



A wireless sensor network for monitoring volcano-seismic signals

R. Lopes Pereira^{1,2}, J. Trindade^{1,2}, F. Gonçalves^{1,2}, L. Suresh^{1,2}, D. Barbosa^{1,2}, and T. Vazão^{1,2}

¹INESC-ID, Av. Prof. Dr. Cavaco Silva, 2744-016 Porto Salvo, Portugal

²Instituto Superior Técnico, Universidade de Lisboa, Avenida Rovisco Pais, 1049-001 Lisbon, Portugal

Correspondence to: R. Lopes Pereira (ricardo.pereira@inesc-id.pt)

Received: 6 June 2013 – Published in Nat. Hazards Earth Syst. Sci. Discuss.: 30 August 2013

Revised: 26 September 2014 – Accepted: 29 September 2014 – Published: 1 December 2014

Abstract. Monitoring of volcanic activity is important for learning about the properties of each volcano and for providing early warning systems to the population. Monitoring equipment can be expensive, and thus the degree of monitoring varies from volcano to volcano and from country to country, with many volcanoes not being monitored at all.

This paper describes the development of a wireless sensor network (WSN) capable of collecting geophysical measurements on remote active volcanoes. Our main goals were to create a flexible, easy-to-deploy and easy-to-maintain, adaptable, low-cost WSN for temporary or permanent monitoring of seismic tremor. The WSN enables the easy installation of a sensor array in an area of tens of thousands of m², allowing the location of the magma movements causing the seismic tremor to be calculated. This WSN can be used by recording data locally for later analysis or by continuously transmitting it in real time to a remote laboratory for real-time analyses. We present a set of tests that validate different aspects of our WSN, including a deployment on a suspended bridge for measuring its vibration.

Temporary deployments of sensor arrays may be performed in order to select the best location for a permanent installation or simply to perform a survey of a volcano. Today's stations can be comprised of a few sensing devices distributed over a small-sized area (with each station covering an area of about 10 × 10 m). The price tag of these sensing devices may limit their placement options due to the risk of physical loss or destruction during a volcanic eruption, as happened in Merapi (Budi-Santoso et al., 2013; Jousset et al., 2013). Complex networking solutions may discourage outside connectivity from being set up for temporary deployments, with data only being stored locally using hard drives or flash devices. In this scenario, the loss of the device will also result in data loss.

In the past few years, there has been an increased research effort in the area of WSN. Acting as distributed data acquisition systems, WSN gather information from the physical world and transmit it to more powerful computers after performing some simple operations. By using small, low-powered computing nodes, WSN are usually simple to deploy and operate. They are being used in a wide variety of scenarios, including environment sensing, military operations or patient health monitoring (Akyildiz and Wang, 2005; Durisic et al., 2012). One of the possible application fields for WSN is volcanic monitoring.

In 2004, a USA research project deployed a small test WSN in the Tungurahua volcano in central Ecuador (Werner-Allen et al., 2006b). During 3 days, data from the active volcano were captured using microphones installed in the MICAz sensing nodes, proving the validity of the approach.

This paper presents the design and implementation of a WSN for volcanic tremor monitoring, created in the context of the MITigate and Assess risk from Volcanic Impact on Terrain and human Activities (MIA-VITA) project. We set

1 Introduction

Volcanologists often use wired arrays of sensors, usually seismometers, to monitor volcanic eruptions and tremor: a low-frequency (0.5 to 5 Hz) seismic signal caused by the movements of magma in the interior of a crater. Volcanic tremor is characterized by a “narrow frequency range or sharply peaked spectra and its long duration compared with earthquakes” (McNutt, 2005). The installation of a sensor array enables seismic tremor to be measured at different places, allowing the location and depth of the magma movements to be estimated (Taisne et al., 2011).

out to design a low-cost, flexible, easy-to-deploy and easy-to-maintain, adaptable WSN for either temporary or permanent monitoring of seismic tremor. In this paper, we describe the challenges we came across and how these were solved in order to reach the goals, while resorting mostly to mainstream commercial off-the-shelf (COTS) components that can easily be procured and replaced in the field. Our system is based on commercially available equipment, such as a single-board computer (SBC) equipped with 802.11 wireless cards and geophones for recording seismic waves. To guarantee a low-cost and easy-to-maintain solution, we used open-source software and, whenever possible, standard protocols. Although our initial goal was to measure volcanic tremor, other analog and digital sensors can be connected to our nodes, enabling their use in other scenarios.

The remainder of this document is organized as follows: Sect. 2 describes the context in which this study was performed and the related state of the art. Section 3 presents the requirements that were defined for the seismic wave monitoring system, and describes the hardware and software architectures and components of our solution. Section 4 details the options taken for instantiating the architecture designs into a physical device and software. Section 5 presents the tests performed to evaluate the proposed solution. Finally, in Sect. 6, conclusions are drawn and future work is presented.

2 Background and related work

2.1 Seismic signal monitoring

In order to understand and predict the behavior of a volcano, it is necessary to gather data from the volcanic ground tremors (Chouet, 1996). A popular model (Chouet, 1992) attributes volcanic tremor to the resonance of the walls of fluid-filled fissures in response to instabilities in the fluid's pressure. The recorded ground motion is often of the surface-wave type, but it can also be formed from body waves if the source is deep (Faria, 2010). To analyze these events, seismic activity caused by the magma movements inside the magmatic chamber may be measured. If detected at an early stage, this can give an early warning of an eruption, enabling proper action from the local authorities (McGuire et al., 2009).

The movements of the magma may occur in any place of the magmatic chamber and, thus, magma may reach the surface in different places. This means that volcanic tremor will be felt with different intensities in the surroundings of the crater. To have detailed information about the complete geographical distribution of the phenomena, sensors must be spread over a wide region and the setup must be easily installed and removed so that the experiment can be easily and quickly repeated in a different region in the vicinity of the volcano. The sensors must cover a region where all the phenomena can be evaluated. Therefore, the coverage area must

comprise, at least, one wavelength of distance between the two sensors farther away from each other. Since both the direction and speed of the tremor wave need to be estimated, at least two sensors at each one of the cardinal points are required.

Typically, seismic data for detecting magma movements in a volcanic area are acquired at 24 bit resolution, using sampling frequencies above 50 Hz (Geoffrey and Welsh, 2010).

