the reduced signal absorption in heavily wooded areas. The manufacturer claims the unit is capable of a line of sight range of 6.5 km with a 2.1 dB dipole antenna. This radio provides sufficient capability to allow a large number of nodes to exchange short messages, i.e., position information and event detection results. To facilitate offline validation of detection results, each node stores the raw signal to a 32GB SD card. Using a compact delta based format, one hour of samples requires 1.4 MB storage.

We include a GlobalTop Technologies, Inc. MTK3339 GPS chip on each node. This chip provides geolocation information, global time, and a pulse-per-second (PPS) signal. This signal is used to interrupt the processor and synchronize the internal processor clock to global time. To mitigate the increased power consumption, control circuitry was included allowing the GPS chip to be completely powered down. A small coin battery is used to maintain the GPS memory during power down allowing quick restart when power is reapplied.

Sealed lead acid (SLA) batteries were chosen to power the nodes. While unnecessarily having the highest energy density, SLA batteries have certain advantages over other choices. They are

inexpensive, commonly available even in third world countries, operate over a wide temperature range, and survive large numbers of charge/discharge cycles. For slow discharge rates, the battery should not be discharged below 10.5 volts. For a small 7 Amp-hour SLA battery with a 0.15 Amp discharge current, the estimated runtime is 3.4 days.

A 20 watt solar panel maintains the battery charge. The charge controller regulating battery charging is a potential point of failure. A failure could cause battery overcharging and subsequent rupturing resulting in the release of corrosive acid. To prevent damage to the electronics the battery was placed in a separate enclosure to isolate it from the electronics.

### 4.3.3 Design Lessons Learned and Generation 4

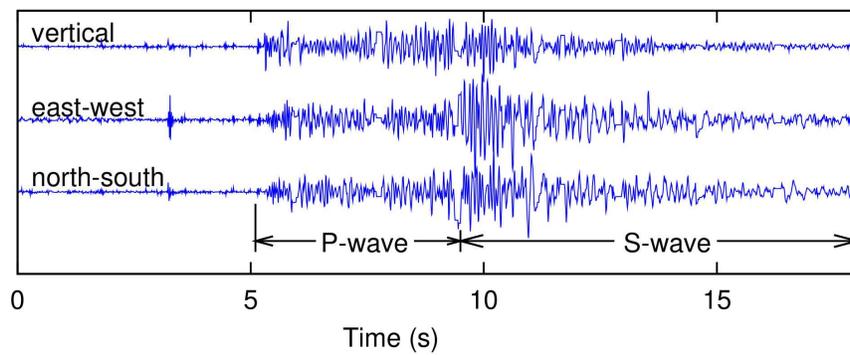


Figure 4.8: Recorded Ecuador Seismic Event

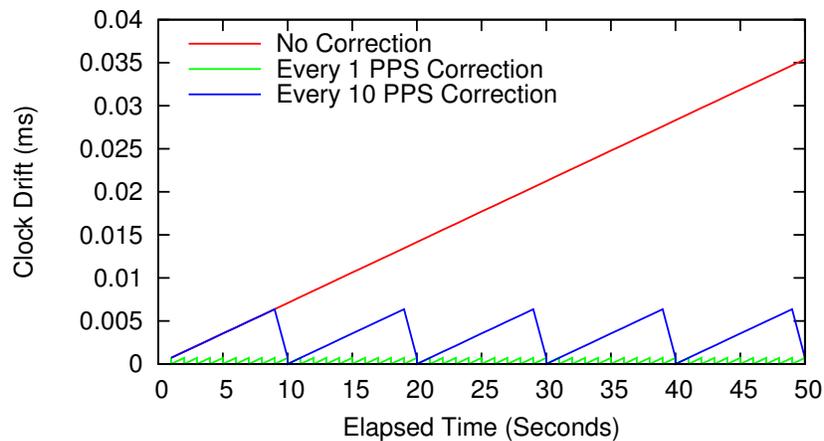


Figure 4.9: The clock drift and correction by GPS 1PPS

Each sensor node generation improved based on lessons learned during the design process. With Generation 1, it was shown an inexpensive ADC with a simple input amplifier could detect local seismic events as shown in Fig. 4.8. This event was approximately 25 km away from the sensor location. The external IOIO board used to interface the phone to the GPS was unable to properly detect the timing signals. This could have been remedied by developing custom code for the IOIO but we were never able to reliably load the custom code. Another issue is the lack of real time support in the Android operating system. This caused unreliable time stamps. The combination of these two factors lead us to explore the Arduino processors for the next generation. While the packaging chosen for this first generation did not suffer any failures during the 2 week Ecuador deployment, further testing identified other issues. The thin walls of the enclosure were easily cracked during shipment and the cover was subject to warping negatively impacting the weather proof nature of the enclosure. This lead us to explore the Pelican cases.

The second generation was primarily used to validate that closer integration with the GPS resulted in more accurate time stamping and clock synchronization (Fig. 4.9). With this we were able to explore clock drift and clock drift correction. The system clock on the Arduino Mega256 drifts 35 ms every 60 seconds. Using the GPS to correct the system clock, the drift can be reduced to the sub-millisecond range. This requires the GPS to be in continuous operation with the corresponding increase in power consumption. Using a modified Arduino kernel, we added the capability to compensate for the clock drift by changing the number of microseconds added to the system time each clock tick. Using this approach we were able to reduce the clock drift without having to operate the GPS continuously. Thus allowing the GPS can be turned off for periods of time.

The third generation was a major refinement of the hardware and packaging. The processor had sufficient processing power to eliminate the need for the phone. The packaging for the node and the battery were such that the node and antennas could be placed inside the battery case reducing the space needed during shipment. The cases were strong enough to survive rough handling without suffering any failure of their waterproof nature.

Deploying the Generation 3 node identified several deficiencies with the hardware. The first was power consumption related to not being able to place the processor into its lowest power state due to having to maintain the system times used to time stamp the seismic samples. This was caused by a limitation of the Arduino Due and how its internal clock was maintained. The second issue identified related to noise in the signal processing chain. While the board had sufficient shielding and bypassing the seismic amplifier still exhibited a high level of noise that interfered with detecting weak seismic events. This could only be by a complete redesign of the board and seismic amplifier. The third issues was the SPI bus contention between the ADC and the SD-Card using the same bus (See Section 4.6.2 the discussion of the impact on data fidelity). While this was initially remedied by using 3 General Purpose I/O pins and simulating SPI bus operation in software this was not an optimal solution. Finally, there were memory management issues which caused the node to stop functioning at random times. To alleviate these issues a redesign of the node was undertaken. The Arduino Due processor was replaced with a Teensy 3.6 processor and the a new board layout was developed providing the following features:

- Realtime clock - The Teensy 3.6 processor includes a realtime clock that continues to operate even when the processor is placed into its lower power state. In addition, when operating at approximately the same clock frequency as the Due processor the Teensy processor consumes less energy.
- Board Layout - The smaller physical size of the Teensy 3.6 processor allowed the sensor board to be reconfigured allowing better separation of the analog and digital components. This reduced noise in the analog signal chain.
- Seismic amplifier redesign - The seismic amplifier was redesigned to include signal filtering reducing high frequency signal aliasing and further lowering the noise levels.
- Multiple SPI busses - this allows the ADC and SD-Card to utilize separate SPI busses removing bus contention.

- Additional memory and hardware floating point computations - the DUE processor has 96KB of SRAM while the Teensy 3.6 has 256 KB of SRAM. The Teensy also adds floating point hardware instructions. This greatly reduces the computation time of complex event detection algorithms. (See Table 4.3)

Fig. 4.7(b) shows the resulting Generation 4 node.

#### **4.3.4 Base Station Design**

The base station is composed of a Beaglebone Black board containing a 1GHz 32bit processor running Debian Linux. A custom cape containing a power regulator and a XBee radio module attaches to the processor (Fig. 4.7(c)). This cape allows the battery and solar panel voltages for both the base station and satellite link to be monitored. The provided Ethernet connector allows the board to communicate with a Hughes 9502 one-piece integrated satellite terminal providing a 448 Kbps broadband connection. This terminal uses 3 to 4 watts with an active TCP/IP connection, 0.01 watts in hibernate mode and wakes up in under 30 seconds with LAN activity.

