EAI Endorsed Transactions
on Internet of Things

Research Article ICST.ORG

PEACH: Predicting Frost Events in Peach Orchards Using IoT Technology

Thomas Watteyne[1,*], Ana Laura Diedrichs[2,5], Keoma Brun-Laguna[1], Javier Emilio Chaar[3], Diego Dujovne[4], Juan Carlos Taffernaberry[2], Gustavo Mercado[2]

[1] Inria, EVA team, Paris, France.
[2] Universidad Tecnológica Nacional (UTN), Mendoza, Argentina.
[3] Instituto Nacional de Tecnología Agropecuaria (INTA), Junín, Mendoza, Argentina.
[4] Universidad Diego Portales (UDP), Santiago, Chile.
[5] CONICET, Mendoza, Argentina.

Abstract

In 2013, 85% of the peach production in the Mendoza region (Argentina) was lost because of frost. In a couple of hours, farmers can lose everything. Handling a frost event is possible, but it is hard to predict when it is going to happen. The goal of the PEACH project is to predict frost events by analyzing measurements from sensors deployed around an orchard. This article provides an in-depth description of a complete solution we designed and deployed: the low-power wireless network and the back-end system. The low-power wireless network is composed entirely of commercial off-the-shelf devices. We develop a methodology for deploying the network and present the open-source tools to assist with the deployment and to monitor the network. The deployed low-power wireless mesh network is 100% reliable, with end-to-end latency below 2 s, and over 3 years of battery lifetime. This article discusses how the technology used is the right one for precision agriculture applications.

Received on 27 April 2016; accepted on 05 May 2016; published on 01 December 2016

Keywords: Smart Agriculture, Precision Agriculture, Deployment, SmartMesh IP.

doi:10.4108/eai.1-12-2016.151711

1. Introduction

In 2013, 85% of the peach production in the Mendoza region (Argentina) was lost because of frost. Because less fruit was produced in the region, 600,000 less work days were needed to process the harvest between November 2013 and March 2014, a reduction in work force of 10,600 people. Across the Mendoza region, frost caused a loss of revenue of 950 million Argentine pesos, roughly 100 million USD (at that time) in the peach business alone.

A frost event happens when the temperature is so low that the crops cannot recover their tissue or internal structure from the effects of water freezing inside or outside the plant. For the peach production, a critical period is when the trees are in bloom and fruit sets (August/September in Mendoza), during which the temperature needs to be kept above -3 C. Even a few hours below that temperature causes flowers to fall, preventing fruit to grow.

Because of the huge economic impact, countermeasures exist, but are expensive. Today, virtually all industrial peach orchards are equipped with a meteorological station that monitors temperature and humidity. Farmers need to deal with false positives (the countermeasure is applied, but there is no frost event) and false negatives (the meteorological station does not pick up a frost event happening in some part of the orchard).

What is needed is a dense real-time monitoring solution deployed in the orchard, feeding a frost prediction model. Having a meteorological station does not provide the measurement density needed. Frost events are micro-climatic: cold and hot air have a different density, wind blows irregularly between the trees, so different parts of the orchard are affected very differently by frost. What we build is a system with

[*] Corresponding author. Email: thomas.watteyne@inria.fr.

a large number of sensing points (air temperature, air relative humidity, soil moisture, soil temperature), at different elevations/depths, throughout the orchard.

Low-power wireless mesh networking technology has evolved significantly over recent years. With this technology, a node is the size of a deck of cards, is self-contained and battery-powered. When switched on, nodes form a multi-hop low-power wireless network, automatically. Off-the-shelf commercial solutions are available today that offer >99.999% end-to-end data reliability and over a decade of battery lifetime.

The goal of the PEACH project is to increase the predictability of frost events in peach orchards by using dense monitoring provided by low-power wireless mesh networking technology.

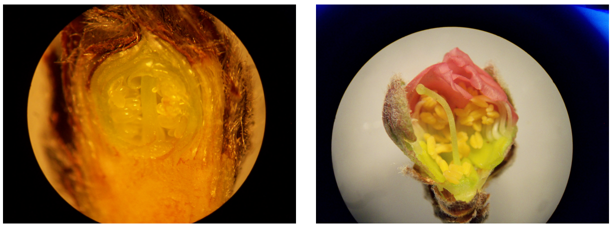
The main contributions of this article are:

- We provide a thorough, complete and "brutally honest" description of the design and deployment of a precision agriculture IoT system.
- We provide a methodology for deploying these networks.
- We contribute with open-source software to perform pre-deployment range tests, manage the network, and store/display sensor measurements and network statistics.
- We describe a complete turn-key solution for precision agriculture applications, built entirely from commercial off-the-shelf proven technology.

The remainder of this article is organized as follows. Section 2 provides the necessary background on how peach growth is affected by frost events, and the major impact frost has on the economy. Section 3 surveys low-power wireless technology used in environmental sensing applications, with a particular focus on SmartMesh IP. Section 4 introduces the hardware deployed in the PEACH project, and the initial range testing procedure used. Section 5 discusses the deployment methodology. Section 6 describes the back-end system. Section 7 shows how the deployed network yields 100% reliability, 3+ years of battery lifetime, and <2 s end-to-end latency. Finally, Section 8 concludes this paper by listing the lessons learnt and discussing the roadmap of the PEACH project.

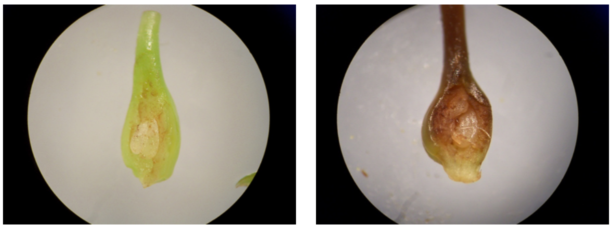
2. Peach Production and Frost

Peach (*Prunus persica* (L.) Batsch.) is a deciduous fruit tree, growing mainly in temperate regions. It has a rest period between vegetative cycles, saving growing organs from extreme winter frost. To leave winter dormancy, buds need to accumulate chill followed by heat, resulting in Spring flowering [1].



(a) Flower bud on vegetative rest (b) Flower bud prior to spring blooming

Figure 1. Peach flower buds at different stages of winter rest. The flower bud has a lower resistance to frost in (b).



(a) healthy (b) damaged

Figure 2. Gynoecium healthy and damaged by frosts. Damage is visualized by brown tissues (extracted from [2]).

Peach flower buds contains floral primordia, consisting of sepals, petals, androecium and gynoecium (see Fig. 1).

Spring frost events are one of the main limiting factors for the production of temperate fruit trees [2]. To develop into a fruit, a flower bud needs to bloom without frost damage, to later be successfully pollinated and fertilized, eventually causing fruit set. When the temperature drops below zero in Spring, the migration of water to intercellular spaces causes tension and breaks the cell membranes. This causes internal solutes to be lost and cells to die.

