Experiences in LP-IoT: EnviSense Deployment of Remotely Reprogrammable Environmental Sensors

Reese Grimsley
Carnegie Mellon University
Pittsburgh, PA, USA
reeseg@cmu.edu

Mathieu Marineau
United States Geological Survey
Sacramento, CA, USA

Bob Iannucci
Carnegie Mellon University – Silicon Valley
Moffett Field, CA, USA

ABSTRACT

The advent of Low Power Wide Area Networks (LPWAN) has improved the feasibility of wireless sensor networks for environmental sensing across wide areas. We have built EnviSense, an ultra-low power environmental sensing system, and deployed over a dozen of them across two locations in Northern California for hydrological monitoring applications with the U.S. Geological Survey (USGS). This paper details our experiences with the design and implementation of this system across two years, including six months of continuous measurement in the field. We describe the lessons learned for deployment planning, remote device management and programming, and system co-design with a domain-expert from the USGS.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; *Embedded systems.*

KEYWORDS

Environmental Sensing, Sensor Networks, LPWAN, LoRa, Hydrology, Experience

ACM Reference Format:
Reese Grimsley, Mathieu Marineau, and Bob Iannucci. 2022. Experiences in LP-IoT: EnviSense Deployment of Remotely Reprogrammable Environmental Sensors. In *1st ACM Workshop on No Power and Low Power Internet-of-Things (LP-IoT '21), January 31–February 4, 2022, New Orleans, LA, USA.* ACM, New Orleans, LA, USA, 7 pages. https://doi.org/10.1145/3477085.3478988

1 INTRODUCTION

Low-Power Internet of Things (IoT) has reached broad appeal in recent years due to technological improvements, such as Low-Power Wide Area Networks (LPWAN), microprocessors with good performance to power ratios, and extensive marketing for sensor network solutions. Experts from fields such as agriculture, meteorology, and hydrology have realized the value of widely distributed sensor networks that operate for months or years on battery power or energy harvesting.

LP-IoT '21, January 31–February 4, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-8702-6/22/01...$15.00
https://doi.org/10.1145/3477085.3478988

Sensing networks built from small, cheap sensors enable data collection at high spatial resolution, providing a better picture of the measured quantities, such as stream heights across many small tributaries of a river basin. A higher number of devices also increases the quality of aggregates, like the mean change in stream height of streams or rivers throughout a storm, as faulty individual sensors constitute a small part of that aggregate. High temporal resolution is similarly valuable, encompassing both the sampling rate and latency, i.e. the time between sensing the environment and receiving those values in a server or database. Data capture at higher rates provides more detailed information about the changes in environment. Latencies on the order of seconds or minutes enable reactive applications such as disaster warning systems.

The U.S. Geological Survey (USGS)[1] has an interest in exploring the application of newer technologies, such as LoRaWAN, a Low-Power Wide Area Network, to improve their ability to gather hydrological data. In many river and lakes, stream height and stream flow are collected for water supply, ecological, or flood monitoring purposes. Low latency data retrieval, often referred to as "real-time data" is particularly useful for early flood warning. Remotely reconfigurable data collection rates would let the sensor collect detailed information during such an event, yet use a lower sampling rate during, for example, a long dry period.

Data collected by environmental sensors are typically stored onsite using "self-contained" loggers or telemetered in real-time using one of several backhaul methods (satellite and cellular are the most common). Drawbacks of self-contained loggers include additional time and expense required to retrieve and archive data; potential loss of data due to vandalism or equipment failure; and lack of real-time awareness of conditions. While conventional telemetry technologies can overcome those drawbacks, they come at a much higher equipment and installation cost and require a larger installation footprint for equipment and a power supply. Further, two-way communications is limited to only certain types of technologies.