2.2 WSNs for environmental monitoring

Research into wireless sensor networks suited for monitoring remote areas for geophysical studies has been performed in the last decade (Yick et al., 2008). In 2004, a small wireless sensor network was deployed on the Tungurahua volcano in central Ecuador as a proof of concept on how these types of networks could efficiently replace traditional monitoring equipment (Werner-Allen et al., 2006b, a). Nodes used an event detection algorithm that, on detection of interesting volcanic activity, triggered reliable data transfer to the base station. Each station consisted of a Moteiv TMote Sky wireless sensor network node designed to run TinyOS (Hill et al., 2000). This research highlighted the benefits of using small, lightweight embedded devices for monitoring remote volcanoes. Although the results were extremely promising, the proposed solution did not fit MIA-VITA use case scenarios, as the nodes used specialized software (TinyOS) and hardware (Moteiv devices and IEEE 802.15.4 radio equipment), making their maintenance in remote locations more difficult and increasing each node cost. Also, the limited resources of each node only allowed the recording of 20 min worth of data, and only transmitted one event at a time. In one of the deployments at the volcano, these properties resulted in the loss of data recordings of a giant explosion. This was due to a smaller, non-interesting eruption that preceded the larger eruption and occupied the network while the larger eruption occurred¹.

Although not used for measuring seismic signals, other Moteiv-based sensor networks have been established to monitor geophysical metrics. The Institute of Automation of the Chinese Academy of Sciences presented a work where multiple nodes were installed in a coal mine to reduce coal mine work-related deaths (Wang et al., 2007). Their proposed system detected current levels of methane, temperature, humidity, and pressure, among others, and when a previously defined set of properties occurred, an event was triggered and transmitted through the network up to a remote location. This option of only transmitting interesting events allows the reduction of network resources used and saves node batteries. A combination of a star and peer-to-peer network topologies was researched, checking which was more adequate for coal mine use case scenarios.

¹This event can be read about in the online magazine <http://www.networkworld.com/news/2006/062606widernet-volcano.html> (retrieved 30 May 2013).

To monitor environmental conditions in petroleum extraction facilities and oil rigs, researchers from Dalhousie University and Cape Breton University implemented and deployed a WSN where a heterogeneous architecture was partially composed of COTS equipment (Johnstone et al., 2007). Contrary to the previous two works, in this proposed WSN, some sensor nodes were Moteiv TMote devices, while other, more powerful nodes, were Acorn RISC Machine (ARM) devices running the Linux kernel. This demonstrated some of the benefits of deploying this type of network using more standard equipment.

The Massachusetts Institute of Technology and the Center for Embedded Networked Sensing developed a self-calibrating distributed sensing platform for acoustic sensors using embedded devices running Linux (Girod et al., 2006). As nodes are spread over a wide geographical area, they are capable of pinpointing the location of a sound source. This functionality required collected samples to be synchronized, and so the proposed platform presented two options: place the *timestamp of interest*, i.e., the time at a pre-defined node, in a network packet, flood the network with this packet, and use that timestamp as the “local time” on every hop through the network, or use a global time service on a node with the Global Positioning System (GPS), and broadcast this time to the rest of the network hop by hop. The first solution provides times relative to the *timestamp of interest*, while the second option makes all nodes synchronized to a global clock. In both cases, an overhead for synchronization messages is imposed on the network.

More recently, a wireless mesh sensing network that is able to give an early warning signal on the event of an earthquake was presented (Fischer et al., 2012). This system is comprised of several nodes, deployed in a small geographic area, that measure the intervals between the arrival of the P and S waves. Through the analysis of these values, the distributed system is able to emit an alarm, with a few seconds of anticipation, that the more destructive waves of an earthquake are about to arrive. Nodes were implemented on x86 single-board computers, increasing the power consumption when compared to an ARM architecture.

3 Architecture

This section presents the architecture of the WSN created for monitoring seismic tremor, and is divided into four parts. First we will detail the requirements we set out to fulfill. These are followed by a description of the global architecture. In the third part, we will present the architecture of each node, describing the several modules that comprise it. Finally, we will introduce the monitoring application that allows users to visualize the collected recordings and monitor the network state, either remotely or in the field.

3.1 Requirements

In the context of the specific requirements associated with the MIA-VITA project, there is a set of characteristics that the solution’s architecture should accomplish. In broad terms, these characteristics are flexibility, adaptability, ease of deployment and maintenance, low cost, low maintenance, allowing for temporary or permanent deployments, use of a common clock reference for timestamping samples, and the provision of easy access to data.

A flexible solution should enable different network topologies to be easily deployed. It should be equally simple to deploy a WSN with 4 or 14 nodes. The supported distance between nodes should be sufficiently large to enable different scenarios. A minimum distance of 40 m between nodes was defined as the goal. It should also be possible to record the collected data locally and/or to transmit them to a remote laboratory. There should be *flexibility* about the network technologies that can be used for transmitting the data.

Although the WSNs were originally intended for use with geophones, they should be *adaptable* to use different types of sensors, such as gas sensors, thermometers or video cameras. Nodes should have enough capacity to allow easy development of basic signal processing software to be run on each node, allowing for adaptation to different scenarios.

Ease of deployment should be targeted, as personnel installing equipment in the field should not be required to have expertise in embedded systems or wireless networks. After deployment, nodes and the network should automatically self-configure. Field personnel should have access, in the field, to tools that allow them to verify the correct operation of the installed WSN.

Nodes may be damaged or malfunction while deployed. The failure of one node must not prevent nodes from communicating with each other. *Ease of maintenance* is required, namely the existence of mechanisms that enable the automatic recovery of the network in case of node failure. Failed nodes should be easy to repair using COTS parts easy to procure worldwide.

The solution should be *low cost*, so that a complete WSN deployment is affordable. This will allow more volcanoes to be monitored, increasing safety for populations. Also, several WSN deployments on the same volcano become more affordable. Low cost of the equipment is also important, as nodes can be destroyed as a consequence of vandalism, by animals or by direct damage from environmental hazards. This is more important in temporary deployments, where the limited time span does not allow for the construction of infrastructure capable of protecting the nodes. It is also important to provide a solution that has *low maintenance*, especially for deployments in remote locations, where equipment may not be easily accessible.