The control software executed by the base station receives detection events from the sensor nodes. Geiger's method is used to compute the hypocenter of the detected seismic events. Events received by a sufficient number of nodes and within the volcano's vicinity are forwarded over the satellite link to a remote server for storage and display. The remote server computes the velocity model and visualizes the result.

#### **4.3.5 Packaging**

The sensor nodes and base station are housed in Pelican 1020 micro cases. These cases, along with the power and sensor connectors, meet the IP67 standard for dust and water. A potential problem exists with the connections for the radio and GPS antennas. The RP-SMA connections used for these connections are not inherently waterproof. To waterproof these entry points, a recess was

machined around the entry hole to accommodate a rubber O-ring. Once the connector, O-ring, lock washer and nut are tightened into place a water proof seal was made.

To eliminate ground effects, a PVC mast raises the radio and GPS antennas to 1.5 m above the ground. This height maximizes the radio range. The Pelican micro cases are attached to the top of the mast using an aluminum bracket. The GPS antenna magnetically attaches to the mast as well.

The battery and charge controller is housed in a MTM survivor box. This box is slightly larger and nicely accommodates either a 7 or 9 amp-hour 12 volt SLA battery.

#### 4.4 System Modeling & Dynamic Task Assignment

Our system consists of three tiers: sensor node (Tier 1), base station (Tier 2), and remote server (Tier 3). A key design question is how to assign processing tasks to specific tiers subject to processing delays, communication throughput and system lifetime constraints. To assign tasks to the processing tiers, it is necessary to model the processing delays, communications throughput and system power consumption. To this end, we carefully analyzed delays at various stages of the information processing pipeline and propose the following task assignment scheme. Our scheme takes advantage of complementary compute/communication capabilities of different tiers of our system, while minimizing usage of the networking layer.

The decision of which tier to place a particular processing task depends on the computational costs associated with the task on each tier as well as the communication cost to transfer the required data. Subject to not exceeding the task's processing delay bound  $D$ , computing a task should be transferred to the next higher tier, if the following relationship is true:

$$t_i > t'_i + c_{i,i+1} + t_{i+1}, \quad (4.1)$$

where  $t_i$  is the processing time for the task if it resides on tier  $i$ ,  $t'_i$  is the tier  $i$  processing time if the task moves to the next higher tier  $i + 1$ ,  $c_{i,i+1}$  is the time to transfer the required data from tier  $i$  to the next higher tier, and  $t_{i+1}$  is the next higher tier's processing time. Evaluating Eqn. 4.1 for each task produces an initial set of task assignments.

Table 4.2: Application processing deadlines

Task	Deadline	Consequence of violating deadline
Sensor sampling and timing	< 10ms	Samples dropped
Event Detection	< 10 ms	Samples dropped unless sufficient buffering
Hypocenter Determination	< 10s	Lack of timeliness reporting event source to end user
Compute Velocity Model	15 min	Slower model update rate
Visualization	< 1 min	Image unavailable to user in timely fashion

#### 4.4.1 System Delay Modeling

To determine which tier a processing task should be assigned, it is necessary to consider the following: application processing constraints ( $D$ ), execution times for each task ( $t_i, t'_i$ ), and communication throughput ( $c_{i,i+1}$ ). Table 4.2 lists the application processing deadlines ( $D$ ) along with the consequence of violating the deadline. For this application, the most critical deadlines relate to the Sampling/Timing task and the Event Detection task, where missing the deadline results in missed samples. For the other tasks, Hypocenter Determination, Tomography and Visualization, violating the deadline will result in slower model updates.

Table 4.3: Task execution times by tier

Task	Tier 1		Tier 2 Base station	Tier 3 Cloud	Assigned Tier
	Gen 3 Node	Gen 4 Node			
Sampling and Timing	100 $\mu$ s	< 50 $\mu$ s	na	na	1
Event Detection (STA/LTA)	0.15 ms	51 $\mu$ s	5 $\mu$ s	na	1
Event Detection (ARAIC)	3.9 sec	206.2 ms	6 ms	na	1
Hypocenter Determination	na	na	278 ms	33 ms	2
Compute Velocity Model	na	na	na	81 sec	3
Visualization	na	na	na	.32 sec	3

The task processing times for each tier are given in Table 4.3. Some tasks can only be executed on a particular tier due to hardware restrictions or data requirements. For example, the Sampling and Timing task can only be performed on Tier 1 since the sensors are connected to the nodes that

make up that tier. The Hypocenter Determination cannot be run on Tier 1 since it requires data from all the nodes before the calculations can be performed. In addition, the Tier 1 nodes have insufficient memory to execute the task. The execution time for the Hypocenter Determination depends on the number of nodes detecting the event, while the time for the Tomography task depends on both the number of nodes and the number of events.

The radios in our system use both link-wise and end-to-end acknowledgements. The maximum data throughput is determined by both the single hop link speed and the number of hops. Based on data provided by the radio manufacturer, the data network throughput can be modeled by

$$T_{kbps} = 91.23h^{-0.945}, \quad (4.2)$$

where  $h$  is the number of hops and  $T_{kbps}$  is the data throughput in kilobits per second. Satellite links have high latency with round-trip time of 1 to 3 seconds. They provide 464 kbps link speeds. Thus, we can compute the communication delay based on the packet size and the number of hops between tiers.

#### 4.4.2 System Lifetime Modeling

System lifetime depends on the battery capacity and the amount of current being drawn from the battery. For SLA batteries, at a constant discharge current, the voltage decreases linearly from its fully charged voltage to its terminal discharge voltage. To maximize battery life, SLA batteries should not be discharged below this terminal voltage. When the SLA battery capacity and discharge current are known, Peukert's Law can be used to estimate battery lifetime. If the system operates in one power mode only, the current drawn can be measured offline and used to estimate system lifetime. For cases when a system has multiple power modes, e.g., sleep and active, the average power consumption can be computed based on the modes' power consumption and the percent of time spent in that mode. Specifically,  $I_{avg} = \sum_{i=0}^n I_i \cdot P_i$ , where  $n$  is the number of power modes,  $I_i$  and  $P_i$  are the current and percent time in mode  $i$  subject to  $\sum P_i = 1$ . Based on the initial task assignments,  $P_{active}$  for a given tier can be estimated as  $P_{active} = \sum_{i=0}^n t_i/f_i$ , where:  $t_i$  is

the processing time and  $f_i$  is the frequency of execution of task  $i$  on the tier. Thus, SLA battery lifetime is

$$T_{lifetime} = H \left( \frac{C}{IH} \right)^k, \quad (4.3)$$

where  $C$  is the battery capacity (in Ah),  $H$  is the rated discharge time (in hours, 20 hours for SLA batteries), and  $k$  is Peukert's constant (1.25 for SLA batteries).

Estimating system lifetime using Eqn. 4.3 does not take into account energy harvesting. Since SLA batteries discharge voltage varies linearly for a constant discharge current, by monitoring the change in battery voltage over time, the charging or discharging time can be computed as follows:

$$\frac{\Delta v}{\Delta t} \begin{cases} > 0, \text{ charging,} & T_{charge} = (V_T - V_c) * \frac{\Delta t}{\Delta v}, \\ = 0, \text{ equilibrium,} & \\ < 0, \text{ discharging} & T_{discharge} = (V_D - V_c) * \frac{\Delta t}{\Delta v}, \end{cases} \quad (4.4)$$

where  $V_c$  is the current battery voltage,  $T_{charge}$  is the time to recharge the battery,  $V_T$  is the battery fully charged voltage,  $V_D$  is the terminal discharge voltage, and  $T_{discharge}$  is the time to discharge the battery (system lifetime).