We jointly developed the EnviSense system with the USGS, improving the ability to collect real-time environmental data by using technology that is lower cost and has a smaller footprint than conventional telemetry equipment. This system has been deployed with over a dozen sensors across two locations in Northern California. **The contributions of our system design and the salient lessons learned are as follows:**

- A full vertical stack environmental sensing system with ultra-low power LoRaWAN data logger design and server infrastructure for secure networking, LoRa gateway health, and data visualization.

[1] Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

- The importance of deployment planning and RF surveying to select sensor and gateway location(s) and equipment (antennas, mounting hardware) prior to deployment.
- The importance of firmware design that is robust to server and network failures while maintaining low-power operation.
- The importance of software interfaces for remote application-level updates of data plane functionality to avoid modifying firmware or visiting deployment sites.

2 SYSTEM GOALS

The EnviSense project was initiated to enable lower-cost stream monitoring for a variety of purposes. Initial testing was part of an effort to build a real-time flood monitoring system at Beale Air Force Base, near Marysville, California, USA. This water observation system was defined with the following design objectives:

- The sensors will measure the water level in streams, but should also measure device health qualities such as internal temperature, humidity, and barometric pressure to establish environmental context.
- Sensors must last for several years on a single battery without requiring energy harvesting via solar or otherwise due to heavy brush around installation sites.
- Sensors must uplink within tens of seconds of sampling to ensure the data's usefulness for time-sensitive action, *e.g.,* flood warnings.
- Sensor interfaces must be compatible with other transducers used by the USGS so they may be fitted to EnviSense data loggers without hardware revisions for use in other deployments and applications.
- Sensors must be robust to communication failures at the gateway or server level, such that they will resume operation once the communication channels are repaired, and should continue collecting and logging data in the meantime.
- Sensors must be remotely configurable, such that sampling rates and the qualities to measure can be updated without visiting the remote installation sites.

3 SYSTEM ARCHITECTURE

The EnviSense system architecture is shown in Figure 1.

EnviSense data loggers measure the water level of streams and other environmental qualities like temperature and barometric pressure according to a locally stored schedule that may be wirelessly updated at runtime. Sensed data, including device health information like battery voltage and capacity, are uplinked over a LoRaWAN connection to a MultiTech IP67 LoRaWAN gateway [3] situated at a central location with high elevation to extend network coverage.

This gateway maintains an internet backhaul, such as a cellular connection, and a secure OpenVPN tunnel to the USGS server, which hosts the LoRaWAN network server that performs MAC-in-the-Cloud. This service, ChirpStack [9], provides over-the-air authentication (OTAA) for EnviSense devices. When an authenticated device uplinks data to the network server, it is forwarded to a custom-built application server.

The application server manages the data and control layers for the deployed sensors, which most often means entering sensor data

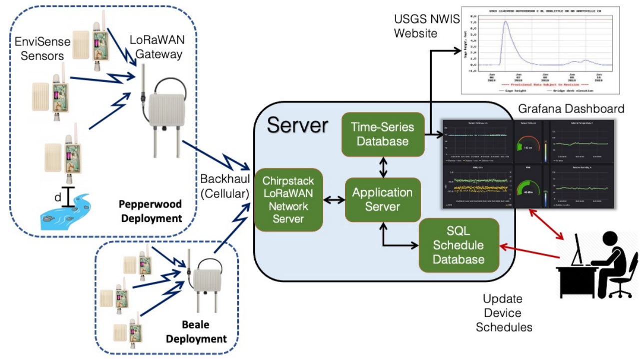

Figure 1: EnviSense System Architecture

into the time-series database, InfluxDB. Measurements in InfluxDB may be queried and displayed in a web dashboard, Grafana [2] for visualization. The application server also handles the control layer by retrieving pertinent information as the devices join the network, including the current time and a new schedule of sensing commands, the latter of which is discussed in Section 5.4.