Temporary or permanent deployments have different requirements. For a temporary deployment, nodes should be light and easy to carry but also robust, as little time is

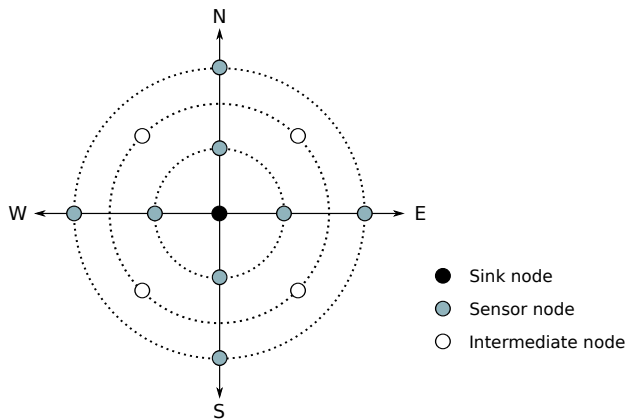


Figure 1. Example of WSN topology. Adjacent nodes may be up to 1 km apart if the line of sight is available.

available to invest in adequate protection. For a permanent deployment, nodes may be sheltered and more powerful, and reliable power sources must be made available. These requirements have to be balanced in order to provide a single, all-round solution.

In order to compare the seismic tremor arrival time at each sensor, all samples must be timestamped using a *common clock*. This is required not only among nodes that comprise a WSN, but also among different WSNs.

Recorded *data should be easy to access*. Either in the field or in a remote laboratory, recorded data should be easy to access, both in real time and post event. In the case of temporary or remote deployments, it might not be feasible to provide remote access from a volcanological laboratory. The WSN should be able to store data for a significant period of sampling time.

3.2 WSN architecture

Seismic signals are collected at the remote volcanic location by a sensor array. A single special node in the sensor array, the sink node, which will be detailed later, is then responsible for gathering all the data and transmitting them to a remote location, e.g., using a satellite gateway. It is from this point that collected samples are sent to the remote volcanological laboratory. There, specialized personnel are able to analyze the data and produce scientific predictions based on the current status of the volcanic event.

Geographically, the proposed topology for standard experiments using the developed WSN array is presented in Fig. 1.

In this topology, each node can perform one of the following three roles.

The sink node is located in the center of the topology in order to reduce the maximum number of hops that a message from any source has to take in order to reach it. Only one sink node may exist at a particular time instant. A node playing this role has a critical impact on the network, as all other

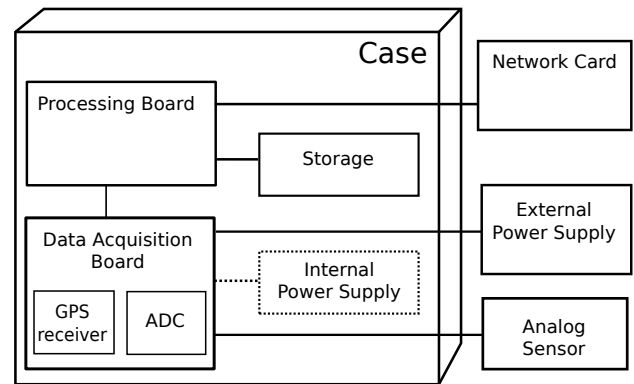


Figure 2. Node hardware architecture – components represented by dotted lines are optional.

nodes transmit the collected data to it. It is therefore crucial for this node to be less exposed to damage caused by the various natural hazards present in a volcanic region.

Nodes that perform the intermediate role are normal sensor nodes, but their specific location is chosen in a manner so as to make them able to substitute adjacent nodes should they fail. It is their purpose to provide a backup link in order to guarantee continuous communication with the sink node. For a node to be considered an intermediate one, its wireless communication radius has to encompass at least two other nodes.

Finally, nodes can have the role of sensor nodes. These devices only acquire data from their sensing devices, and transmit the collected information to the sink node or to another node that is on the path to the sink node. Sensor nodes will relay data towards the sink for other nodes that are unable to reach the sink directly.

The proposed base topology can be extended with extra sensor nodes in the extremities for an increased range. It is important to notice that as the number of hops from a sensor to the sink increases, so does the delay in data packets arriving at the sink node.

3.3 Hardware architecture

The physical elements that constitute each node in the WSN are presented in Fig. 2. Components are divided into two main groups: those located in the case, and the external components.

The five components present inside the case are the processing board, the data acquisition board, the global clock, the local storage and the internal power supply. The processing board is the element responsible for interacting with the remaining elements, receiving data from the data acquisition board, transmitting/receiving data to/from the network cards and executing the various software components developed for the system. The data acquisition board connects to the seismic sensors, periodically retrieving samples using an

analog-to-digital converter (ADC). It also houses the GPS receiver and the voltage conversion circuitry. The samples collected are stored in the storage component. The last element present in the case is the internal power supply, a battery. Its main use is to support intervals where the external power component is unplugged from the case.

Outside the case, three elements are to be found: the network card, to transmit/receive data to/from other nodes, the external power supply responsible for powering the remaining elements for long periods of time, and the seismic sensors.

3.4 Software architecture

Each node is composed of the software components shown in Fig. 3. Software components can be executed in user space or kernel space. Kernel space is strictly reserved for running the kernel components and device drivers.

In our architecture, there are two applications that run in user space, the monitoring application and the data manager. All remaining software components are executed in kernel space, as this allows for lower latency and direct access to the hardware. All these software components will be detailed in the following sections.

3.4.1 Controller

The controller is the central software module present at each node. It receives samples from the communication manager (from other nodes for relaying or storage) or from the sample acquisition modules and, depending on the node's role in the network, either sends data samples to the communication manager, for delivery to the sink node, or to the data manager for persistent storage in the case of the sink node.

One other function of the controller module is the aggregation of data samples. To reduce the number of packets sent through the network and, consequently, to reduce the power consumption, each node aggregates groups of data samples. Samples are aggregated by destination address, which for all samples is the sink node. This ensures that aggregated packets are all destined to the same node, which will ease the computation complexity of the desegregation protocol. Aggregation can be done at two distinct levels: application and network level. Application-level aggregation aggregates protocol data units (PDUs), while network-level aggregation aggregates Internet Protocol (IP) packets.

Desegregation functionality is separated from aggregation to provide extra flexibility. This way a node can desegregate traffic without needing to load the aggregation interceptor, and vice versa. The desegregation interceptor will separate packets inside an aggregated IP packet and inject them into the node's network stack.