Fruit production mainly depends on the number of healthy flowers that resist subzero temperatures. Different peach cultivars have different levels of flower bud resistance to Spring frost, depending on the date of full flowering, the lethal temperature, and the flower density [2], see Fig. 2.

In Mendoza, Argentina, there are 14216 ha of peaches, including 5759 ha for fresh consumption [3] and 8457 ha for the canning industry [4]. During Spring 2013, a series of frost events (see Fig. 3) caused major damages in the peach production. 56% of the fruit production area was affected by frost. It is estimated that 85% of the peach production in the Mendoza region was lost. Because less fruit was produced, 600,000 less work days were needed to process the harvest between November 2013 and March 2014, resulting in a reduction in work force of 10,600

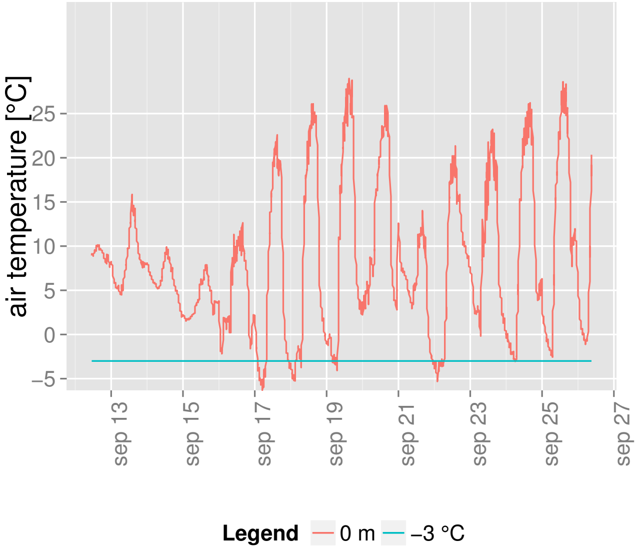


Figure 3. Air temperature at ground level measured in Junín, Mendoza in Spring 2013, showing several frost events.

people. Across the Mendoza region, frost caused a loss of revenue of 950 million Argentine pesos, roughly 100 million USD (at that time) in the peach business alone [3, 5]).

Active frost control is possible [6]. One option is to position heaters across the orchard, but the fuel used is a pollutant and is expensive. Over-plant sprinklers can also be used (latent heat is produced when the water from the sprinklers freezes), but installation cost is high and a lot of water is needed. Another option is to mix the warmer air above the orchard with the cooler air at ground level, causing an increase in temperature around the flowers. This can be done by vertical wind towers, or even flying a helicopter above the orchard. Regardless of the method used, it is expensive. It is therefore essential to be able to predict the frost event.

To predict a frost event, one needs to know the dew point. The problem is that the dew point depends on the relative humidity: at lower relative humidity, air temperature drops faster and to lower values. To predict the minimum temperatures – hence a frost event – one needs to observe the temperature at several locations across an orchard, at different elevations, since winds and foliage cause micro-climatic differences in the orchard. Using a single meteorological station in the orchard is not enough; the temperature and humidity it measures are not representative of the entire orchard.

To accurately characterize and predict frost events, what is needed are air temperature and relative humidity measurements at different locations across the orchard, at different elevations, as well as soil temperature and soil moisture at different depths. This data can then feed to a machine-learning algorithm to determine which features are important to predict frost events, and eventually build an early warning system.

3. Low-power Wireless Technology

The goal of the PEACH project is to retrieve data from different types of sensors deployed densely across a peach orchard. Because heavy machinery is used throughout the field, using wires to interconnect the sensors is not an option. We instead use low-power wireless mesh networks. Our requirement is that the security and reliability of those wireless networks must be equivalent to that of a wired network. Our target battery lifetime for the devices is "at least a year", the natural grow cycle of peaches.

This project is not a low-power wireless project, i.e. we need a technical solution that "just works". While very interesting research can be done with academic projects [7, 8] (there are many in the research field of low-power wireless), we need a *product* that is thoroughly tested, proven and commercially available off-the-shelf. We have selected SmartMesh IP, which satisfies our requirements. Section 3.1 introduces SmartMesh IP technology, Section 3.2 reviews related projects that use low-power wireless networks for environmental applications, Section 3.3 lists the goals of the PEACH project.

3.1. SmartMesh IP

SmartMesh IP, a product line from the Dust Networks group at Linear Technology[1] fits the requirements listed above. Dust Networks is initially a spin-off company from the University of California at Berkeley, now the world leader in supplying low power wireless mesh networks for demanding applications. Over 50,000 SmartMesh networks are deployed today in applications including industrial process monitoring, city-wide parking management, building automation and remote environmental sensing.

The protocol stack implemented in SmartMesh IP devices is entirely standards-based, and includes standards such as IEEE802.15.4 "Time Synchronized Channel Hopping" (TSCH) and IETF 6LoWPAN [9]. A SmartMesh IP network is composed of exactly one manager and up to 100 motes. The manager serves as the gateway of the SmartMesh IP network, and is typically connected to a computer, itself connected to the Internet. A mote is a standalone device, typically battery powered, itself connected to sensors and actuators. The SmartMesh IP network offers bi-directional connectivity: the motes can send sensor measurements to the gateway (and from there to

[1] http://www.linear.com/dust

some server on the Internet), the server can send actions/configurations to the motes.

Security is built into the SmartMesh IP protocol stack around an AES-128 cipher, and cannot be switched off. A set of keys ensures confidentiality, integrity and authentication of the data. The SmartMesh IP network is operated in such a way that it offers wire-like reliability, with over 99.999% end-to-end reliability.

The motes in a SmartMesh IP network automatically and periodically generate "Health Reports", a series of counters and statistics to assess the overall health of the network. Three types of statistics are generated: internal counters of the mote (e.g. packet counters), the list of neighbors the mote is communicating with, and the list of neighbors the mote can hear but is not communicating with. A mote generates a complete set of health reports every 15 min. They are the base for assessing the performance of the network, as done in Section 7.

A SmartMesh IP network is fully synchronized, and time is split up into timeslots. A schedule orchestrates all communication in a SmartMesh IP network and indicates what to do in each timeslot: transmit, listen or sleep. How the schedule is built and maintained allows a clear trade-off between the amount of data generated by the sensors, the communication latency, and the power consumption of the motes.

The "Dust Networks SmartMesh Power and Performance Estimator" is a tool provided by Dust Networks[2] that allows one to estimate the power consumption and latency of a SmartMesh IP network, given the topology and amount of data generated. A typical SmartMesh IP network offers over 99.999% reliability and over a decade of battery lifetime when motes are powered by a pair of AA batteries [10].

3.2. Related Projects

In recent years, low-power wireless is being used more and more by the scientific community for remote environmental sensing.