The user, such as USGS personnel, can interact with the system in two ways. Through the Grafana dashboard, they can query the time-series database for a specific range of time. The data is displayed across an array of user-defined panels, showing the current and historical measurements with plots, gages, heatmaps, etc., and there is an application programming interface (API) to define more custom visualizations. The user can also update device data-collection schedules within a SQL database to reprogram the devices' sensing behavior. Collected measurements are also sent to the USGS National Water Information System (NWIS) [13].

4 ENVISENSE DATA LOGGERS

The EnviSense data logger was custom designed by our team at CMU. It uses a ATSAM3X8E Cortex-M3 microcontroller with 96 kB SRAM and 512 kB of FLASH storage. It contains sensors for temperature, humidity, acceleration (for orientation and tilt), barometric pressure, and battery health (voltage and current monitor). The main parameter measured is water level using a non-contact ultrasonic range sensor or a vented pressure transducer. We used an ultrasonic range sensor from MaxBotix [4] to measures distance between 0.5 and 5 m or 1 and 10 m at cm resolution, depending on the model. Distance is converted to water level based on the height of the sensor from the stream bed. The pressure transducers measure submerged sensor depth and use SDI-12 [7], which is a serial communications protocol commonly used in environmental sensors. We use a Semtech SX1276 LoRa [6] radio to wirelessly communicate with the rest of the system. EnviSense also includes a GPS module for localization and precision time-synchronization, but is rarely used in the current application aside from RF surveying (Section 5.1). All transducers and radios can be shut off in hardware using load-switches and control signals from the processor; this is essential for attaining a low power sleep state consuming 75 µA.

The device may be powered by any battery that has a JST-2 connector or a D-cell form factor and that provides more than 2.5 Volts. For our deployments, we use a 6600 mAh Lithium Polymer battery

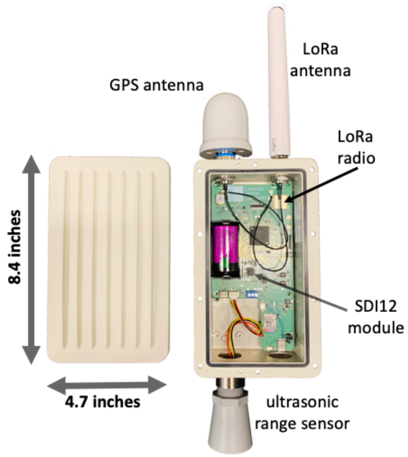

Figure 2: EnviSense data logger

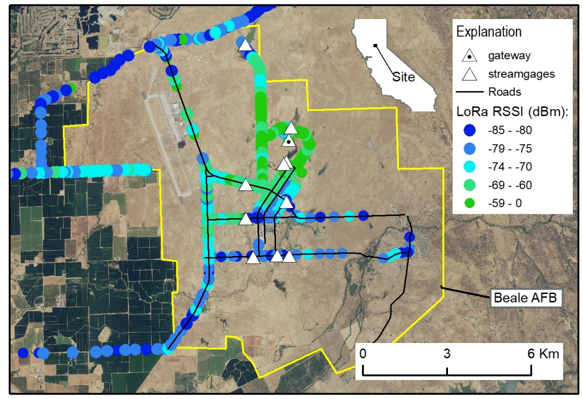

Figure 3: RF coverage survey at Beale Air Force Base. Circles show RSSI measurements, and triangles show the locations of deployed EnviSense devices and the gateway.

pack that nominally supplies 3.7 V. As the battery is nearly empty, the voltage will drop; however, the buck-boost voltage regulator will continue to supply 3.3 V to the rest of the device until the battery voltage drops below 2.5 V.

All hardware is encased in an IP67-rated enclosure as shown in Figure 2, including four ports for the LoRa antenna, GPS antenna, ultrasonic or SDI-12 sensor, and device management. The device management port has a clear removable cap that provides easy access to the power switch, reset button, and SD-card as well as visibility of the internal LEDs.