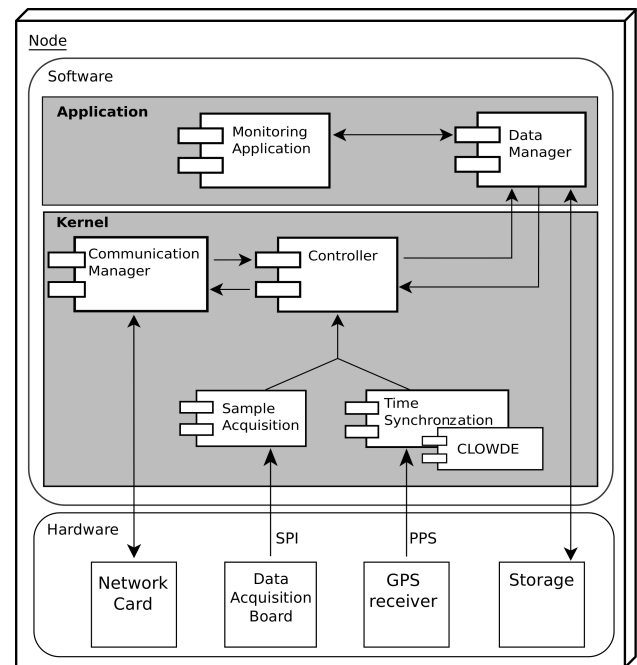


Figure 3. Node software architecture.

3.4.2 Data manager

The data manager module is responsible for storing and retrieving samples from a non-volatile medium (in our case a USB pendrive or disk). It can only receive input samples transmitted by the controller module. Although this happens in all node roles, it is more important for the sink node, where samples gathered from the whole network are stored. The data manager is able to retrieve stored samples, when asked by the monitoring application, via two formats. The first is a compressed, space-efficient, non-standard binary representation of the samples. The other format represents samples through a standardized verbose JavaScript Object Notation (JSON) format that can be easily read by humans or computers. An open-source exporter capable of transforming both these formats into miniseed has been developed² in order to allow interoperability with other systems.

3.4.3 Sample acquisition

Seismic tremors are sensed using geophones that produce an analog signal. A sample acquisition system is required in order to convert the analog signals into digital data. Samples must be collected at a constant rate, which has to be more accurate than what can be achieved with a software-controlled solution. Thus, a hardware solution had to be designed using an oscillator driving an ADC. The oscillator provides a drift of less than 50 ppm. Whenever a sample is available, an

²Available at https://github.com/cnm/mia_vita.

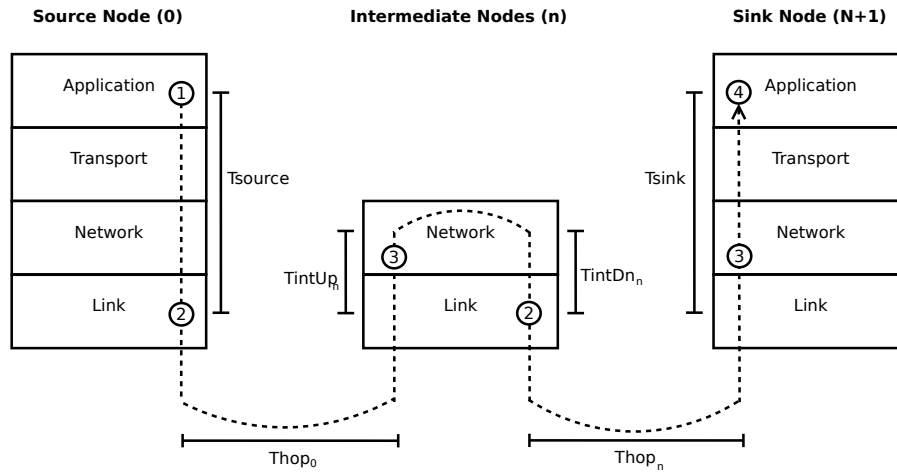


Figure 4. Time synchronization time retrieval points.

interruption is raised, triggering the sample acquisition module to collect the data.

3.4.4 Time synchronization

Samples take different amounts of time to arrive at the sink node from the time they are created. Time will vary due to central processing unit (CPU) contention, network contention and number of hops. Unlike wired sensor arrays, this precludes the sensor from simply timestamping samples using its own clock. Our solution was to equip each sensor node with a GPS device, providing highly accurate time synchronization within the WSN. The GPS device is utilized for two purposes: to provide the controller module with a time reference to timestamp samples and to synchronize the ADC itself. Every 1 s, the ADC is reset using a pulse per second (PPS) signal from the GPS receiver, thus limiting the impact of oscillator drift. The PPS signal is generated with a 50 ns accuracy.

3.4.5 CLOWDE

Under certain circumstances, it may not be possible to use GPS on all the nodes, e.g., due to heavy tree cover or because we wish to place some of the nodes underground, in caves or tunnels. In order to enable the use of our WSN under these circumstances, we decided to develop a delay estimating algorithm capable of approximately calculating the time it takes for data to go from the application that created them, through the various hops in the network, up to the application present on the sink node. This algorithm, named cross-layer one-way delay estimation (CLOWDE), enables our system to be used even when some of the nodes are unable to receive a GPS signal. Only the sink node requires the use of GPS. This also ensures that different WSNs share a common clock. In our proposed solution, each node calculates the time it contributes in delaying the messages, and then sends this value to

the next node; each one of the intermediate nodes computes the accumulated delay experienced by the message since it was created until it is received by the next node; and finally, the sink node uses its reference time (obtained using GPS) and subtracts the accumulated time to estimate the creation time.

A schematic representation of the algorithm is presented in Fig. 4. As the first step (1), after capturing a sample, the application creates a message at the source node (T_{creation}) containing the data to be transmitted to the sink. Then, the application sends the message, which must travel down the IP stack until the Link layer pushes it to the network card (2). This period of time is identified as T_{source} . The network card will then transmit the packet to the next hop. This period of time, T_{hop_0} , comprises the propagation and transmission delay. Finally, the packet reaches the sink node (3), where it will travel up the IP stack until it is delivered to the destination application (4), taking an additional T_{sink} . Should there be intermediate hops involved, there will be three additional times for each hop n along the way. First, the packet must be delivered to the IP layer (3), where the forwarding decision is performed ($T_{\text{int}} U_{p_n}$); second, it is pushed down to the Link layer (2), taking the value of $T_{\text{int}} D_{n_n}$; and third, it is sent over the air to the next hop (T_{hop_n}).