One community at the forefront of adopting low-power wireless technology is the hydrology community. Bogena *et al.* highlight the potential of low-power wireless for measuring soil water content variability [11], Pohl *et al.* do a similar analysis for understanding the snow cover [12]. Rice and Bales show how embedded sensors can be used to evaluate the water content of snow [13] and Qian et al. propose a system using short range nodes and cellular phones to capture orchard data [14]. Simoni *et al.* use wireless sensor networks to model the hydrologic response of an alpine watershed [15]. Li *et al.* summarize lessons learnt from deploying a wireless sensor network for soil monitoring [16]. Gutierrez *et al.* use low-power wireless to monitor water and automate irrigation [17]. Moreover, Ojha et al. [18] survey the state of the art of wireless sensors for agriculture.

One of the leading groups in using low-power wireless for environmental monitoring is UC Berkeley Prof. Steven Glaser's team. They have designed a complete remote environmental monitoring solution around SmartMesh IP [19].

This system is now used in the UC Berkeley Botanical Garden. This garden has over 13,000 types of plants from around the world, cultivated by region in naturalistic landscapes over its 34 acres. Irrigation in such a diverse botanical environment is complex since different plants have different needs. The Glaser team has deployed a network of 5 motes and 8 repeaters[3] to assist the team of botanists with continuous in-situ measurements. Sensor stations are equipped with air temperature, relative humidity, soil moisture, soil temperature and soil electrical conductivity sensors at two depths. The gateway is connected to a server hosted at UC Berkeley through a cellular connection[4].

The largest deployment of the Glaser team is the American River Hydrological Observatory (ARHO)[5]. ARHO is a set of 15 networks deployed across the Sierra Nevada mountain range in California, USA, to monitor the melting process of the snowpack. A total of 945 sensors are wirelessly connected through SmartMesh IP technology, each reporting their measurements every 15 min to a central server on the UC Berkeley campus.

3.3. Goals of the PEACH Project

The PEACH project uses technology similar to the ARHO deployments, and applies that to monitoring peach orchards. The goal of the project is to predict frost events by applying machine-learning to the data gathered. The project is organized in 3 phases:

- Phase 1: Deploy the low-power wireless mesh network, with only air temperature sensors.
- Phase 2: Augment the network with relative humidity, soil temperature and soil moisture sensors.
- Phase 3: Apply machine learning to the data and predict frost events.

The project is currently in Phase 1. This article focuses exclusively on the low-power wireless network:

[2] http://www.linear.com/docs/42452

[3] "Student-installed Sensors Help Monitor Botanical Garden", http://news.berkeley.edu/, 20 October 2015.

[4] A simplified interface is accessible at http://glaser.berkeley.edu/wsn/botgn/hist.html.

[5] http://arho.org/

what hardware is used (Section 4), how the deployment is done (Section 5), how the back-end system is built (Section 6), and how the system performs (Section 7).

4. Hardware

The PEACH network consists of 21 low-power wireless motes connected through a SmartMesh IP network to a manager, the gateway of the network. Section 4.1 and 4.2 introduce the motes and manager used, respectively. Section 4.3 shows how the devices are packaged. Section 4.4 describes the range testing done before the deployment.

4.1. Low-power Wireless Motes

Fig. 4 shows the three types of motes that are used. All boards are manufactured by Linear Technology. 16 DC9003 (Fig. 4(d)), 2 DC9018 (Fig. 4(c)) and 3 long-range prototype boards (Fig. 4(b)) are deployed.

The DC9003 (Fig. 4(d)) is the development board for SmartMesh IP networks. This is the board used to experiment with and prototype SmartMesh IP technology. All the pins of the micro-controller are exposed, allowing a developer to interface sensors and actuators over digital and analog interfaces. The DC9003 features the LTC5800 chip (a SoC with an ARM Cortex-M3 and an IEEE802.15.4 radio) connected to a chip antenna. The DC9003 is an off-the-shelf commercially available board.

The DC9018 (Fig. 4(c)) is functionally equivalent to the DC9003, but with a 2 dBi external antenna, rather than a chip antenna. It is expected that the DC9018 has a better range than the DC9003. The DC9018 is an off-the-shelf commercially available board.

The long-range board depicted in Fig. 4(b) is a prototype of a new product Linear Technology allows us to test. It is a fully SmartMesh IP-compliant product (i.e. it interoperates with DC9003 and DC9018 boards in the same network), with a line-of-sight range of 1500 m. This board features a 2 dBi external antenna.

The boards deployment in Phase 1 of the PEACH project are developer/prototyping boards. We expect to transition to the NeoMote in Phase 2, a ruggedized version of the motes, manufactured by Metronome Systems[6].

All the boards deployed during Phase 1 run the default SmartMesh IP firmware. Each board automatically joins the network and sends a temperature reading every 30 s to the manager, using its built-in temperature sensor. In Phase 2, we will reprogram the motes with custom firmware, using the “DustCloud” development environment provided by Linear Technology[7].

4.2. Low-power Wireless Manager

Fig. 4(a) shows the SmartMesh IP manager used in the PEACH project. It is composed of an LTP5902-IPR SmartMesh IP manager chip, a 2 dBi external antenna and USB connectivity, in the form-factor of a USB pen drive. USB connectivity is used to connect the DC2274 to a computer. The computer can run a program to configure the DC2274, send commands and configurations to the motes, and retrieve sensor measurements. The DC2274 is a standalone device, i.e. the computer does not play any active role in operating/managing the network, it just serves as an interface between the SmartMesh IP network and the Internet. The DC2274 is an off-the-shelf commercially available board.

4.3. Hardware Integration

The PEACH network is deployed in an orchard for 2 years, and will be exposed to direct sunlight, freezing temperatures, dust and rain. All motes and manager are therefore protected by an enclosure with Internal Protection 65 (IP65) rating[8].

Fig. 5(a) shows the manager enclosure with the lid open. It contains the DC2274 board, a Raspberry Pi single-board computer running the `solmanager` software (see Section 6.2), and the power adapter for that board. The DC2274 is powered through USB. A water-tight pass-through opening in the manager enclosure is used to pass power and Ethernet cables.

Fig. 5(b) shows a mote in its enclosure. The mote is powered by a pair of Energizer L-91 AA batteries with a charge of 2821 mAh[9]. The mote, antenna and battery are glued in place using silicone. No opening is present in the mote enclosures.