With energy harvesting, the time to discharge or recharge the battery depends on both the discharge current and the amount of energy harvested, i.e., the net battery current. Using the change in battery voltage, the net battery discharge current can be estimated by  $I = \frac{C}{H \cdot \left(\frac{t}{H}\right)^k}$ , where  $C$ ,  $H$ , and  $k$  are the same as in Eqn. 4.3 and  $t$  is  $(V_T - V_D) \cdot \Delta t / \Delta v$  in minutes.

#### 4.4.3 Dynamic Task Assignment

Using the processing times and communication rates presented in the previous section, task assignments were made based on evaluating Eqn. 4.1 when assigned to a tier (Table 4.3). For instance, the Event Detection task was assigned to the sensor node (Tier 1) versus the base station (Tier 2) one hop away. This task can be computed in 0.15 milliseconds ( $t_1$ ) on a 32-bit 84MHz Arduino processor. It takes approximately 19 milliseconds ( $c_{12}$ ) to transmit 1 second of sensor readings and

associated timing information one hop. In this case, no matter how fast the task can be computed in Tier 2 ( $t_2$ ), the inequality will be false and the processing should remain on Tier 1.

Another consideration in deciding tier assignments is the data needed. For example, event detection only requires data from a single sensor. In contrast, the hypocenter computation needs event information from multiple nodes. For rendezvous operations such as this, the communications impact for both the single node case and the multi-node case must be considered. When a node detects a seismic event, the associated event time must be transmitted to the base station.

The seismic event information can be transmitted in a 16 byte packet, the packet rate depends on the frequency of seismic events and cannot be predicted. It must be determined if the network has sufficient throughput to handle the event packets within a large network. To determine this we considered nodes placed in a 20km diameter ring surrounding a volcano with 250 meter spacing such that each node could communicate with at least two other nodes. To complete this ring approximately 500 nodes would be required. A single node transmitting one 16-byte packet every 10 seconds would require a data throughput of 0.0128 kbps, assuming no transmission failures. Assuming the packet must travel through half the nodes to reach the base station a 250 hop path provides 0.49 kbps data throughput. Since all messages must flow through the final link to the base station the aggregate data throughput needed for a 500 node network is 6.4 kbps. This is well below the 91 kbps single hop data throughput rate.

With a such a large network, impacts of delays and lost packets must be considered from the application point of view. For seismic tomography each event packet send by a node results in 1 ray to be included in the model calculation. Each ray has the potential to provide the information needed to compute the velocity in a new portion of the model. Since the model will be computed using thousands of rays the loss of a low percentage of rays should have minimal impact on the resulting model. The impact of a delayed packet depends on how frequently the model will be recomputed. For example, assume the model were recomputed once 25,000 to 50,000 new rays have been collected. If seismic events were occurring once every minute and reported by half the stations (250 nodes reporting) it would take 100 to 200 minutes to receive the number of events

required to recompute the velocity model. Since delays will most likely be on the order of seconds the resulting system will be tolerant to communication delays. The Hypocenter Determination task, based solely on Eqn. 4.1, should be assigned to the remote server. This equation only considers processing time and communications overhead. It does not consider other aspects of the application environment. For example, not all seismic events contribute data to the velocity model. These include seismic events only detected by a few number of nodes (e.g. man made events) or events with a source outside the local vicinity of the volcano. Once the event hypocenter has been computed it can be used to identify these types of events. Performing this task on the base station allows the system to avoid using the high latency satellite link to transmit data which would not be used to update the model. This provides power savings, as the satellite station can be placed in sleep mode when the link is not being used.

Off-line task assignment using Eqn. 4.1 also does not take into account the dynamic nature of WSNs and the system lifetime constraints. For example, during an extended period of cloudy days, insufficient energy may be harvested from a solar panel to meet a node's system lifetime constraint. Under that condition a node may elect to shift a high energy task to another node having sufficient energy. This can be accomplished by incorporating the system lifetime ( $T_{discharge}$  calculated by Eqn. 4.4) into the task assignment process. If the system lifetime is less than a specified threshold, a decision can be made to shift one or more assigned tasks to another node or tier. Once a decision has been made to shift a task, the system records the current battery voltage as  $V_{decision}$ . When  $V_c + \tau > V_{decision}$ , the task will be shifted back to the node, where  $\tau$  is the desired amount the battery voltage should recover by.

We now illustrate the impact of our dynamic assignment scheme on the system lifetime using an experiment. Consider a node with two power states: active and sleep while performing sampling, STA/LTA and ARAIC. Sampling takes  $100 \mu s$  every 10 ms, STA/LTA takes 0.15 ms every 10 ms, and ARAIC takes 3.9 s whenever an event is detected. If an event is detected every 60 seconds,  $P_{active} = (100 \mu s)/10 \text{ ms} + 0.15 \text{ ms}/10 \text{ ms} + 3.9 \text{ s}/60 \text{ s} = 0.09$ . If  $I_{active}$  is 125 mA and  $I_{sleep}$  is 25 mA,  $I_{avg} = 25 \text{ mA} \cdot (1 - 0.09) + 125 \text{ mA} \cdot 0.09 = 34 \text{ mA}$ .

A 7Ah SLA battery discharging at 125 mA has an estimated life of 84.5 hours. With 34 mA discharge rate, the battery life is 523 hours. If the ARAIC processing task were assigned to another node,  $P$  would change from 0.09 to 0.031 (allowing 0.35 seconds to transmit the required data to an adjacent node), the average current would be lowered to 28.1 mA providing a battery life of 699 hours.

## 4.5 Deployments

Table 4.4: Key characteristics of the deployments

Deployment site	Time	Duration	Area of coverage	# of stations	Station type	Communication	Cost (US\$)
Tungurahua, Ecuador	Jul 2012	1 week	3 patches, 2 stations each, 200 m separation	6	Smart-phone based	None	2.1K
Llaima, Chile	Jan 2015	4 months	925 m $\times$ 475 m	16	Fig. 4.7(a)	XBee-based	6.4K
			33 km $\times$ 26 km	50	Traditional	None	572K

We deployed our systems on Tungurahua Volcano, Ecuador and Llaima Volcano, Chile (Fig. 4.10) to evaluate their operations in real volcano environments. These two deployment sites present different environments. Tungurahua is at a higher altitude, wetter, and with more vegetation. Llaima is at a lower altitude, with less vegetation, and significantly drier in the summer. Table 4.4 summarizes the characteristics of the two deployments. Each deployment had different evaluation goals as described in the subsequent sections.

### 4.5.1 Tungurahua Volcano Deployment

To test the first generation sensor nodes, we traveled to the Tungurahua Volcano near Baños Ecuador and deployed six nodes in July, 2012. This field trip afforded us an opportunity to test under real field conditions. It also allowed us to gain first-hand experience of the conditions we would face in future larger scale deployments. The primary goal was to test whether our sensing hardware can record seismic signal.



(a) Sensor node



(b) Base station

Figure 4.10: Llaima Volcano deployment, Chile, 2015.

The deployment faced several challenges. The first was the remote location. Only a few sites were accessible by road. Vast areas were only reachable via hiking on foot. Utilizing helicopters to assist during deployment was not feasible due to the altitude and the highly variability of the weather conditions. Teams must backpack all the equipment several kilometers to reach the deployment locations. Besides the distances covered, working at high altitude (12,000 to 14,000 feet) presented its own challenges. From the sensor design standpoint, care must be taken to minimize the package weight. It was important for the teams to be able to maximize the number of packages carried on each trip to minimize the number of trips. The wet weather conditions presented the second challenge. Between fog, that commonly covered the volcano, and rains challenged the equipment packaging.