Considering a path with $N + 1$ nodes, T_{creation} , the instant where the sample was captured is given by

$$T_{\text{creation}} = T_{\text{reference}} - \sum_{n=0}^{N+1} (T_{\text{hop}_n} + T_{\text{int}} U_{p_n} + T_{\text{int}} D_{n_n}), \quad (1)$$

where

$$\begin{aligned} T_{\text{int}} U_{p_0} &= 0, \quad T_{\text{int}} D_{n_0} = T_{\text{source}} \\ T_{\text{int}} U_{p_{N+1}} &= T_{\text{sink}}, \quad T_{\text{int}} D_{n_{N+1}} = 0. \end{aligned}$$

The time packets spent inside each node can be measured using that node’s SBC local internal clock. This way, T_{source} , T_{sink} , $T_{int} U_{p_n}$ and $T_{int} D_{n_n}$ may be determined by intercepting the packet and retrieving the local time at key places in the kernel and in the application. Although the local clock may drift, the time intervals are small enough (in the range of u_s) for this to be ignored. Determining the time it takes to transmit the packet over the air requires a different approach, as we have no guarantees about the clocks of different nodes being synchronized.

The estimation of these values depends on the protocols used, mainly at the Link layer, due to the specific characteristics of the medium access technology used. In our use case, communication is performed using WiFi (IEEE 802.11b), working in distributed coordination function (DCF) mode without using a request to send/clear to send (RTS/CTS) mechanism. We believe this to be an interesting case, due to the wide range of situations it covers. Thus, we will describe how time is estimated using 802.11b interfaces.

Figure 5 illustrates the phases of a successful transmission using WiFi (IEEE 802.11), working in DCF mode without using RTS/CTS frame collision reduction mechanisms. A frame is transmitted, and an acknowledgement (ACK) packet is received. We need to estimate the time it takes from the instant the sending node starts the transmission (tA) until the receiver nodes receives the full packet (tB). Equation (2) presents how this value is computed:

$$T_{hop} = tB - tA = DIFS + T_{tx} + T_{prop}, \quad (2)$$

where DIFS is the distributed coordination function (DCF) interframe space time interval, T_{tx} is the transmission delay, and T_{prop} is the propagation time.

DIFS has a constant value of $50 \mu s$ in IEEE 802.11b. The propagation delay is simply the distance between nodes divided by the speed of light ($3 \times 10^8 \text{ m s}^{-1}$). The transmission delay can be obtained if the transmission rate is known, as it is simply the frame length divided by the transmission rate. However, it must be taken into account that parts of frame preamble are transmitted at slower (1 Mb s^{-1}) speeds.

Although the air time of a first-try successful transmission can be estimated, the same does not hold true of situations with initially busy channels, transmission errors or collisions where the frame has to be retransmitted. In this situation, the sender performs an exponential backoff delay before attempting a new retransmission. However, the wait period is not deterministic, and is usually performed by the network card hardware, making it very hard to measure. This only allows us to estimate the air time of those frames that did not need to be retransmitted. Packets need to have a sequence number, and we need to know which packets were retransmitted and which ones went through every hop with a single transmission.

This limitation, although important for some use cases, does not have a noticeable impact in our scenario, where nodes continuously produce data at a constant rate defined

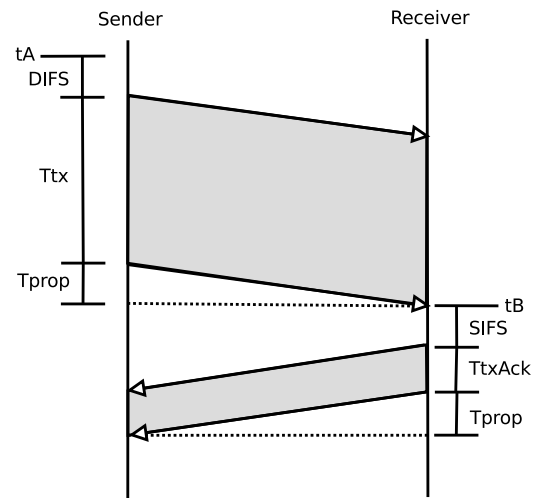


Figure 5. Successful packet transmission in 802.11 networks.

by the oscillator driving the ADC. Even if we are unable to determine the air time for some packet, the time at which it was created can be fairly accurately approximated by interpolating it with the previous and next packet at the sink node, as each node produces samples at a known rate, and clock skew between two samples will be minimal.

CLOWDE time synchronization works differently from the GPS synchronization we employ in our sample acquisition board. Using the GPS signal, the ADC in each node is periodically synchronized with all others using a common time reference (the GPS clock). As such, each node will collect its samples at the same time (approximately). If no GPS signal is present, each receiver’s internal clock will drift, and the PPS signals will not be synchronized over all the nodes in the WSN. Oscillators will be allowed to drift and different nodes will collect samples at different instants. CLOWDE will enable samples to be timestamped, but these will have been collected by each node at different times.

The accuracy afforded by CLOWDE is evaluated in Sect. 5.2.

3.4.6 Communication manager

To disseminate data from sensor nodes to the sink node, we required the use of a routing protocol with low energy consumption due to the limited battery from the node, standardized and compatible with the transmission control protocol (TCP)/IP stack to keep the development time low. The routing protocol should also support node failure and be able to calculate the best routes according to the wireless signal strength. Surveys regarding this subject are presented in Mogaibel and Othman (2009), Akyildiz and Wang (2005), and Al Basset Almamou et al. (2009).

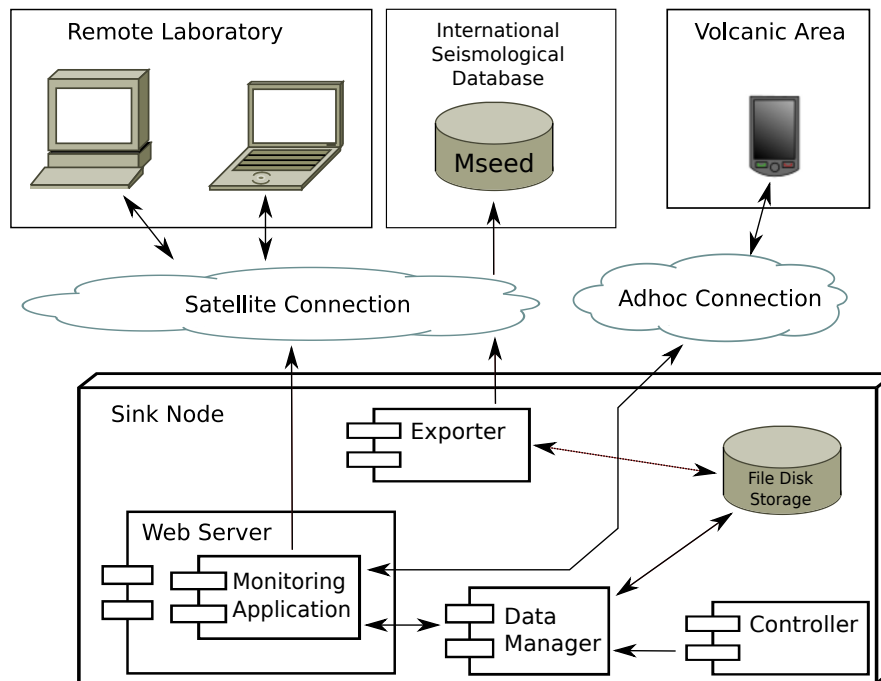


Figure 6. Monitoring application architecture.