4.4. Initial Range Testing

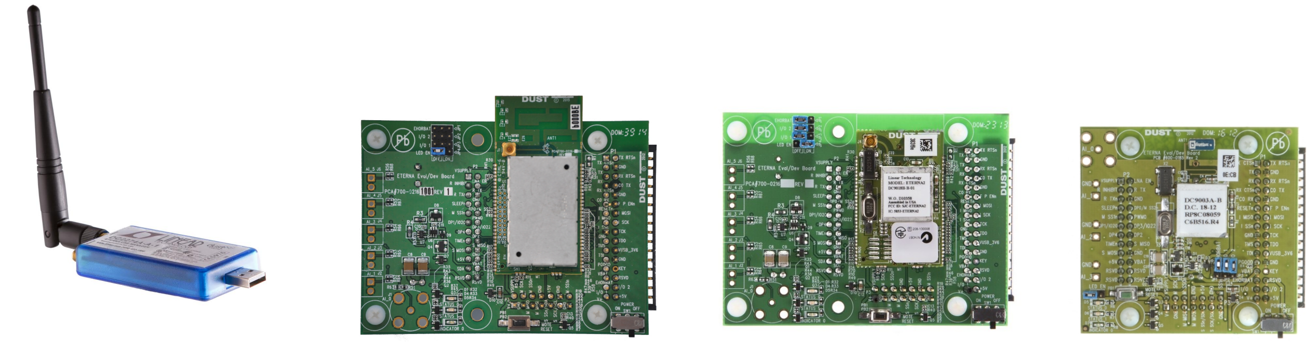
To plan the deployment, it is important to get some intuition about the communication range of the different types of motes. One question is whether the long-range motes can connect the orchard to the manager 380 m away. Another question is whether the DC9003 motes have a communication range large enough to create a mesh network inside the orchard. The DC9003 is meant to be used by a developer on an office desk to prototype connectivity with external sensors, and features a chip antenna with a gain not exceeding 1 dBi.

[6] http://metronomesystems.com/

[7] http://www.dustcloud.org/

[8] Totally dust tight and protection against low pressure water jets in all directions.

[9] The charge mentioned in the datasheet of the Energizer L-91 is 3135 mAh, but we assume a 10% derated charge to account for shelf-life.



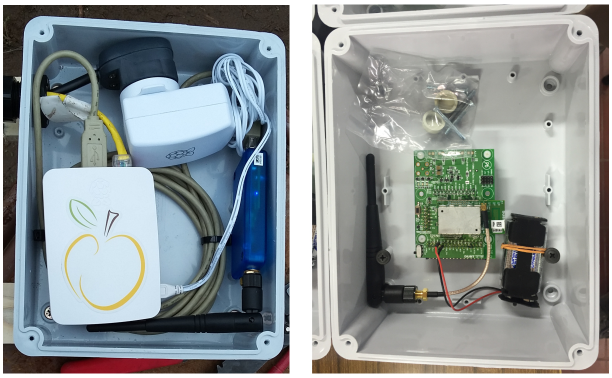
(a) DC2274: Low-power wireless manager *(1 deployed)*

(b) Long-range prototype board *(3 deployed)*

(c) DC9018: External antenna mote *(2 deployed)*

(d) DC9003: chip antenna mote *(16 deployed)*

Figure 4. The boards deployed.



(a) manager

(b) mote

Figure 5. The devices encased in 20 cm × 15 cm IP65 enclosures.

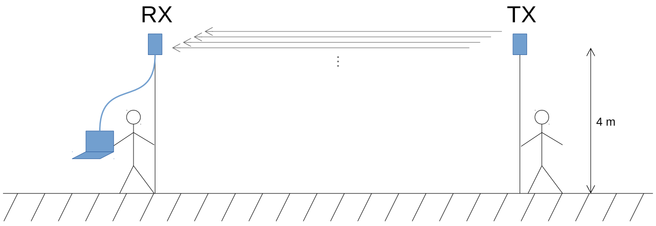


Figure 6. Setup for range testing.

Any SmartMesh IP device has a built-in "radiotest" mode, typically used to verify that the antenna is well matched to the radio chip. We use that mode to perform range testing, and combine it with a RangeTest application we developed[10]. Fig. 6 illustrates the setup used. One transmitter mote (marked "TX") is configured in radiotest mode to transmit 40-byte packets every 10 ms, round-robin on 4 frequencies evenly spaced in the 2.4 GHz frequency band[11]. We use multiple frequencies since SmartMesh IP uses channel hopping. A receiver mote (marked "RX") is connected to a laptop computer running the RangeTest application. This application, through the serial API of the mote, has the mote listen for 4 s on each of the 4 frequencies, and counts the number of received packets. During that 4 s listening window, the RangeTest application expects to receive 100 packets; receiving fewer packets indicates a less-than-perfect wireless link. Transmitter and receiver motes are placed in their enclosures on top of 4 m poles, which are moved around, and mimic the poles the motes will be deployed on (see Section 5). We record the average Packet Delivery Ratio (PDR) across all 4 channels, i.e. the portion of received packets, between different locations.

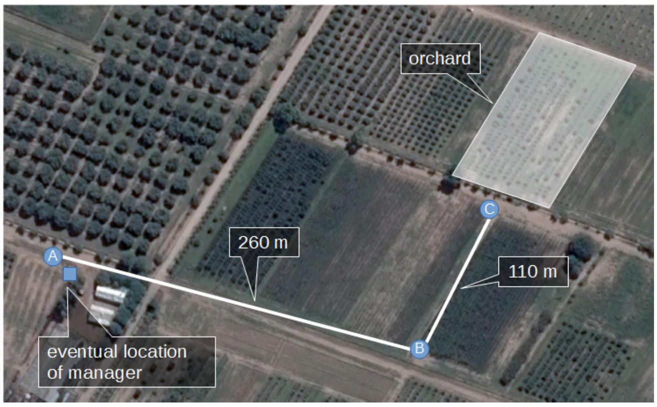
The first question is: *Can the long-range motes connect the orchard to the manager, some 380 m away?* We use long-range boards as both transmitter and receiver, and position them at positions A–B (a 260 m link), then B–C (a 110 m link). Positions and results are depicted in Fig. 7: the long-range links have an excellent PDR (>95%).

We continue range testing in and around the orchard. As depicted in Fig. 8, we position the transmitter and receiver poles in different locations across the orchard. We classify links as good (80-100% PDR), medium (50-80% PDR) and bad (0-50% PDR). We use a DC9018 board as receiver, a DC9003 board as transmitter.

The second question is: *Can a network deployed inside the orchard reliably connect to a mote in position* C*?* We position the receiver at position C and move the transmitter across positions D–H. As depicted in Fig. 8, motes in positions D and E have a good connection with a mote in position C.

[10] As an online addition to this article, the source code of the RangeTest application is available under an open-source BSD license at `https://github.com/realms-team/rangetest`.

[11] We use frequencies 2.405 GHz, 2.425 GHz, 2.445 GHz and 2.465 GHz



positions	distance	PDR
A–B	260 m	97%
B–C	110 m	100%

Figure 7. Range testing using long-range prototype boards.