#### 4.5.2 Llaima Volcano Deployment

The second deployment was on Llaima Volcano located near Melipeuco, Chile. The primary goal of this deployment was to test the new generation of hardware over a long duration period. We deployed 16 new-generation nodes in January 2015, in a 800 m by 1400 m patch. Fig. 4.11 shows

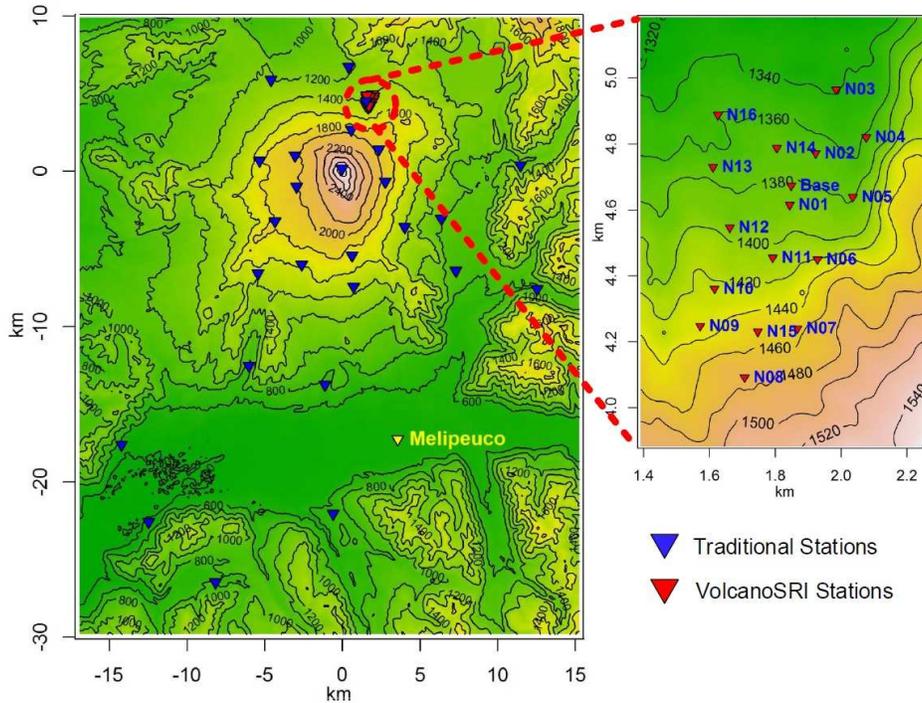


Figure 4.11: Node locations in Llaima Volcano deployment.

the deployment locations relative to the volcano summit and the terrain around the nodes. Llaima has an access road encircling it and the lower slopes can be reached using an off-road vehicle. With the varied terrain, it also provided a good field test of the node to node communications. The 16 nodes were deployed by three 2-person teams in one day. One team acted as a survey team establishing the final node locations. These were chosen based on topology to ensure that at least three other node locations were visible. Fig. 4.10(a) shows a deployed sensor node. A base station with satellite link, as shown in Fig. 4.10(b), was deployed. The system ran from January 10th to March 25th, 2015.

Together with our WSN nodes, 26 traditional seismic stations were deployed across a broad geographic area surrounding the volcano, as shown in Fig. 4.11. These traditional stations were installed over a period of two weeks by four multiple-person teams. The number of people in each deployment team depended upon how far the equipment had to be carried from the nearest road. Two stations were transported on horseback.

### **4.5.3 Deployment Lessons Learned**

We gained several important insights from the deployment. First, the communication range was longer than that observed during our initial testing. In the field, the signal strength from the link tests followed what was predicted by the path loss equations with consideration of terrain. Based on this, given topological information, it would be possible to estimate link quality during the planning phase of a deployment. This estimation is critical for larger scale deployment of 500 to 1,000 nodes.

Second, the antenna pole was fashioned out of two one-meter PVC pipe sections. In the field, it was discovered that the wind caused the pole to vibrate. This vibration can be transmitted to the geophone, introducing noise in the seismic signal. Heavier material should be used or guy wires attached to stabilize the poles. Based on the communication range results, an even simpler solution would be to use a single section of PVC pipe only. This would simplify deployment and be less likely to vibrate in the wind.

Third, assembling the solar panel frame required installation of several bolts. This proved difficult and increased the time to deploy a node. Altering the leg design would allow them to be installed prior to shipment. In the field, they could simply be unfolded, simplifying their installations.

For the Llaima Volcano deployment the 2-meter cable attached the geophone to the node was adequate. This cable should be increased to 6 meters to allow more flexibility in sensor placement. This would also place the sensor further from the antenna pole further reducing an noise generated by the pole vibrating in the wind.

## **4.6 Evaluation and Deployment Experiences**

This section presents the system evaluation and experiences during the Llaima Volcano deployment.

### 4.6.1 System Delay

To evaluate the computation overhead of the hypocenter computation, the algorithm was executed while varying the number of stations detecting the event. The resulting execution times are shown in Fig. 4.12 and is roughly linear with the number of stations. Extrapolating to 500 stations, the execution times for the base station increases to approximately 4 seconds while on the remote server (or cloud) to 0.3 seconds. To accurately compute the hypocenter it would not be necessary to utilize event information from all 500 stations. Only a small subset of events from geographically distributed stations are required. By limiting this subset to 10% of the nodes the execution time can be tuned to not exceed the task deadline. Tomography involves computing a velocity model using matrix inversion. The dimensions of this matrix is based on the resolution of the model, the number of stations and events. Increasing these results in more computation overhead as shown in Fig. 4.13.

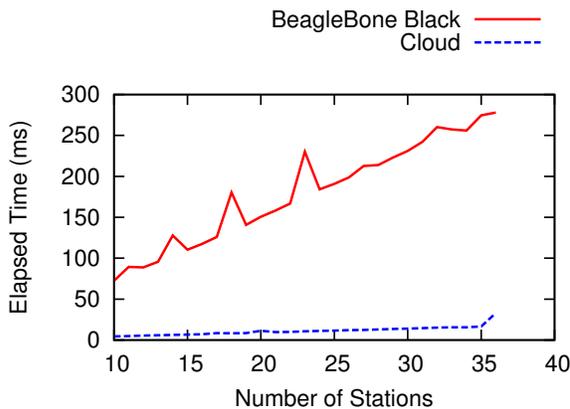


Figure 4.12: Hypocenter execution times by number of stations.

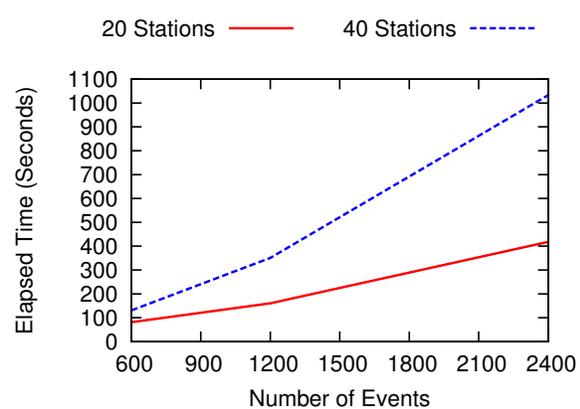


Figure 4.13: Tomography execution times by number of events.

Of the tasks executed by the sensor node, the ARAIC event detection task takes the longest time (3.9 seconds on a Gen 3 node). If this task were executed on the base station it would take only 6 ms. Using Eqn. 4.1, if the communication time to transmit the data ( $c_{i,i+1}$ ) plus the execution time on the node ( $t_i$ ) and base station ( $t_{i+1}$ ) is less than the 3.9 seconds then this task could be executed more efficiently on the base station. To evaluate this, a node repeatedly analyzed a simulated seismic signal that contained an event. Each time the STA/LTA event detection task detected an

event, the seismic signal was transmitted 1-hop to a basestation for detection using the ARAIC task. The ARAIC task requires 16 seconds of seismic data (1600 samples). Using a compressed format the required data could be transmitted in 55 99-byte packets and one 27-byte packet. To transmit, with acknowledgements, took 2.7 seconds with minimal processing overhead on the node. On the surface, it would appear to make sense to move the ARAIC task to the basestation. This is not the case when the network impact of having 500 nodes sending the seismic data over multiple hops to the basestation. This would saturate the network capacity. When the node executes the ARAIC task the average current consumption is 40 ma. When transmitting the seismic data to the basestation the average current consumption increases to 60 ma. Considering the battery capacity consumed, executing the ARAIC task on the node uses 156 msec (40 ma x 3.9 sec) versus executing on the basestation using 162 msec (60 sec X 2.7 sec). For the Gen 4 node, transmitting this seismic data takes the same amount of time while executing the task on the node takes only 206 ms. Thus for Gen 4 nodes, it is vastly more efficient to compute ARAIC task on the node.