Our firmware implementation is based on Arduino Dué [1] for low-level drivers. The firmware executes a finite state machine, in which most states relate to communication with the application server to control what information the device requests or responds to. The main state at runtime processes the application schedule (Section 5.4), which will sample from sensors, put values into an outbound data buffer, and uplink to the server over LoRa.

5 EXPERIENCES AND LESSONS

Throughout several iterations of design and deployment, we have learned much about building Low Power IoT systems for remote sensing applications. The notable challenges have been managing and reconfiguring deployed sensors without incurring the overhead of physically visiting each of them nor overusing the LoRa radio, which has a high energy cost. Due to the remote nature of these systems, it is in our interest to seldom visit, and this will become more necessary as EnviSense is scaled to larger deployments.

5.1 Radio Signal Coverage Survey

LoRaWAN as a Low-Power, Wide Area Networking (LPWAN) technology is useful for providing coverage in sensor networks at low cost and low power. An LPWAN requires significantly less infrastructure and fewer devices than 802.15.4-based sensor networking solutions configured in a mesh topology (e.g. ZigBee). This reduces cost and complexity, albeit at the cost of communication throughput, *i.e.,* a few kbps in LoRa *vs.* hundreds of kbps with 802.15.4.

LoRaWAN uses unlicensed 903-928 MHz frequency bands in the U.S.A., and requires that we install our own gateway to provide coverage to our deployed data loggers. Due to the topography of the region, *i.e.,* rolling hills, and the locations of streams, *i.e.,* valleys and ravines, selecting the right sites for gateways and sensors is nontrivial from a network planning perspective. For example, in the first deployment attempt, one sensor site was not properly surveyed, leading to several hours of lost effort because a large hill obstructed the signal path to gateway, rendering the site unusable. There exist tools [5, 10] to generate estimates of the wireless link budget, *i.e.,* the acceptable loss in the signal's energy between transmitter and receiver, but it is useful to collect real data using tools that will compose the real system. Performing an Radio Signal or "RF" Survey is recommended prior to deploying sensors and gateways to ensure that the two devices will reliably communicate with each other.

To perform a survey, a gateway should be set up at a prospective installation location, such as a hill or rooftop, with an omnidirectional antenna on a tall mast (5-20 meters). This base-station will communicate with survey devices to measure the signal quality, *i.e.,* the received signal strength indicator (RSSI), as the devices are carried between prospective sensor deployment sites. LoRa signals will propagate over wide areas, on the order of 20 km in radius, with an unobstructed line of sight. However, obstructions like trees or large hills will cause refraction and diffraction, respectively, reducing the RSSI at the receiver; realistically, signal coverage will support a 1-5km region, but large hills, mountains, or buildings may still render areas unreachable.

Our initial deployment preparations used a custom solution to test LoRa signal strength and log GPS coordinates while walking along streams of interest. Later, we used EnviSense itself with a custom "RF Survey" schedule (Section 5.4) that would perform the same tasks as the custom solution without extra hardware. With the EnviSense version of the system, we receive real-time updates about the signal quality seen by the sensor and the gateway.All GPS and RSSI values are logged to the on-board SD card for additional post-processing. Correlating these GPS and RSSI values generates a map of the area like the one in Figure 3.

To generate this map, the surveying device was mounted to a vehicle while driving between prospective sites and was carried to the specific sites to collect measurements for 2-3 minutes. In

RF-friendly environments, such as those with line-of-sight to the gateway, this is sufficient for interpolating RSSI measurements in the local area. In RF-unfriendly environments, such forests with heavy growth or deep ravines, the signal strength may change drastically even within a few meters due to fading and diffraction from obstacles. In this scenario, interpolating RSSI measurements is more error-prone and requires more careful modeling of the environment.

We believe this topic is worth further investigation to provide predictive maps of RF survey coverage from prospective gateway locations to reduce the overhead of RF surveying. It would be beneficial to have a tool that interpolates the discrete set of RSSI measurements into a continuous map of the signal strength.