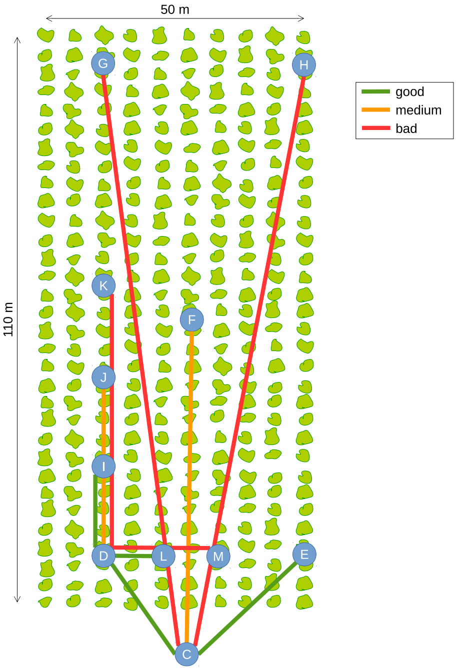


Figure 8. Locations for range testing in the orchard.

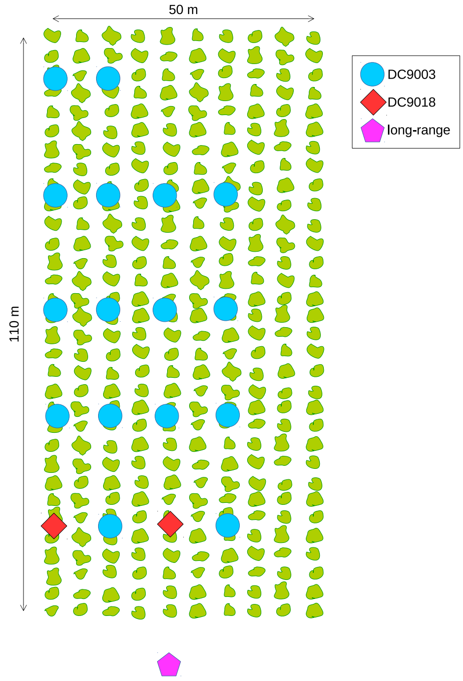


Figure 9. Final locations of the deployed motes in the orchard.

The third question is: *How good do motes connect within the orchard?* Results are depicted in Fig. 8: connection is good until 5 trees away in the same row, and until 2 rows away.

These initial measurements give us *some* intuition about the range of the different motes, and the feasibility of the deployment. It does not, however, provide any (long-term) guarantees, as the quality of the links changes over time. In particular, it does not preclude us from building a system that observes long-term connectivity statistics and provides the user with feedback about the “health” of the network. Such a system is part of the back-end developed for the PEACH network, and presented in Section 6.

5. The PEACH Deployment

The initial range testing (Section 4.4) suggests a full deployment is feasible. Together with the INTA agronomy team, we determine the best locations to position the sensor stations in the orchard, as shown in Fig. 9. The manager and the long-range motes are placed as depicted in Fig. 7.

We start by switching on the manager, and verify it is operational and connected to the Internet. We use a laptop computer in the orchard, connected to the Internet through a cellular link. We connect the laptop to the manager over SSH, and continuously list the motes that have joined the network. We then deploy the motes from the manager outwards: starting by the long-range motes in positions A, B and C in Fig. 7, then the motes in the orchard.

We install all motes 4 m above the ground on wooden poles. Long-range motes are attached to existing poles (Fig. 10(a)). Dedicated poles are placed in the orchard (Fig. 10(b)).

Deployment itself is straightforward and fast. We switch on the mote to be deployed, close the lid, screw it on its pole, and verify that it has joined the network and that we can “ping” it over the SmartMesh IP network. The entire PEACH deployment took less than 4 hours; most of the time spent moving from location to location and securing the mote enclosures to the poles.

(a) Long-range mote at position B in Fig. 7.

(b) Mote in orchard.

Figure 10. Members of the team deploying motes.

6. SOL: Back-end System

We call "back-end" the system that connects to the SmartMesh IP manager, retrieves all the sensor measurements and network statistics (the "health reports" generated by the motes), and ensures those appear in a remote database. We implemented a complete back-end system[12]. As illustrated in Fig. 11, the system consists of the `solmanager` application that runs on the Raspberry Pi single-board computer, and the `solserver` application running on a server in the Inria-Paris research center.

6.1. SOL data representation

The SmartMesh IP continuously publishes information: the data produced by the motes, the health reports produced by the motes, and network events (e.g. a mote joined the network) produced by the manager. The data produced by the motes contains the bytes generated by the application running on the mote; this data is formatted differently depending on the application.

We created the "Sensors Object Library" (SOL) to wrap that information as generic SOL objects. A SOL object is always composed of the same fields: the identifier of the device generating the information, the timestamp of when this information was generated, the type of information, the value of the information. A SOL object can be seen as a generalization of the Type-Length-Value (TLV) format.

A SOL Registry contains the list of SOL types currently used, and their associated format. This includes data generated by the motes, health reports or network events. Each SOL object can be represented either in a compact binary form, or in a more easily parseable JSON format. The SOL registry is publicly maintained and can be easily augmented with new types of sensor data[13].

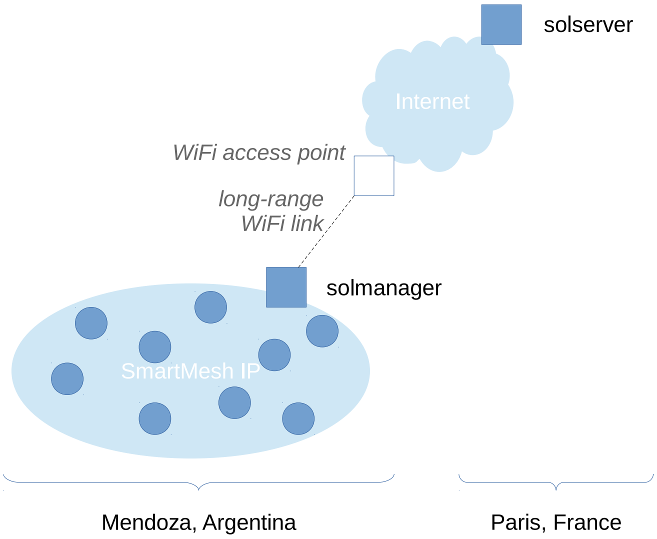


Figure 11. The SOL back-end system.

6.2. `solmanager`

The `solmanager` application runs on the Raspberry Pi and connects to the serial port of the SmartMesh IP manager. It subscribes to receive all the notifications from the SmartMesh IP manager: data, health reports, network events. It then parses that information and formats it as a series of SOL objects, which it forwards to the `solserver` application.

In the PEACH deployment, the Raspberry Pi is connected to an Ethernet-to-WiFi bridge equipped with a long-range directional antenna pointing to a building with Internet access 200 m away. In practice, this means the Raspberry Pi is directly connected to the Internet. Through that connection, the Raspberry Pi forwards all the generated SOL objects, in binary format, to the `solserver` application over HTTPS. Public/private SSL keys and an HTTP token provide confidentiality, integrity, authentication and authorization of the `solmanager` ↔ `solserver` communication.