#### **4.6.2 Data Fidelity**

There are two dimensions to assessing data fidelity. First, how accurately sensor samples are time-stamped. The GPS provides a PPS interrupt with a jitter of 10 ns. To achieve a sub-ms precision, we integrate Due's system clock and an internal 4- $\mu$ s resolution timer. Specifically, when the node receives the 1PPS interrupt, the system clock is resynchronized with the global time and the current value of the timer is saved. The time stamp of a seismic sample is the global time plus the difference between the timer's current and saved values. The timer's drift was measured against the PPS signal. Under normal temperature conditions, it drifts 16  $\mu$ s per second. The drift of the Real-Time Clock (RTC) in the Gen 4 node was also evaluated against GPS time. To evaluate this the node was allowed to run for a 32 hour period. At the end of this period the GPS time was compared with the RTC time. During this period the RTC had drifted 2.24 seconds. This was a 19.01 parts per million (PPM) drift. The RTC also has the capability to compensate for drift such as this. The required compensation factor required is calculated by the following formula [37]:

$$C = \text{int} \left( \frac{D_{ppm}}{0.1192} + 0.5 \right) \quad (4.5)$$

where  $D_{ppm}$  is the drift in Parts Per Million and  $C$  is the compensation factor. Positive compensation values result in the RTC running faster, while negative values slow the RTC. For the 19.01 PPM drift a compensation value of 160 was applied, the RTC synchronized with the GPS and allowed to run an additional 32 hours. This compensation value properly corrected for the RTC drift. By periodically monitoring the drift it is possible to correct for changes in the amount of drift caused by temperature changes.

Table 4.5: Shared SPI bus ADC Sample Intervals

Sample Interval (ms)	Number of samples
9	43
10	154237
11	53
34	1
Total	154334

Second factor impacting data fidelity is sampling frequency consistency. This is impacted by method used to communicate with the ADC. Two approaches for communicating with the ADC were evaluated. Both the ADC and the SD card utilize a Serial Peripheral Interface (SPI) bus for communication. The first approach, the ADC and the SD card shared the hardware SPI bus. With this approach care must be taken ensure the two devices do not interfere with each other when accessing the SPI bus. Since the ADC is accessed at interrupt level, interrupts must be disabled when accessing the SD card. This causes two side effects. First, the ADC data ready interrupt can be delayed resulting in an inconsistent sample rate. Second, the PPS interrupt can be delayed, causing clock skew. During a test for half an hour, the sampling intervals range from 9 ms to 34 ms as shown in Table 4.5. In the second approach, the ADC is linked to three GPIO pins with SPI signaling performed by software routines. This approach provided a consistent 10 ms sampling period.

The ability to detect weak seismic activity is determined by the sensitivity of the sensor as well as the noise level of the signal chain. The noise of the signal chain was measured for the Gen 3

node and two variations of the Gen 4 node. The first Gen 4 (Gen 4a) variation used exactly the same input amplifier as the Gen 3 node. The only difference being the board layout. The second Gen 4 (Gen 4b) variation used an entirely different amplifier design that incorporated a 6-pole low pass filter. To measure the noise of the seismic amplifier, the input of the amplifier was terminated with an input resistance equivalent to that of the seismic sensor. 10000 samples were collected using both a Gen 3 node and a both Gen 4 variations. The Root Mean Square value for these three sets of samples were computed and compared. Changing the sensor board layout (Gen 3 vs Gen 4a) resulted in a 10 db reduction in the noise level. Adding the input low pass filter (Gen 3 vs Gen 4b) resulted in a 16.8 db reduction in the noise level.

### 4.6.3 Communication Performance

Communication performance is affected by station spacing, antenna gains, transmitter power, and receiver sensitivity. Traditional WSNs rely on high density to ensure network connectivity. For volcano monitoring, geographic coverage is more important and necessitates wider spacing.

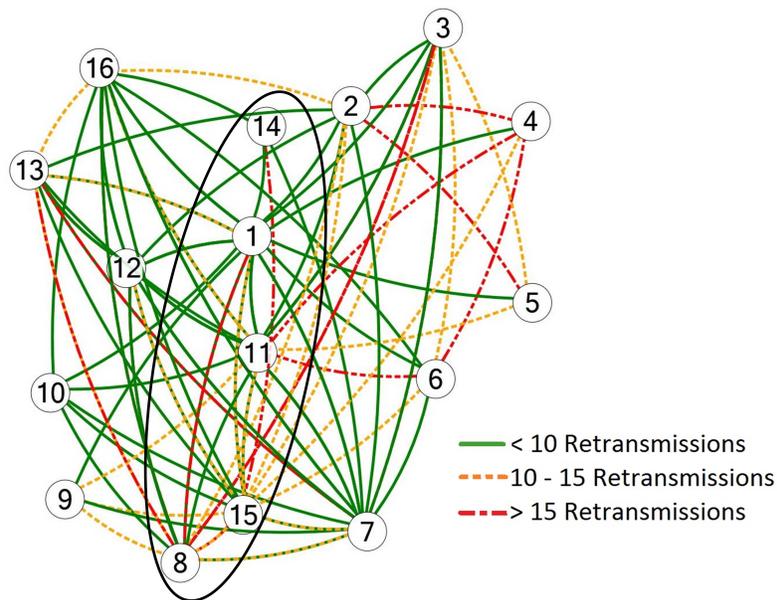


Figure 4.14: One-hop link quality (circles represent nodes).