5.2 Network Failures

Our LoRaWAN gateway forwards data to and from the server via cellular backhaul. If the server, cellular connection, or gateway fails, the devices must continue sensing, even if data may not reach the database. We have encountered issues including server resets/power outages, gateway unresponsiveness over LoRa, cellular connection failure, etc. that have each manifested in different ways at the level of the sensor. These issues often manifest as frequent retries, thereby expending excessive energy.

We have observed and addressed numerous failures modes. In some cases, failed communication is simply a random collision, which an exponential backoff delay or other channel contention strategy on the data logger will solve. However, communication may fail due to issues in the gateway, Chirpstack network server, or our custom application server. If the LoRa gateway is down, then no packets will be received or acknowledged; the device should assume the network is inaccessible and continue sampling. Network server (ChirpStack) resets will invalidate authentication keys, so EnviSense firmware must attempt to rejoin the LoRaWAN network after prolonged failure. Application server failures will cause data to not be logged nor time/schedule requests responded to. The network server may acknowledge that LoRaWAN frames were successfully received, but this does indicate that they were inserted into the database: The application server should respond with more than a simple acknowledgement to data uplinks.

In practice, we found the gateways were most often the source of network failures. After several months of observation, we see the gateways fail to transmit packets at either the LoRa or cellular interfaces as a Linux service fails. We can detect and solve these issues by frequently checking the interfaces using Chirpstack metadata and network pings to discover which services must be reset.

The danger in not solving these issues is that the devices may put themselves into a state of constant retransmission when the network is unresponsive. This will quickly drain the battery. When designing this system, the developer must be cognizant of the various points of failure within the networking chain and what effects those show at the level of the sensor; diagnostics and logging at each level of this chain is beneficial.

5.3 Visualization Tools and Dashboards

Our application uses Grafana [2] to display the time-series data collected by EnviSense devices. This was originally intended only

Figure 4: Grafana dashboard with data traces for distance (analogous to water level), RSSI, and gateway ping latency. Some periods lack sensor data due to an application server issue.

for the USGS to help them see changes in the environment, but we found it to be useful for other purposes as well.

When debugging a deeply embedded application, unexpected behaviors may occur across a variety of timescales, sometimes on the order of months. Short-term, easily reproducible issues can be fixed entirely with bench testing, but subtle bugs can be much more challenging to track down. In many cases, we found that we could track bugs by periodically checking our Grafana dashboard; an example is shown in Figure 4. In particular, networking or server bugs are easy to locate (in time) as all data channels would show periods in which nothing was recorded. This is shown by the marked blue regions with no sensor readings, in which an error in the application server prevented data from being stored properly.

Further, we used the Grafana dashboard to log other device health information, such as battery consumption, number of watchdog or power-failure resets, current firmware version, and free memory. Deploying bug-free code is unsurprisingly difficult as the complexity of an application increases. We have learned it is essential to build in features that will help discover and reproduce errors so they can be fixed for the next deployment or site visit.

5.4 Remote Application-Level Programming

At the beginning of the EnviSense project, we identified a need for remotely reprogrammable sensors such that the data collection behavior could be updated at runtime. Existing solutions have fixed sensing behavior, which produce large amounts of redundant data under ordinary conditions or too little data under rapidly changing conditions, *e.g.*, the onset of a flood event. Further, application needs may change throughout a device's multi-year life cycle. However, the actual changes themselves are often small: increasing or decreasing a sampling rate, measuring an auxiliary sensor like an accelerometer to check sensor orientation, etc. Assuming the

Table 1: An application schedule that reads sensors on 15 minute intervals and uplinks the readings over LoRaWAN at 15 and 45 minutes past the hour. The parameter of '10' influences a random network delay to limit collisions.