[12] As an online addition to this article, the source code of the back-end system (including the `solmanager` and `solserver` applications) is available under an open-source BSD license at `https://github.com/realms-team/`.

[13] `https://github.com/realms-team/sol/blob/master/registry.md`

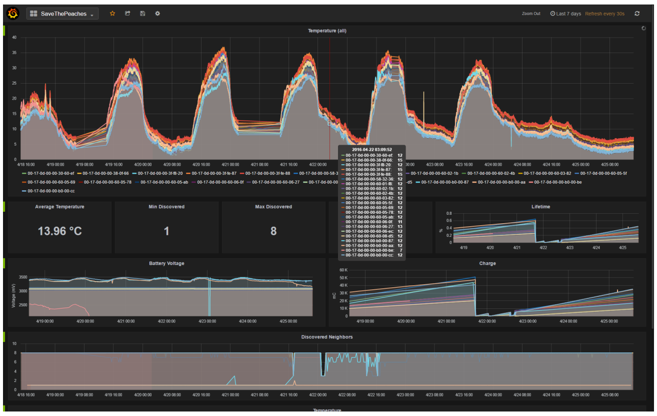

SaveThePeaches
Zoom Out
Last 7 days
Refresh every 30s
Temperature (all)
2016-04-22 03:09:52
00-17-0d-00-00-30-60-ef: 12
00-17-0d-00-00-38-0f-66: 15
00-17-0d-00-00-3f-f8-20: 12
00-17-0d-00-00-3f-fe-87: 15
00-17-0d-00-00-3f-fe-88: 15
00-17-0d-00-00-58-32-36: 12
00-17-0d-00-00-60-01-f8: 12
00-17-0d-00-00-60-02-1b: 12
00-17-0d-00-00-60-02-4b: 12
00-17-0d-00-00-60-03-82: 12
00-17-0d-00-00-60-05-5f: 12
00-17-0d-00-00-60-05-69: 12
00-17-0d-00-00-60-05-78: 12
00-17-0d-00-00-60-05-ab: 12
00-17-0d-00-00-60-06-0f: 11
00-17-0d-00-00-60-06-27: 13
00-17-0d-00-00-60-06-ec: 12
00-17-0d-00-00-60-08-d5: 12
00-17-0d-00-00-b0-00-87: 12
00-17-0d-00-00-b0-00-aa: 12
00-17-0d-00-00-b0-00-be: 7
00-17-0d-00-00-b0-00-cc: 12
Average Temperature
13.96 °C
Min Discovered
1
Max Discovered
8
Lifetime
%
Battery Voltage
Voltage (mV)
Charge
mC
Discovered Neighbors
Temperature

Figure 12. Example dashboard showing the last 7 days of data.

6.3. solserver

The `solserver` application runs on a server in the Inria-Paris research center. The application listens for incoming SOL objects. When those objects are received from a properly authorized `solmanager` application, the `solserver` converts them from their binary representation to their JSON representation, and writes them into a database[14].

A dashboard allows a user to visualize key information[15]. Fig. 12 shows an example dashboard with 7 days worth of data, including the temperature (the day/night temperature swing is clearly visible, the last day being cold), the number of discovered neighbors, and the charge consumed by the devices.

7. Performance Results

All motes in the network produce a complete health report every 15 min. On top of this, the SmartMesh IP manager provides a complete CLI (Command Line Interface) on one of its serial ports for an operator to type commands and retrieve information about the network. We base the analysis in this section on the data retrieved from the health reports and the CLI of the SmartMesh IP manager.

We divide the performance results in the performance of the overall network (Section 7.1) and the performance of the motes (Section 7.2).

7.1. Performance of the Network

Table 1 summarizes the network performance after 5 days and 6 hours of operation. After approx. 24 hours of operation, a SmartMesh IP network reaches "steady state", so we expect the numbers presented in this section to be representative of the lifetime of the network.

reliability	100% (Arrived/Lost: 243089/0)
stability	93% (Transmit/Fails: 1462435/96923)
latency	800 ms

Table 1. Key network performance indicators.

The network **reliability** indicates the portion of the packets generated by the motes that reach their final destination, in our case the SmartMesh IP manager. A packet is said "lost" when it never reaches its destination. The motes in the PEACH deployment have generated 243,089 packets, all were successfully received by the SmartMesh IP manager, possibly after multiple hops, yielding 100% reliability.

The wireless medium is unreliable in nature, and it is common that a link-layer frame sent by mote A is not received by mote B, forcing A to *re*-transmit. The network **stability** represents the average PDR over the link. The motes in the PEACH deployment have sent 1,462,435 data-link frames, 96,923 unsuccessfully (i.e. no link-layer acknowledgment was received), yielding a network stability of 93%. This number is very high, indicating that the nodes are deployed close enough in an environment with little external interference and multi-path fading.

A SmartMesh IP network is fully synchronized. A mote generating a data packet adds a timestamp to its data so that what is written in the database is the time the sensor is sampled (not the time it reaches the database). As a side effect, it is possible, at the SmartMesh IP manager, to calculate how long the packet was traveling in the multi-hop wireless mesh network. This is called average network **latency**. In the PEACH deployment, it takes on average 800 ms for a sensor measurement to travel from the mote that generated it to the SmartMesh IP manager, over the multi-hop low-power wireless mesh network. This latency is very small compared to the variation speed of the signal we are observing (temperature).

7.2. Performance of the Motes

The health reports generated by each mote contain a wealth of information. Table 2 contains a summary of the most important information for assessing the performance of the network.

Each line in Table 2 is indexed by the unique identifier of the mote, its "**MAC** address". This is a unique 8-byte number – called EUI64 – written into the mote at manufacturing time. The EUI64 of all motes starts with `00-17-0d-00-00`; in the interest of space, this 5-byte prefix does not appear in Table 2.

[14] We use an influxdb (`https://influxdata.com/`) database.
[15] The dashboard is publicly available on the PEACH project website (`http://www.savethepeaches.com/`).