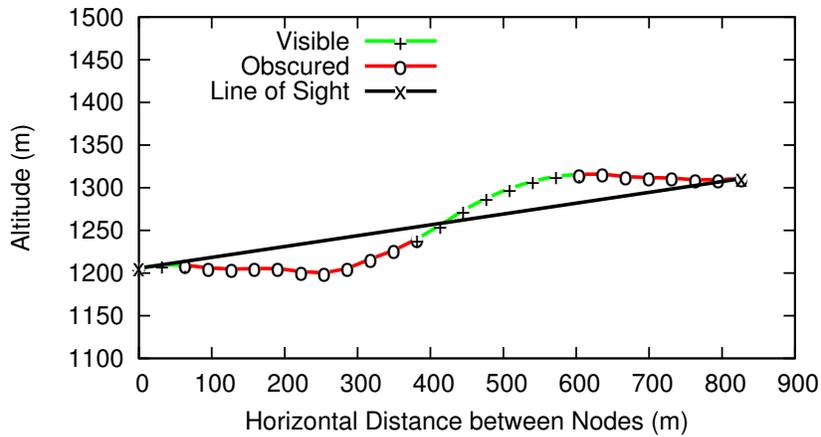


Figure 4.15: Line of Sight Path between Node 9 and Node 3

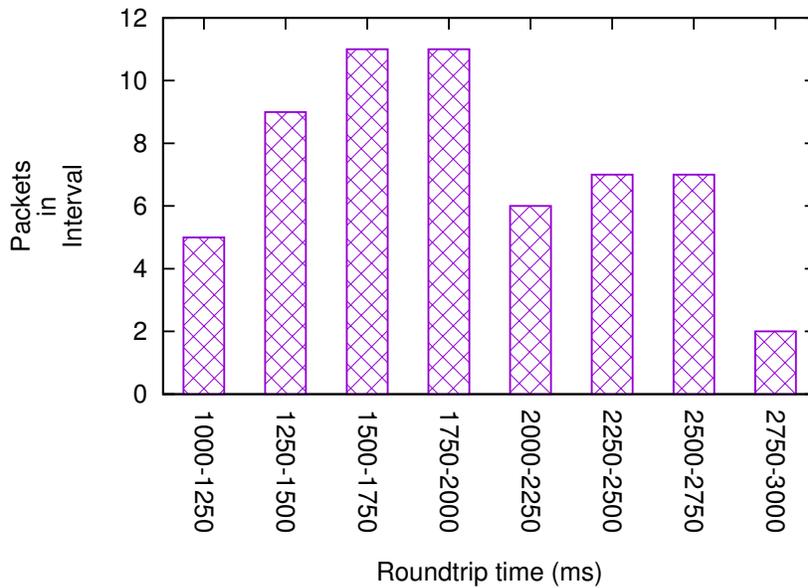


Figure 4.16: Distribution of Satellite Round Trip Ping Times

During the field deployment, the nodes periodically assessed link quality. For each assessment, a hundred 64-byte packets were transmitted between two nodes. Fig. 4.14 shows the link quality for single hop links. In our assessment, a link was considered “good” (shown with solid lines) if there were less than 10 retransmissions, “weak” (shown with dotted lines) if there were between 10 and 15 retransmissions, and “bad” (shown with dash-dot-dot lines) if there were between 15 and 20 retransmissions. Links with more than 20 retransmissions were not included in the figure. The link quality shown in Fig. 4.14 is consistent with the ground topology. There was a ridge running

roughly south to north that nodes 1, 8, 15, 11 and 14 were installed on, as indicated by the oval in Fig. 4.14. The south to north ridge that runs between these nodes As a result, nodes on either side of this ridge did not have a line-of-sight path to each other. The topographic profile between nodes 9 and 3, Fig. 4.15, illustrates this ridge and how it blocks the line of sight between these nodes. There was also a rise between nodes 9/10 and 13/16 which blocked the line of sight between these two groups. To provide a network path between these two groups, Node 12 installed on the rise to provide an additional network link between the two groups. Node 5 was also blocked from several nodes by intermediate rises.

The base station communicated with the remote database server through a satellite link. To assess the performance of this link, the base station periodically used the ping command to determine the round trip times. The test showed a median round trip time of 1849 ms. Fig. 4.16 shows the distribution of these round trip times.

The satellite communication overhead of two different protocols was also assessed. The first used a simple web service interface based on HTTPS Post requests. While easy to implement and test, it is a heavy-weight protocol. For example, 962 events generated 5.85 MB of upload traffic and 8.227 MB download. Typically, satellite links have data packages with a limited amount of data which can be transmitted each month. This level of traffic quickly exceeded our 2 MB per-month data limit. The second protocol sends binary encoded messages over a TCP/IP connection. This approach generated less than 100 KB of upload traffic for the same number of event messages.

#### **4.6.4 Battery and Solar Panel Performance**

Our design uses a 7Ah SLA battery recharged using a 20 watt solar panel. Fig. 4.17 shows a 24-hour discharge/charge cycle for this design during a full-sun daylight period. As shown in the figure, after discharging during the night period, the battery was completely recharged during the next daylight period. Analysis shows that the battery discharged at a rate of 0.0282 volts per hour during the dark period. At this rate, it takes 88.6 hours (3.5 days) to discharge the battery to 10.5 volts, similar to the result obtained through Peukert's Law. This was confirmed during the

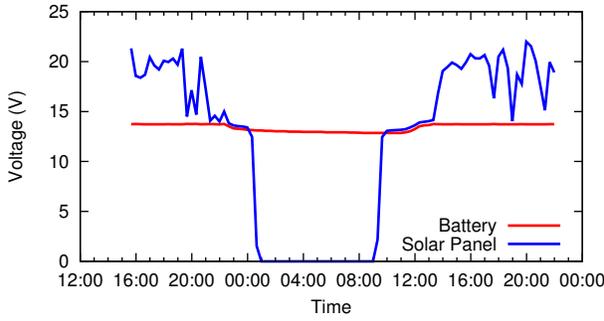


Figure 4.17: Battery daily charging cycle.

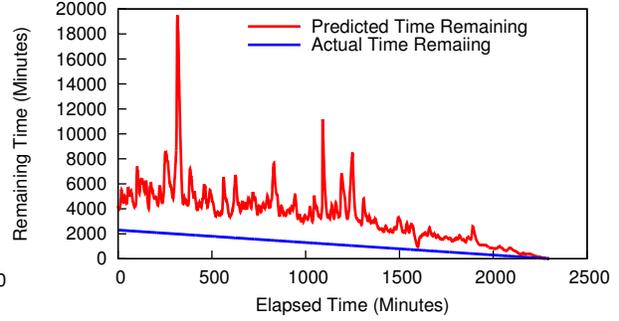


Figure 4.18: System Life Time Estimation

field deployment when nodes operated for four days under heavy cloud cover. Post deployment analysis determined 75% of the batteries would not accept a charge due to repeated cycles where the batteries discharged below 10 volts. To resolve this, a new power control circuit was designed to disconnect the node from the battery when discharged to 10 volts and reconnect when recharged to 11 volts.

To evaluate predicting system life time using Eqn. 4.4, a node was allowed to operate processing a simulated seismic signal for several days. The simulated seismic signal generated an event every 16 seconds that was transmitted to a base station. While operating the battery voltage sampled once per second. These reading were then passed through an exponential low-pass filter with a period of 11 minutes ( $a = 0.985$ ) to compute long term average. The change in battery voltage ( $\Delta t/\Delta v$ ) was then computed the over the same 11 minute period updating this estimage every minute. The amount of time for the battery to discharge from its current voltage to the terminal discharge voltage ( $V_D$ ) of 10.5 volts was predicted using Eqn. 4.4. Fig. 4.18 displays the predictions between the battery fully charged voltage ( $V_T = 12.5\text{volts}$ ) to the terminal discharge voltage. As can be seen from this figure this approach in general over estimates system life time and is very sensitive to minor changes in the battery discharge rate as evidenced by the large spikes in the estimated life time particularly at approximately the 300 minute point.

#### **4.6.5 Packaging and Ease of Deployment**

To deploy a node, the antenna pole is driven into the ground and attached the top section. Once completed, the Pelican case housing the sensor electronics snaps into a metal bracket. Next, the GPS antenna and radio antenna are attached to the sensor node. The geophone is buried in a hole dug nearby and the cable is plugged into the sensor node. The most time consuming task is to assemble the solar panel leg frames using several small bolts. Once these are attached, the solar panel and the battery box can be positioned and cabled to the sensor node. Zip ties secure the cables in place prior to powering on the node. Nodes were placed approximately 200 meters apart across the lower slope on the north side of the volcano. Two people could deploy a node in approximately 20 minutes including the time to hike between locations.

To assess the waterproof nature of the sensor and battery boxes, they were subjected to three days torrential rain. During the four months of deployment, the boxes remained dry inside with no failure of the electrical connections. While the weather condition during the deployment was generally dry, moisture can condense on the equipment during the night.

#### **4.7 Conclusion**

This paper presents the development of a seismic monitoring system supporting data-intensive applications such as volcano tomography. By utilizing processing tiers with varying computing capacities and a task assignment approach, low-cost sensors can support such applications. Through our two deployments, we have shown that low-cost hardware can be used to record seismic events in the immediate vicinity of a volcano. Moreover, we learned lessons and identified subtle but important modifications to the system, which ensure the success of future deployments. Our Chile deployment identified the importance of minimizing mechanical noise sources and ensuring that sensor cables are long enough to allow the flexibility needed for optimal sensor placement. Communication measurements showed that the use of a wireless mesh network over varied terrain is a viable communication medium, even over longer distances than originally expected.