Time	Schedule Function	Parameter
0	READ_RH_AND_TEMP	0
0	READ_BATTERY_VOLTAGE	0
0	READ_DISTANCE	0
900	READ_BATTERY_UAH_CONSUMED	0
900	READ_BAROMETER	0
900	SEND_LORA_FRAME_NOW	10
1800	READ_BATTERY_UAH_CONSUMED	0
1800	READ_BATTERY_VOLTAGE	0
1800	READ_DISTANCE	0
2700	READ_FREE_MEMORY	0
2700	SEND_LORA_FRAME_NOW	10

firmware itself has no need to change, we can encode the application using a short schedule of commands, similar in practice to the virtual machine Maté [11].

We encode a sensing schedule informing what to measure and when within an hour into a single LoRaWAN downlink packet, which is upper bounded to 51 bytes at high spreading factor [8]. These hourly schedules contain 11 commands, 4 bytes each, that set which function to run, such as sensing the ultrasonic distance sensor; the time to run it, relative to the top of the hour in seconds; and an optional parameter, like a network delay parameter. There is an additional command to retrieve a new schedule, at which point it will uplink the current version of its schedule. If the server responds that the current schedule is older than the newest one in the SQL database storing each device's schedules, then the sensor will request the newest one. An example of a schedule is shown in Table 1, which reads a variety of sensors at 15 minute intervals, and sends on 30 minute intervals.

This formulation of remote programming is straightforward and simple, yet we have found useful in unexpected ways. For instance, the RF surveys in Section 5.1 and Figure 3 were conducted by slightly modifying the scheduling firmware to include a single control instruction for branching, GOTO, and an instruction to log the most recent Received Signal Strength Indicator (RSSI) and Signal to Noise Ratio (SNR). We have also used the scheduling mechanism to read an arbitrary SDI-12 transducer, which come in many types commonly used for environmental sensing. This would give field technicians the ability to swap out or add on new transducers without modifying firmware or resetting the device.

This simple combination of macroinstructions and time directives is a powerful tool for remotely programming low-power sensor networks over low throughput wireless links. We plan to continue down this path in greater depth and detail in future work to explore more flexible programming without sacrificing battery life by sending large chunks of machine code over LoRa.

6 DEPLOYMENTS

The first deployment of sensors at Pepperwood Preserve in August of 2019, and the installed sensors remained untouched for approximately four months. During this time, they behaved as intended,

Figure 5: Solar-Powered LoRaWAN gateway at the Pepperwood Preserve deployment

Figure 6: Deployed EnviSense Data Logger at Beale Air Force Base

which meant uplinking temperature, humidity, and most importantly, water level every 30 minutes, where data was sampled every 15 minutes. At the second deployment in December 2019, another sensor was installed at Pepperwood, and the existing sensors had their batteries and SD cards (which contain log files with debugging information and sensor data) swapped for fresh ones.

A third deployment of these sensors was performed in January 2021 at Beale Air Force base to replace an older water level sensor with LoRaWAN using EnviSense data loggers. Several of these EnviSense data loggers use SDI-12 transducers instead of the non-contact, ultrasonic distance sensors to measure water depth (converted to water level).

In general, the size and density of the deployments will depend on the environmental objectives and characteristics of the measured quantities. In reactive applications, like early flood warnings, latency of a few seconds may be critical to safety; this incurs a higher power cost due to frequent communication. Less time-sensitive applications may save energy by communicating larger data packets infrequently, or even sending aggregates like mean water height across a day.