MAC	Uptime	Neig.	Cells	Hops	Latency	Recv'd	Lost	Relia.	Charge	Lifetime
60-3C-D9★	5-06:57:07	2	40	*N.A.*	*N.A.*	*N.A.*	*N.A.*	*N.A.*	*N.A.*	*N.A.*
B0-00-AA	5-06:56:06	3	20	1.0	170 ms	16450	0	100%	33263 mC	4.42 years
B0-00-CC	5-06:54:55	5	18	2.7	280 ms	16657	0	100%	28396 mC	5.18 years
58-32-36	3-09:29:11	10	21	3.0	420 ms	10788	0	100%	27546 mC	3.43 years
B0-00-BE	3-10:17:42	6	15	2.8	670 ms	10560	0	100%	15147 mC	6.30 years
60-06-0F	3-09:01:05	4	11	3.5	1020 ms	10386	0	100%	15759 mC	5.96 years
B0-00-87	3-10:06:30	6	37	1.2	120 ms	10628	0	100%	20038 mC	4.75 years
3F-F8-20	3-09:21:15	2	9	3.6	1180 ms	10428	0	100%	11907 mC	7.92 years
30-60-EF	3-09:16:26	9	24	2.7	340 ms	10551	0	100%	22471 mC	4.19 years
60-03-82	3-09:11:22	4	11	3.5	740 ms	10409	0	100%	15977 mC	5.89 years
60-08-D5	3-09:04:03	2	9	3.5	810 ms	10389	0	100%	11173 mC	8.41 years
3F-FE-88	3-08:56:18	3	10	3.5	1210 ms	10384	0	100%	12924 mC	7.26 years
3F-FE-87	3-08:51:28	2	9	4.2	1440 ms	10372	0	100%	11900 mC	7.88 years
60-05-5F	3-08:45:53	2	9	4.4	1860 ms	10346	0	100%	10867 mC	8.61 years
60-06-27	3-08:45:09	4	12	3.6	770 ms	10368	0	100%	14629 mC	6.40 years
60-05-69	3-08:40:17	2	9	3.6	1100 ms	10334	0	100%	10915 mC	8.57 years
60-01-F8	3-08:37:02	3	10	3.6	640 ms	10322	0	100%	11292 mC	8.28 years
60-02-4B	3-08:31:59	2	9	4.3	1520 ms	10326	0	100%	13186 mC	7.08 years
60-02-1B	3-08:28:39	6	14	3.5	650 ms	10301	0	100%	14700 mC	6.35 years
60-05-AB	3-08:22:30	3	10	4.0	920 ms	10298	0	100%	12808 mC	7.27 years
60-06-EC	3-08:21:17	2	9	4.4	1740 ms	10289	0	100%	10964 mC	8.50 years
38-0F-66	3-08:03:38	2	9	4.4	1430 ms	10254	0	100%	10781 mC	8.61 years
60-05-78	3-08:00:26	2	9	3.6	950 ms	10247	0	100%	10710 mC	8.66 years

★ the manager

Table 2. Key mote performance indicators.

The **Uptime** is the time the mote has been operational, and is of format `day-hour:min:sec`. The 3 first motes were deployed 2 days before the others; it is hence normal to have a 2-day gap in their uptime.

Each node reports the number of neighbors it currently sees. This is shown in column "**Neig.**" in Table 2. The number of neighbors indicates the density of the deployment. Dust Networks recommends for each mote to ideally have 3 or more neighbors. This is not the case for 10 of the motes in the PEACH deployment. We expect that, when swapping the DC9003 boards (chip antenna) for NeoMote boards (external antenna), the density will increase.

All communication in a SmartMesh IP network is orchestrated by a communication schedule. The column **Cells** in Table 2 indicates the number of cells in the schedule the mote is active in (transmitting or receiving). That number is directly proportional to the energy it consumes. We analyze the energy consumption more finely below.

Each packet sent through the low-power SmartMesh IP mesh network contains a hop count field[16]. This allows the SmartMesh IP manager to know how many hops this packet took to go from its source to the manager. The value printed in the **Hops** field is an average calculated over the packets received so far. As can be seen from Table 2, the closest (resp. furthest) mote is 1.0 (resp. 4.4) hops away from the manager. Dust Network recommends to keep that number below 8 for efficiency; a condition we satisfy.

The **Latency** column shows the average latency for a packet to go from that node to the manager. The average value of all motes is the network latency shown in Table 1. As can be seen, the latency increases with the number of hops.

The overall network reliability is shown in Table 1. Table 2 shows the same information, but broken up per mote. Column "**Recv'd**" shows the number of packets generated by that mote that have reached the manager. Column "**Lost**" shows the number of packets lost en route. No packets are lost, the **Reliability** for each node is 100%.

In their health reports, motes indicate the amount of **Charge** drawn so far from their battery, in mC. Knowing their uptime and the theoretical charge of

[16] Equivalent to the "hop limit" field in the IPv6 header.

their battery[17], we calculate the expected **Lifetime** of each mote. The expected lifetime varies between 3.43 years and 8.66 years, depending on the type of motes (a long-range mote consumes more than a DC9003) and its activity (a mote with more cells consumes more). These numbers are well above the targeted 1 year of lifetime.

8. Lessons Learnt, Ongoing Work and Roadmap

We conclude this article by listing the lessons learnt from this deployment (Section 8.1) and detail ongoing work and roadmap (Section 8.2).

8.1. Lessons Learnt

Although frost events may appear harmless, they yield enormous losses in fruit production, for example peaches. The most important lesson learnt is that IoT is the right technology for this precision agriculture application, and that using it makes a huge monetary difference. Moreover, as seen throughout this project, IoT technology is ready for this type of application. We strongly believe that the combination of technologies demonstrated through the PEACH project is the right one for a large number of "Smart Agriculture" applications. We hope that the present article can guide end users put together the right technical solution.

The main outcome of this project is a perfectly working end-to-end low-power wireless distributed sensor system, built exclusively from commercial off-the-shelf components. Deploying such a network took a couple of hours, putting the whole hardware and software together a couple of days. The resulting network is 100% reliable, offers latency below 2 s, and 3+ years of battery lifetime.

In 2006, Langendoen *et al.* wrote a foundational (and "brutally honest") paper listing everything that went wrong in a precision agriculture deployment similar to the PEACH project [20]. This included board failure, batteries running out, sub-optimal software tracking, etc. So what went *right* this time, about a decade later? The teams in both cases are of the same caliber, so the *wrong* conclusion would be to blame/praise the people involved. What has really happened is that the field of low-power wireless has evolved substantially, and has radically changed in that decade.

In 2006, *researchers* bought boards and programmed them from scratch with academic open-source projects. In 2016, *end users* buy complete working systems, including all the hardware and software.

In 2006, the protocol stack implemented was a combination of the most promising research papers from recent years. In 2016, the protocol stack implemented is entirely standards-based, with standards matured over years in a commercial/industrial mind-set.

In 2006, every new deployment was pushing the frontier. In 2016, tens of thousands of these network have been running for years in the most critical applications.

To put it plainly, **Wireless Sensor Network/Internet of Things technology has successfully transitioned from the academic to the commercial world**.

The result is a necessary redefinition of the work of the academic community. The era of building everything from scratch is over. This is in particular true for software implementations; a small research group is simply no match for the heavy lifting full-time development teams do. In our opinion, the energy of small and agile research teams should be spent on (1) ensuring that clever ideas transition to the commercial world through standardization and (2) be very well connected to the commercial world, for example by continuously surveying existing products/technologies. As for point (2) above, we encourage journal editors and conference program chairs to include "lessons learnt from practical deployments" onto the topics of interest.

8.2. Ongoing Work and Roadmap

Today, we are only 5 months into the 2-year PEACH project. While the choices for the base networking technology are validated by the work presented in this article, a lot still needs to be done.