Better Approach to Mobile ad hoc Networking (BATMAN) (Johnson et al., 2008)³ is a routing protocol for multi-hop ad hoc mesh networks. This protocol's main focus is the decentralization of knowledge regarding the best routes through the network, resulting in no single node having all the data. BATMAN acts as a distance-vector protocol, and does not try to determine the complete route. It uses a hop-by-hop routing strategy. To spread topology information, every node periodically sends out a broadcast with the objective of informing all its neighbors about its existence. The neighbors then relay this message to their own immediate neighbors. This operation carries information to every node in the network. In order to find the best way to a certain node, BATMAN registers the originator messages and logs from which neighbor the message was received. Under real-world conditions, it was shown that BATMAN exhibits high levels of stability but slightly slow convergence times (Abolhasan et al., 2009).

One of the main benefits of BATMAN is that its implementation is small and simple. Besides requiring very little processing power for its operation, it was relatively easy to patch the BATMAN open source code from the x86 environment to the ARMv4 CPU architecture. Also, the way BATMAN was implemented for the Linux operating system (OS) continues to allow the use of Netfilter kernel hooks for packet processing. This was extremely important for this project, as it meant the routing protocol could be developed completely

independently of the time synchronization system. Finally, BATMAN enables the network to auto-adjust if some node ceases to work. As long as the network still possesses a functioning node equipped with a GPS device, there is support for replacing the sink automatically, if this node happens to become damaged.

3.5 Monitoring application

The analysis and processing of seismic data require special tools (Claerbout, 1997; Kurin, 2007; Murillo and Bell, 1999; Rodriguez and Sacchi, 2011). Traditionally, these tools, although powerful, require a high degree of parametrization in order to obtain meaningful results. Also, due to their complexity, it is common for them to require a high amount of computation, taking tens of minutes or even hours until data can be visualized. These properties are not an obstacle to specialized analysis in near real time (Scognamiglio et al., 2009), or are performed on historical data where no real-time requirements in visualizing collected data exist. In MIA-VITA's use case requirements, we listed that the monitoring system should enable personnel in the field to be able to observe quickly and effectively the status of the various nodes collecting data, and of the network.

With this requirement in mind, the proposed monitoring application is composed of four components, as depicted in Fig. 6. The web server is the system that delivers the web pages requested by the clients using Hypertext Transfer Protocol (HTTP). The chosen web server application was

³<http://www.open-mesh.org>

Table 1. Technical specification – case.

Case	
Enclosure dimensions	22 × 14 × 5.5 cm
Weight	1.2 kg
<i>Processing board</i>	
Board	TS-7500
Processor	Cavium Networks ARM9 CPU
Processor speed	250 MHz
Memory	64 MB DDR
Internal storage	4 GB MicroSD
Watchdog	Yes
Network card	Ethernet 100 MB
<i>Data acquisition board</i>	
ADC model	Texas Instruments ADS1278
ADC effective resolution	19 bits
ADC operation mode	Low power
Input voltage range	±2.5 V
Number of channels available	8
Number of channels used	4
Input impedance	24 kΩ
Sample rate	50 Hz
<i>Storage</i>	
Data storage	USB flash 16 GB
<i>Global clock</i>	
GPS	Trimble Condor C2626
<i>Internal power supply</i>	
Voltage converter	TRACOPOWER – TEN 8-1211
Input range	9–18 V
Output range	5 V ± 1 %
Internal battery	12 V 0.8 Ah lead acid
Internal battery weight	260 g
Internal battery dimensions	9.6 × 2.5 × 6.2 cm
<i>Input/output</i>	
Sensor input	Geophone proprietary plug
USB input/output	Type A female
Status output	4 × light-emitting diode (LED)
Status enable	1 × interrupter
Power on/off	1 × interrupter
Power input	AC plug 4.0 × 1.7 mm

lighttpd⁴, due to its optimization for speed-critical environments with small memory resources, being standards compliant and historically having few security problems. The monitoring application is responsible for creating the web pages through which users can interact with the system. It consists of a set of javascript functions that, upon client request, are transferred to client browsers and then executed locally. This option allows the sink node to support several clients, with a very low overhead imposed on the system. All computation of the graphical elements in the interface is distributed to the clients. The client component is a modern web browser (e.g., Internet Explorer 8, Firefox 3, Google Chrome 16 or more recent) that is able to execute javascript commands. As the generation of the graphical interface was based on

⁴<http://www.lighttpd.net/>

Table 2. Technical specification – external components.

External power supply	
<i>Battery</i>	
Type	12 V 12 Ah lead acid
Dimensions	9.3 × 9.8 × 15.3 cm
Weight	2.9 kg
<i>10 W solar panel (optional)</i>	
Peak power	10 W
Dimensions	28.5 × 37 × 0.15 cm
Weight	0.4 kg
<i>20 W solar panel (optional)</i>	
Peak power	20 W
Dimensions	36 × 50.5 × 0.15 cm
Weight	0.7 kg
Network card	
<i>WiFi card</i>	
Model	PowerLink Max Distance
Frequency	2.4 GHz
Protocols supported	IEEE 802.11b/g
Maximum distance	1500 m (line of sight)
Height	60 cm
Analog sensor	
<i>Tri-axial geophone</i>	
Model	Mini Seis-Monitor geophone
Sensitivity range	3.0 to 50.8 mV m ⁻¹ s
Dimensions	15 × 15 × 8 cm
Weight	4.5 kg

standards, it functions on desktops or in mobile systems such as smartphones or tablets.

The developed monitoring application is able to provide users with three main types of information. First, it shows graphical plots with information on the most recent data acquired by the multiple sensor in the network. This functionality is shown in Fig. 7a. Figure 7b indicates the location of various sensors in the network over a geographical map of the region. Finally, Fig. 7c provides personnel in the field with a quick overview of the current network status.