## CHAPTER 5

### CONCLUSION

Designing and deploying a multi-node heterogeneous wireless sensor network presents a number of challenges. These include:

**Event Detection and Sensing Accuracy** While environmental changes can be detected using relatively simple algorithms and hardware it is not always easy to assign them to a specific source. For example in Supero, simple light and sound detectors are able to detect the operation of a number of lights and appliances. To automatically assign the detection event to a specific light or appliance it was necessary to utilize application specific knowledge (power decay of light) and a priori knowledge (distance of the lights from the sensors). By correlating the detection events across multiple sensing modes it was possible to eliminate the majority of the false alarms while being able to estimate per-appliance energy consumption with an average error of less than 7.5%.

**Controlling Power Consumption and Operational Lifetime** For systems which must operate unattended for long periods of time, controlling power consumption is a major consideration. This is commonly handled by powering-down portions of the system when not in use. For some applications this is handled by using the lowest sensor sampling rate the application can tolerate and while providing the required sensing accuracy. This was particularly a challenge for the acoustic sensing performed by Supero. Supero utilized a two-phase sampling approach based on the observation that for large portions of the time the environment is relatively quiet. During these quiet periods Supero utilized a very slow sampling rate to conserve power. Once an increase in the acoustic background sound was detected it switch to a higher sampling rate to provide the necessary acoustic features needed to identify the appliance.

The seismic sensing system utilizes a similar approach in that it was able to power down

major components, such as the GPS, and place the processor into lower power states between samples. To allow this to occur, it was necessary to utilize additional features of the more modern processors such as built-in real-time clocks. This system also utilized power harvesting from solar panels to recharge its batteries. This was needed due to the extended operational lifetime and the inaccessibility of the nodes once deployed.

**Assigning Processing Tasks to Functional Units** In a WSN utilizing a heterogeneous mixture of nodes with varying levels of computing resources, careful consideration must be given to which node (functional unit) a processing task is assigned. When assigning a task, the designer must consider the amount of data required by the task, processing resources needed to perform the task, communication overhead and bandwidth available to move the required data from one functional unit to another, and any real-time processing deadlines imposed by the application. For Supero, the sensing nodes needed to be small so as to be unobtrusive as possible once installed. This resulted in nodes running on small batteries (two AA batteries) utilizing very power efficient processors. In addition, the appliance detection algorithms required event information from all the sensing nodes as well as significant computational resources to execute the algorithms. The combination of these two factors determined the partitioning of the processing tasks with the sensing nodes only executing simple event detection algorithm as well as feature extraction while the base station performed the remaining tasks.

The sensing applications composed of a wider variety of processing tasks require a more quantitative analysis for assigning the tasks. As part of the seismic sensing system such an approach was developed. This approach, quantified the computational/communication/data resources required for each processing task and developed a decision criteria for assigning a task to a particular functional unit based on the application processing deadlines. This approach was then utilized to assign the various tasks to a functional unit, in this case a tier in the system's hierarchical architecture.

**Ease of Deployment** As sensing networks are more widely deployed they will be installed by individuals who do not have specific knowledge or expertise related to the network. They will be more than likely be just ordinary home owners or experts in some other subject area, such as geologists. As a result, it will be necessary for the deployment to be as simple yet robust as possible. For example with both Supero and the seismic sensing network very simple deployment instructions/guidelines were provided. In the case of Supero, instructions were “place a light sensor with unobstructed view to the light” and “place an acoustic sensor on top of the microwave.” For the seismic sensing network, the guidelines were “deploy the nodes approximately 200 to 300 meters apart in a location where 2 or 3 other nodes are visible”. In both cases, the systems were able to be successfully deployed by individuals with little or no expertise with sensor networks.

The experiences from deploying the wireless sensor network resulted in several lessons learned. The first lesson learned early on in the development of Supero involved node reliability. The small, embedded system sensing nodes such as the Telosb and IRIS motes are difficult to identify and debug system problems. The general lack of modern debugging tools relegates a designer to using print statements to determine what code paths are being taken during operation. These provide little help when the node locks up and completely stops functioning. While the Arduino class of processors provide greater functionality they still lack modern debugging tools. In both cases, a JTAG hardware debugging capability maybe supported it is difficult to use, requires special additional hardware, and may not be easily accessible on a commercially produced processor board. For example, for the Teensy 3.6 processor utilized in Generation 4 of the seismic sensor node, utilizing the JTAG associated functionality requires modifications to its circuit board. Once installed on the main sensor board the JTAG connections are no longer accessible.

Another lesson learned was related to the importance of rapid prototyping and multiple small field deployments when developing sensor hardware. While careful design can eliminate a number of hardware related issues, some can not be identified until actual hardware has been produced and attempted to be used in an actual sensing application. It is only by evaluating actual hardware

can subtle interactions between components be identified. Two such cases arose with the seismic sensing nodes. The first was related to SPI bus contention which caused erratic sensor sampling rates. The second was related to noise in the sensor input amplifier caused by interaction between the analog and digital circuits located on the main sensor board.

The importance of field experience associated with deploying a sensor network can not be discounted. In the case of the seismic sensor network the field deployments identified several areas that needed to be improved. During the first deployment, this identified a packaging issue that needed to be changed. The initial boxes were not sturdy enough, did not seal properly in some cases, to difficult to open in the field to service a node, and were too bulky for multiple units to be carried by one person. This resulted in the development of an alternative packaging approach that was tested during the second development.

Prior to the second deployment, it was believed the solar panel support frames required a wide range of adjustability to account for the angle of the sun. Having this adjustability in fact was not utilized in the field and increased the difficulty associated with transporting and deploying the nodes. By redesigning the frames with less adjustability in mind, the frames could be attached before transporting the panel to the deployment site and shortened the time to deploy a node by approximately 10 minutes. While 10 minutes might not seem like a long period of time, it is when you must deploy 15 to 25 nodes in a day in the rain with it very cold. Under those conditions every minute counts. As a result, performing multiple deployments, even those which might have limited scope, is highly recommended for an designer of a sensor network for there are some things that can only be identified once a system is out in the field.

The system developed can also be applied to other sensing domains. For example, a current area of study within the geophysical community is low frequency sound propagation at high altitudes. For studies in this area, infrasound sensors are sent aloft in a high altitude balloon and used to record the sounds. As the balloon proceeds on its flight path GPS coordinates, altitude, and course information is recorded. In 2015, a Generation 3 node was included on such a flight with the seismic sensor replace with an infrasound sensor. [4]

Another application area is animal tracking. For this application, a camera and motion sensor are added to the sensing nodes. When movement is detected the node captures a picture then applies an image recognition algorithm to identify the source of movement. If the source is one of the types of animals being tracked, the WSN is used to transmit the detection event to a central location so that additional tracking resources can be dispatched to observe the animal.

In both of these additional application areas, field experience obtained during trial deployments is critical to the successful final deployment. It is only through a trial deployment can one be assured that the nodes and overall system will operating in its final environment. In the case of the high altitude balloon flight, it is difficult to simulate the environmental conditions experienced by the system. How will the system respond to the effects of high altitude operation or the extreme changes in temperature? For the animal tracking, how will animals, which are curious by nature, react to seeing something new (a sensor node with a solar panel) in their environment. Will they react in a way that introduces a system failure? Such questions can only be answered through field experience.