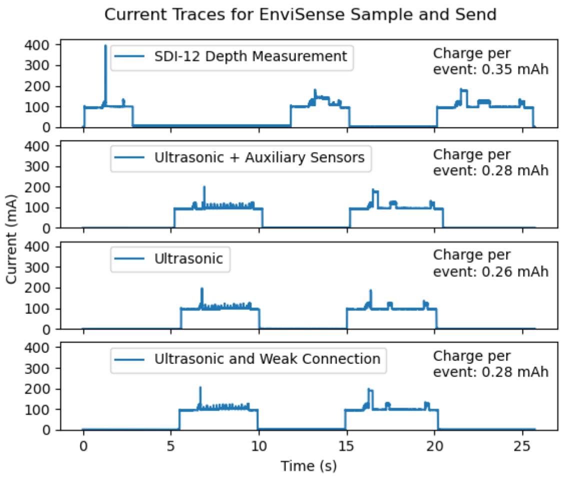

Figure 7: Power trace of EnviSense sampling and transmitting (a) SDI12 pressure transducer uplinked at SF8, 500 kHz bandwidth, (b) Ultrasonic + auxiliary; SF8 500kHz BW, (c) Ultrasonic distance measurements; SF8, 500 kHz BW, and (d) Ultrasonic; SF9 125 kHz BW.

7 RESULTS

In the field, our data loggers have lasted 4-6 months on LiPo battery packs rated at 6600 mAh. This is shorter than desired. We continue to experiment with different chemistries, nominally $LiSOCl_2$ (Lithium Thionyl Chloride), which is a primary cell with good temperature stability and low self-discharge. However, our first attempt experienced brown-outs when transmitting because these batteries have a high internal resistance, causing voltage droop during periods of high current draw. We now use a 100 mF super capacitor to stabilize the input voltage. Our enclosure has enough space to house a substantially larger battery pack of any chemistry that supplies at least 3 V, so long as it uses a JST-2 connector.

The battery charge consumed by a single sampling event (including data uplink with acknowledgement) was measured at 0.262 mAh when using the ultrasonic sensor and 0.353 mAh when using a singular SDI-12 sensor under favorable network conditions using a benchtop supply at 3.7V. When communication fails, signaled by a lack of acknowledgement, the energy spent on a sampling event increased further. The LoRa data rate is adaptive, such that a poor channel will use a higher spreading factor, taking longer to transmit and consuming more energy. Figure 7 shows power traces when sampling and sending data from the SDI-12 pressure transducer measuring water depth, ultrasonic distance and auxiliary sensors (like barometric pressure), ultrasonic sensor only, and ultrasonic under poor network conditions. SDI-12 takes at least 5 seconds to boot, and has a slow interface contributing to higher energy expenditure. Auxiliary sensors (like thermometers) are cheap to sense, but cost nontrivial energy when samples are uplinked. When the channel is poor such that a lower bandwidth and higher spreading factor (SF) are used, the energy cost per bit transmitted also increases.

Our 4-month deployment of sensors, each sampling at 15 minute intervals, is shorter than desired. Figure 8 shows optimistic battery life estimates for various sampling rates with a 6600 mAh LiPo battery pack. Data is assumed to be uplinked immediately after sampling, but the estimates do not account for retransmissions, battery self-discharge, or temperature fluctuations, which account for up to a 40% discrepancy between the estimates and real observations.

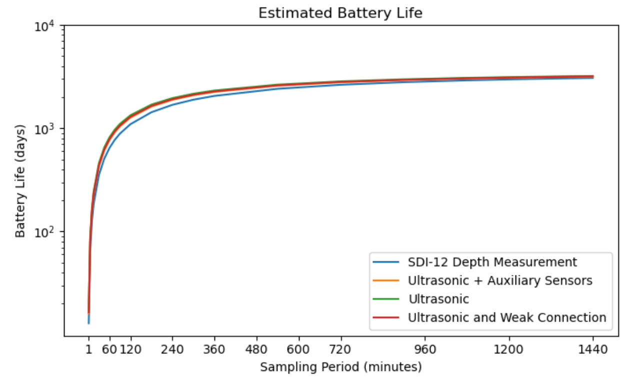

Figure 8: Battery Life estimates for varied sampling rates

This 15 minute sampling interval data is not strictly necessary in most parts of the year for Northern California, where rainfall is concentrated in the winter and spring. We can improve the lifetime without sacrificing data quality by using the remote programming interface to modify sampling rates. Using historical data, there are an average 75 rainy days throughout a year in Sonoma County, CA [12] the location of Pepperwood Preserve. We estimate a battery lifetime of 2.8 years (a 380% improvement) if we sample every 15 minutes on rainy days and every 6 hours on non-rainy days, demonstrating the usefulness of a remote scheduling interface for optimizing system lifetime. By integrating this remote scheduling interface with information feeds like weather forecasts, this could be handled autonomously at scale.