4 Implementation

In this section, we will detail the options taken for instantiating the architecture designs into a physical device and software. The following sections will describe the SBC, the power supply, the WiFi equipment and the case. Tables 1 and 2 summarize the technical details for the main hardware components.

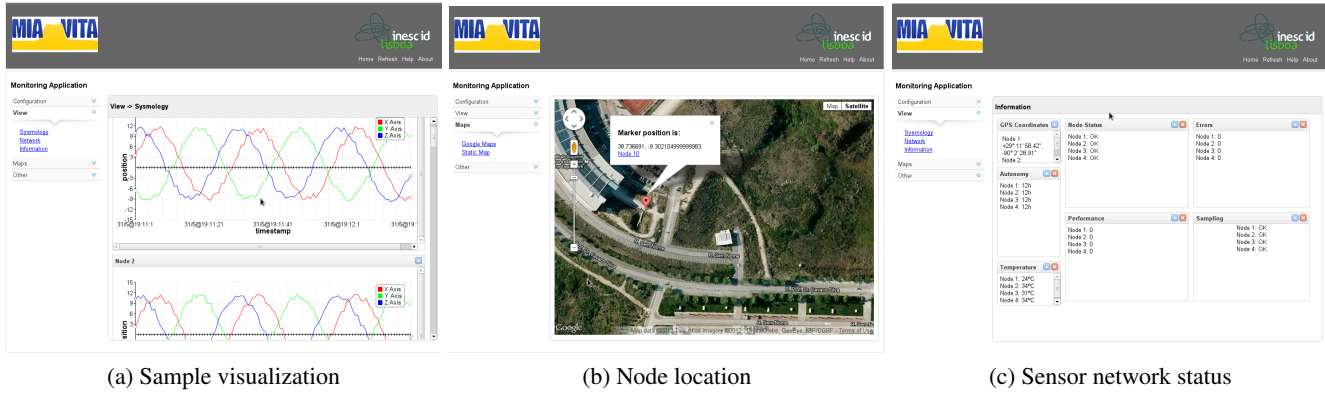


Figure 7. Monitoring application screenshots.

4.1 Single-board computer

The chosen processing unit to be installed in the multiple nodes was the TS-7500, an ARM-based embedded device supplied by Technology Systems⁵. This device is small, measuring 66.6 × 74.3 mm, and is very light (less than 50 g). It is equipped with a 250 MHz ARM version 9 CPU and has 64 MB of RAM available for the OS. Regarding inputs and outputs, it has the Serial Peripheral Interface (SPI) and USB buses that are used to connect external hardware. This type of equipment was chosen mainly due to its low power consumption (400 mA at 5 V) and its option to run a vanilla Linux kernel, increasing the standardization of the proposed solution. The complete OS and our software are executed from a micro SD card that allows for easy substitution if an update is required.

Data acquisition board

The Mini Seis-Monitor tri-axial geophone was selected as the sensor to measure volcanic tremor. This produces an electrical signal in response to ground motion. We built a custom board for analog signal acquisition using a four-layer printed circuit board (PCB) and surface-mount devices (SMDs). Figure 8 shows the main components of this board and their interaction with the SBC. The board serves several purposes: it provides an eight-channel ADC; it integrates a GPS receiver that provides a reference clock and allows nodes to be located; it provides a high-power USB port; it allows four LEDs to be used to convey information to the users; and it powers the SBC. This board is connected to the SBC using a 44-pin header, from which we obtain access to interruption lines, serial port, SPI bus and digital input/output (DIO) lines.

Sample acquisition is performed by a Texas Instruments ADS1278 eight-channel simultaneous acquisition 24 bit ADC. The ADC is driven by a 3.6864 MHz clock, producing

⁵<http://www.embeddedarm.com/products/board-detail.php?product=TS-7500>

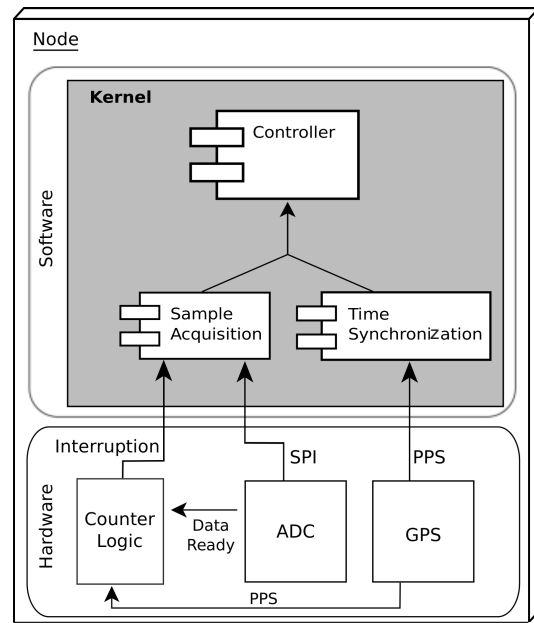


Figure 8. Data acquisition board.

samples at a rate of 7200 Hz. In order to accommodate the GPS time synchronization, only one sample for every 144 is used, resulting in a sample collection rate of 50 Hz. This rate can easily be changed, and we have experimented with success rates above 1 KHz, but these were not required for our use case. When a sample counter reaches 144, an interruption is raised, causing the Sensor Acquisition kernel module on the SBC to read the sample using the SPI bus. When a GPS device is present, the sample is timestamped using the time provided by the Time Synchronization kernel module.

The GPS device needs only be present in the sink node in order to provide the WSN with a reference clock. It is optional for the collector nodes. When the collector nodes do not have a GPS device installed, which represents about one fourth of the node’s cost (without the geophone), the

CLOWDE algorithm is used. When a GPS device is present, a highly accurate timestamp is created for each sample at the collector node. Trimble Condor C2626 GPS receivers were used. These devices are controlled using a serial port, which provides times with accuracies within a tenth of a millisecond. As we required greater precision, we also used a PPS line serving a dual purpose: reset the ADC and raise an interruption on the SBC. The PPS signal provides an accuracy of 50 ns. The ADC is thus reset every 1 s, ensuring that all nodes sample at the same time. After a reset, the ADC does not output any data for 129 cycles, which is why we only use one sample for every 144 samples, in order to achieve a precise 50 Hz sample rate. The interrupt raised by the PPS signal enables a kernel module to set the SBC internal clock every second. Samples within each second are timestamped using the internal clock, which is not expected to drift much during a single second. Taking into consideration the interrupt and syscal measurements performed, we estimate the timestamps to be accurate within $100 \mu\text{s}$. The GPS device also allows the node's location to be known.