The first step is to add additional sensors to the motes, including relative air humidity, soil moisture, soil temperature and soil conductivity. This might include adding NeoMotes to the deployment.

One aspect of the PEACH project that is missing so far is the data analysis. We plan to apply machine learning techniques, in particular ensemble regression-tree, on the large-scale sensor measurement dataset. This will require us to augment the sensor measurements with features that could help predict frost events. We plan to connect an anemometer to the deployment, and to conduct partial flooding of the orchard to quantify the impact of flooding on the intensity of frost events.

On the software side, one immediate next step is to augment the visualization on the back-end system. While the dashboard shown in Fig. 12 is a convenient tool for navigating the raw data, it does not interpret that data. We are working on "network control tower" software which, based on the health reports and events it receives, is able to alert an operator that the network functions sub-optimally, and make recommendations.

Finally, and most importantly, the combination of technologies is perfectly suited for micro-climatic modeling in precision agriculture, but we believe it is also perfectly suited for other fields. We are actively

[17] We assume 2821 mAh, per Section 4.3 of charge, or 101,556,000 mC.

looking to extend this solution to Industry 4.0, building automation and infrastructure monitoring applications.

Acknowledgments

The authors thank the Dust Networks product group at Linear Technology and Metronome Systems for their technical support. The authors thank Nicolás Altamiranda, Guillermo Grünwaldt and Matías Pecchia for their assistance during the deployment. This publication was made possible by the support of the STIC-AmSud program, under grant 16STIC08-PEACH.

References

[1] J. Chaar and D. Astorga, "Use of a correlation model for the determination of peach tree [prunus persica (l.) batsch.] heat and chilling requirement," *Revista de Investigaciones Agropecuarias (RIA)*, vol. 38, no. 3, pp. 289–298, 2012.

[2] J. E. Chaar, "Characterization of Peach [Prunus persica (L.) Batsch.] Cultivars for Frost Resistance," *Acta Agronómica*, vol. 64, no. 3, pp. 230–237, 2015.

[3] Instituto de Desarrollo Rural. (2010) Censo Frutícola Provincial 2010 Mendoza - Argentina. [Online]. Available: http://www.idr.org.ar/wp-content/uploads/2012/02/publicacion_censo2.pdf

[4] I. de Desarrollo Rural. (2014) Censo Provincial de Productores de Durazno para Industria, Mendoza, 2014. [Online]. Available: http://www.idr.org.ar/wp-content/uploads/2015/06/InformeCensoDuraznoInd2014Final.pdf

[5] Dirección de Agricultura y Contingencias Climáticas. (2014) Estadística por Daño de Cultivos. Heladas. Campaña agrícola 2013-2014. [Online]. Available: http://www.contingencias.mendoza.gov.ar/web1/pdf/camp1314hel.pdf

[6] Food and A. O. of the United Nations, *Frost Protection: Fundamentals, Practice, and Economics*, Food and A. O. of the United Nations, Eds. FAO, 2005.

[7] F. Mesas-Carrascosa, D. V. Santano, J. Meroño, M. S. de la Orden, and A. García-Ferrer, "Open source hardware to monitor environmental parameters in precision agriculture," *Biosystems Engineering*, vol. 137, pp. 73–83, 2015.

[8] S. Budi, P. de Souza, G. Timms, V. Malhotra, and P. Turner, "Optimisation in the design of environmental sensor networks with robustness consideration," *Sensors*, vol. 15, no. 12, pp. 29 765–29 781, 2015.

[9] T. Watteyne, L. Doherty, J. Simon, and K. Pister, "Technical Overview of SmartMesh IP," in *International Workshop on Extending Seamlessly to the Internet of Things (esIoT)*, Taiwan, 3-5 July 2013.

[10] T. Watteyne, J. Weiss, L. Doherty, and J. Simon, "Industrial IEEE802.15.4e Networks: Performance and Trade-offs," in *International Conference on Communications (ICC)*. London, UK: IEEE, 8-12 June 2015.

[11] H. R. Bogena, M. Herbst, J. Huisman, U. Rosenbaum, A. Weuthen, and H. Vereecken, "Potential of Wireless Sensor Networks for Measuring Soil Water Content Variability," *Vadose Zone Journal*, vol. 9, no. 4, pp. 1002–1013, 2010.

[12] S. Pohl, J. Garvelmann, , J. Wawerla, and M. Weiler, "Potential of a Low-cost Sensor Network to Understand the Spatial and Temporal Dynamics of a Mountain Snow Cover," *Water Resources Research*, vol. 50, no. 3, pp. 2533–2550, March 2014.

[13] R. Rice and R. C. Bales, "Embedded-Sensor Network Design for Snow Cover Measurements around Snow Pillow and Snow Course Sites in the Sierra Nevada of California," *Water Resources Research*, vol. 46, pp. 1–13, 2010.

[14] J.-p. Qian, X.-t. Yang, X.-m. Wu, B. Xing, B.-g. Wu, and M. Li, "Farm and environment information bidirectional acquisition system with individual tree identification using smartphones for orchard precision management," *Computers and Electronics in Agriculture*, vol. 116, pp. 101–108, 2015.

[15] S. Simoni, S. Padoan, D. Nadeau, M. Diebold, A. Porporato, G. Barrenetxea, F. Ingelrest, M. Vetterli, and M. Parlange, "Hydrologic Response of an Alpine Watershed: Application of a Meteorological Wireless Sensor Network to Understand Streamflow Generation," *Water Resources Research*, vol. 47, no. 10, pp. 1–16, 2011.

[16] Z. Li, N. Wang, A. Franzen, P. Taher, C. Godsey, H. Zhang, and X. Li, "Practical Deployment of an In-Field Soil Property Wireless Sensor Network," *Computer Standards & Interfaces*, vol. 36, no. 2, pp. 278–287, February 2014.

[17] J. Gutierrez, J. F. Villa-Medina, A. Nieto-Garibay, and M. A. Porta-Gandara, "Automated Irrigation System Using a Wireless Sensor Network and GPRS Module," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 1, pp. 166–176, 2014.

[18] T. Ojha, S. Misra, and N. S. Raghuwanshi, "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," *Computers and Electronics in Agriculture*, vol. 118, pp. 66–84, 2015.

[19] B. Kerkez, S. D. Glaser, R. C. Bales, and M. W. Meadows, "Design and Performance of a Wireless Sensor Network for Catchment-Scale Snow and Soil Moisture Measurements," *Water Resources Research*, vol. 48, no. July 2011, pp. 1–18, 2012.

[20] K. Langendoen, A. Baggio, and O. Visser, "Murphy Loves Potatoes - Experiences from a Pilot Sensor Network Deployment in Precision Agriculture," in *International Parallel & Distributed Processing Symposium (IPDPS)*. Rhodes Island, Greece: IEEE, 25-29 April 2006.