8 CONCLUSION

Environmental sensing continues to be a popular application of wireless sensor networks, with devices lasting for years at a time as networking options have improved. Our sensing platform, EnviSense, has been deployed for nearly six months in a water observation study with the USGS at two sites in Northern California, USA. We have compiled the salient lessons and experiences here. In particular, we re-emphasize the importance of application-level reprogramming tools in which the programs themselves are extremely small (around 100 Bytes) to reduce their cost over low throughput, Low Power Wide Area Networks. We will continue to explore such programming environments as we expand existing sensor networks and deploy new ones.

ACKNOWLEDGMENTS

We would like to thank the USGS National Innovation Center for funding the development and early deployments of the EnviSense system. We also thank Beale Air Force Base and the Pepperwood Foundation for supporting the first deployments of EnviSense, including Ryan Ferrell for his help in selecting sensor sites and setting up mounting hardware at Pepperwood Preserve. This material is based upon work supported by the National Science Foundation under Grant No. 1646235.

REFERENCES

[1] Arduino 2021. *Arduino Due*. Arduino. Retrieved May 24, 2021 from https://store.arduino.cc/usa/due

[2] Grafana Labs 2021. *Grafana: The open observability platform*. Grafana Labs. Retrieved May 21, 2021 from https://grafana.com/

[3] Multitech 2021. *MultiTech Conduit® IP67 Base Station*. Multitech. Retrieved May 24, 2021 from https://www.multitech.com/brands/multiconnect-conduit-ip67

[4] MatBotix Inc. 2021. *Outdoor Beam Sensor - Outdoor Ultrasonic Sensor*. MatBotix Inc. Retrieved May 21, 2021 from https://www.maxbotix.com/product-category/hrxl-maxsonar-wr-products

[5] Link Technologies 2021. *Quickly Create RF Propagation Maps for Anywhere in the World!* Link Technologies. Retrieved May 21, 2021 from https://www.towercoverage.com/

[6] Semtech 2021. *Semtech SX1276 LoRa Radio*. Semtech. Retrieved May 24, 2021 from https://www.semtech.com/products/wireless-rf/lora-core/sx1276

[7] SDI-12 Support Group 2021. *Serial Digital Interface at 1200 Baud*. SDI-12 Support Group. Retrieved May 27, 2021 from https://sdi-12.org/

[8] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. 2017. Understanding the limits of LoRaWAN. *IEEE Communications magazine* 55, 9 (2017), 34–40.

[9] Orne Brocaar. 2021. *ChirpStack, open-source LoRaWAN® Network Server stack*. Retrieved May 21, 2021 from https://www.chirpstack.io/

[10] Roger Coudé. 2020. *Radio Mobile*. Retrieved May 21, 2021 from https://www.ve2dbe.com/english1.html

[11] Philip Levis and David Culler. 2002. Maté: A tiny virtual machine for sensor networks. *ACM Sigplan Notices* 37, 10 (2002), 85–95.

[12] Sperling Best Places. 2020. *Climate in Sonoma County, California*. Retrieved June 5, 2021 from https://www.bestplaces.net/climate/county/california/sonoma

[13] United Stated Geological Survey. 2021. *Nation Water Information System*. USGS Water Data for the Nation. Retrieved June 5, 2021 from https://waterdata.usgs.gov/nwis