## **BIBLIOGRAPHY**

## BIBLIOGRAPHY

- [1] Alertme. Alertme, August 2015.
- [2] American Coalition for Clean Coal Electricity. Study finds families burdened by ever-increasing energy costs, 2011.
- [3] Rajesh Krishna Balan, Mahadev Satyanarayanan, So Young Park, and Tadashi Okoshi. Tactics-based remote execution for mobile computing. In *MobiSys*, 2003.
- [4] D. C. Bowman, C. S. Johnson, R. A. Gupta, J. Anderson, J. M. Lees, D. P. Drob, and D. Phillips. High Altitude Infrasound Measurements using Balloon-Borne Arrays. *AGU Fall Meeting Abstracts*, pages S54B–06, December 2015.
- [5] Eduardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. Maui: making smartphones last longer with code offload. In *MobiSys*, 2010.
- [6] The Energy Detective. The energy detective, August 2015.
- [7] Shannon Doocy, Amy Daniels, Shayna Dooling, and Yuri Gorokhovich. The human impact of volcanoes: a historical review of events 1900-2009 and systematic literature review. *PLoS currents*, 5, 2013.
- [8] Steven Drenker and Ab Kader. Nonintrusive monitoring of electric loads. *IEEE Computer Applications in Power*, 12(4):47–51, 1999.
- [9] E.T. Endo and T. Murray. Real-time seismic amplitude measurement (RSAM): a volcano monitoring and prediction tool. *Bulletin of Volcanology*, 53(7), 1991.
- [10] Linda Farinaccio and Radu Zmeureanu. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy and Buildings*, 30(3):245–259, 1999.
- [11] Matthew Faulkner, Michael Olson, Rishi Chandy, Jonathan Krause, K Mani Chandy, and Andreas Krause. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 13–24. IEEE, 2011.
- [12] Jason Flinn, SoYoung Park, and M. Satyanarayanan. Balancing performance, energy, and quality in pervasive computing. In *ICDCS*, 2002.
- [13] Scott W French and Barbara Romanowicz. Broad plumes rooted at the base of the earth’s mantle beneath major hotspots. *Nature*, 525(7567):95–99, 2015.
- [14] Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. Electrisense: single-point sensing using emi for electrical event detection and classification in the home. In *12th ACM International Conference on Ubiquitous Computing (UbiComp)*, pages 139–148, 2010.

- [15] George W. Hart. Nonintrusive appliance load monitoring. *Proceedings of IEEE*, 80(12):1870–1891, 1992.
- [16] Bo-Jhang Ho, Hsin-Liu Cindy Kao, Nan-Chen. Chen, Chuang-Wen You, Hao-Hua Chu, and Ming-Syan Chen. Heatprobe: a thermal-based power meter for accounting disaggregated electricity usage. In *13th International Conference on Ubiquitous Computing (UbiComp)*, pages 55–64, 2011.
- [17] Insteon. Insteon, August 2015.
- [18] Xiaofan Jiang, Stephen Dawson-Haggerty, Prabal Dutta, and David Culler. Design and implementation of a high-fidelity ac metering network. In *The 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 253–264, 2009.
- [19] Xiaofan Jiang, Minh Van Ly, Jay Taneja, Prabal Dutta, and David Culler. Experiences with a high-fidelity wireless building energy auditing network. In *The 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 113–126, 2009.
- [20] Deokwoo Jung and Andreas Savvides. Estimating building consumption breakdowns using on/off state sensing and incremental sub-meter deployment. In *The 8th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 225–238, 2010.
- [21] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 254–263. IEEE, 2007.
- [22] Younghun Kim, Thomas Schmid, Zainul M. Charbiwala, and Mani B. Srivastava. Viridiscop: design and implementation of a fine grained power monitoring system for homes. In *The 11th International Conference on Ubiquitous Computing (UbiComp)*, pages 245–254, 2009.
- [23] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, and Mark Yarvis. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 64–75. ACM, 2005.
- [24] J.M. Lees. The magma system of mount st. helens: non-linear high-resolution p-wave tomography. *Journal of volcanology and geothermal research*, 53(1-4):103–116, 1992.
- [25] Jonathan M Lees. Seismic tomography of magmatic systems. *Journal of Volcanology and Geothermal Research*, 167(1):37–56, 2007.
- [26] Liqun Li, Guoliang Xing, Limin Sun, Wei Huangfu, Ruogu Zhou, and Hongsong Zhu. Exploiting fm radio data system for adaptive clock calibration in sensor networks. In *The 9th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 169–182, 2011.
- [27] Memsic Corp. Iris datasheet, 2011.

- [28] Memsic Corp. TelosB datasheet, 2011.
- [29] Mohammad-Mahdi Moazzami, Dennis E Phillips, Rui Tan, and Guoliang Xing. Orbit: a smartphone-based platform for data-intensive embedded sensing applications. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, pages 83–94. ACM, 2015.
- [30] Kim Munro. Automatic event detection and picking of p-wave arrivals. *CREWES Report*, 2004.
- [31] Ryan Newton, Sivan Toledo, Lewis Girod, Hari Balakrishnan, and Samuel Madden. Wishbone: Profile-based partitioning for sensornet applications. In *NSDI*, 2009.
- [32] P3 International Corp. P4400 Kill A Watt™ Operation Manual, 2012.
- [33] Jeongyeup Paek, Krishna Chintalapudi, John Caffrey, Ramesh Govindan, and Sami Masri. A wireless sensor network for structural health monitoring: Performance and experience. *Center for Embedded Network Sensing*, 2005.
- [34] Shwetak N. Patel, Sidhant Gupta, and Matthew S. Reynolds. The design and evaluation of an end-user-deployable, whole house, contactless power consumption sensor. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 2471–2480, 2010.
- [35] Shwetak N. Patel, Thomas Robertson, Julie A. Kientz, Matthew S. Reynolds, and Gregory D. Abowd. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. In *The 10th International Conference on Ubiquitous Computing (UbiComp)*, pages 271–288, 2007.
- [36] D. E. Phillips, R. Tan, M. M. Moazzami, G. Xing, J. Chen, and D. K. Y. Yau. Supero: A sensor system for unsupervised residential power usage monitoring. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 66–75, March 2013.
- [37] prjc.com. Teensyduino, version 1.40 source code, 2017.
- [38] Moo-Ryong Ra, Bin Liu, Tom F. La Porta, and Ramesh Govindan. Medusa: a programming framework for crowd-sensing applications. In *MobiSys*, 2012.
- [39] Alberto Rosi, Matteo Berti, Nicola Bicocchi, Gabriella Castelli, Alessandro Corsini, Marco Mamei, and Franco Zambonelli. Landslide monitoring with sensor networks; experiences and lessons learnt from a real-world deployment. *Int. J. Sen. Netw.*, 10(3):111–122, August 2011.
- [40] Ritei Shibata. Selection of the order of an autoregressive model by akaike’s information criterion. *Biometrika*, 63(1):117–126, 1976.
- [41] R. Sleeman and T. van Eck. Robust automatic p-phase picking: an on-line implementation in the analysis of broadband seismogram recordings. *Physics of the earth and planetary interiors*, 113, 1999.

- [42] Wen-Zhan Song, Renjie Huang, Mingsen Xu, Andy Ma, Behrooz Shirazi, and Richard LaHusen. Air-dropped sensor network for real-time high-fidelity volcano monitoring. In *Proceedings of the 7th international conference on Mobile systems, applications, and services, MobiSys '09*, pages 305–318, New York, NY, USA, 2009. ACM.
- [43] Jacob Sorber, Nilanjan Banerjee, Mark D. Corner, and Sami Rollins. Turducken: Hierarchical power management for mobile devices. In *MobiSys*, 2005.
- [44] Ivan Stoianov, Lama Nachman, Sam Madden, and Timur Tokmouline. Pipeneta wireless sensor network for pipeline monitoring. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks, IPSN '07*, pages 264–273, New York, NY, USA, 2007. ACM.
- [45] Z. Cihan Taysi, M. Amac Guvensan, and Tommaso Melodia. Tinyyears: spying on house appliances with audio sensor nodes. In *The 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, pages 31–36, 2010.
- [46] TPCDB. The power consumption database, August 2015.
- [47] A Trnkoczy. Understanding and parameter setting of sta/lta trigger algorithm. *IASPEI New Manual of Seismological Observatory Practice*, 2:1–19, 2002.
- [48] U.S. DoE. Annual energy outlook, 2006.
- [49] U.S. Energy Information Administration. Residential energy consumption survey, 2011. <http://www.eia.gov>.
- [50] Agustín Udías Vallina. *Principles of seismology*. Cambridge Univ. Press, 1999.
- [51] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh. Deploying a wireless sensor network on an active volcano. *Internet Computing, IEEE*, 10(2):18–25, March 2006.
- [52] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. A wireless sensor network for structural monitoring. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys '04*, pages 13–24, New York, NY, USA, 2004. ACM.