By using 4 DIO lines to drive LEDs, the same board allows us to convey some status information to the user. The LEDs are placed on the outside of the node's case, and their use is described in Sect. 4.4.

The last function of this board is to provide USB devices with more power than the SBC could. We experienced difficulties using long-range WiFi USB cards, which the SBC could not provide adequate power to. As such, we intercepted a USB port from the SBC, replacing the power lines from the SBC's USB port with direct connections to the switching power supply used to power the node.

4.2 WiFi equipment

For the WiFi equipment, we have opted to use the *PowerLink PT-H9DN-ROC* USB card. This type of equipment is weather sealed, equipped with an omnidirectional antenna and capable of communicating over distances of 1000 m if a line of sight exists between the two hops. By default, the connecting cord is long, 1.5 m, and as such allows for some freedom in installing the antenna for better positioning to improve communication with the rest of the network nodes. Although there is support for the 802.11b, 802.11g and 802.11n protocols, in our devices, we force the corresponding kernel driver to use the 802.11b protocol, as this is the sole option parameter that correctly supports ad hoc mode operation. This behavior is due to limitations of the open-source driver used.

4.3 Power supply

To power each node, we decided to use 12 V lead-acid batteries. 12 V is the tension used in automobiles and motorcycles; as such, batteries of different sizes and capacities are available worldwide. There is also a wide selection of solar panels and wind-powered generators. Given the deployment

flexibility we were aiming for, we decided to equip each node with a small 0.8Ah battery. As a fully operational node consumes 4.25 W, the internal battery is able to power it for a little over 2 h. The internal battery also enables the node to keep on operating while other power supplies, such as an external battery, are changed.

For temporary deployments during daytime, we have used solar panels rated to 10 W peak power from NASA Marine. We have used them successfully in sunny weather in September in the Lisbon region. Their light weight and dimensions make them convenient to carry.

The external battery is able to power a node for a little over 1 day. We have used lead-acid batteries that are relatively heavy but much cheaper and easier to procure. Other technologies enable lighter batteries. For instance, a higher capacity (15 Ah) LiFePO₄ battery provides a 1.5 kg reduction in weight, but its price is still tenfold the EUR 25 of the ones we used.

The external battery can be combined with solar panels for a permanent or long-term installation. The capacity of the solar panels will have to be calculated according to the latitude of the deployment. A permanent deployment, required to run all year long at high latitudes, including on the shorter and cloudier winter days, will require a higher capacity than a deployment covering only the sunny, long days of summer months at lower latitudes. We have run multi-day deployments powered by the sun. These are presented in Sect. 5.4.

4.4 Case

The several components present in each node are installed in a rectangular cuboid case made of aluminum and measuring $22 \times 14 \times 5.5$ cm. This material was chosen due to its ability to resist the impacts inflicted while transporting the material to the target location and due to the fact that it is weather proof. As the case is made from a conductive metallic material, we connected the case to the GND signal received from the power supply to isolate the GPS from the rest of the electronic equipment.

On the front of the case, there are the following inputs, as shown in Fig. 9:

- one button for enabling/disabling LEDs to save energy if not in use;
- two green LEDs for indicating “Power On” and if the GPS device has a satellite lock; and
- two yellow LEDs, one indicating an internal fault and the other warning that the system is booting.

The back of the case has the following outputs, as shown in Fig. 10:

- a geophone proprietary plug to connect the device to external sensors;



Figure 9. Case front.



Figure 10. Case back.

- a male USB adapter to connect to an external WiFi antenna;
- a power adaptor to connect to a battery or solar panel; and
- a power on/off switch for the whole device.

An image of the interior of the device's case is shown in Fig. 11. In the image, the following components are highlighted: (a) a button for enabling/disabling the LEDs, (b) LEDs, (c) internal battery, which can be replaced with an external hard drive for sample storage if the user desires, (d) USB connection to an external WiFi card, (e) an Ethernet port, (f) a TS-7500 embedded ARM device, (g) a GPS antenna, (h) a data acquisition module, (i) geophone connector, (j) a power connector, and (k) a power switch.

Finally, in Fig. 12, we show the complete hardware case, including the geophone sensor and the external WiFi antenna, installed in the terrain.

5 Evaluation

We believe that the proposed architecture has accomplished the requirements specified in Sect. 3.1.

The communication manager component, with the use of the BATMAN protocol, allows for a flexible, dynamic number of nodes to participate in the WSN. The long-range WiFi cards used were verified to enable nodes to be up to 1000 m apart when the line of sight is present, exceeding the 40 m requirement. This provides *flexibility* in deployment in different scenarios, using different numbers of nodes and distances between them. Connection to external entities, like a remote

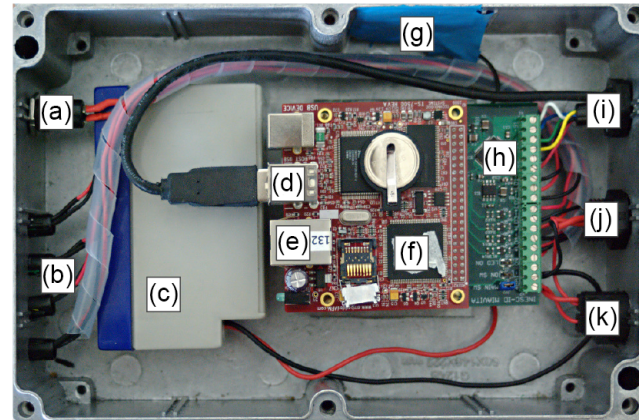


Figure 11. Open case – (a) button for enabling/disabling status LEDs, (b) status LEDs, (c) internal battery, which can be replaced with an external hard drive for sample storage if the user desires, (d) USB connection to an external antenna, (e) Ethernet port, (f) TS-7500 embedded ARM device, (g) GPS antenna, (h) data acquisition module, (i) geophone connector, (j) power connector, (k) power switch.



Figure 12. Node deployed in the terrain with wireless antenna installed on the ground and a geophone sensor.

volcanological laboratory, is facilitated by the *flexibility* provided by the use of the Debian GNU/Linux OS: many networking protocols and network devices are supported out of the box. Connectivity is only limited by the available interfaces: Ethernet and USB ports. The use of USB storage devices enables a large choice of options for data storage, from a flash drive for low-power limited storage to a hard disk drive, which can handle several years of raw data. The use of 12 V power supplies provides *flexibility* in the choice of battery and solar panel capacity, enabling cost, weight and autonomy to be adjusted to match each deployment scenario.

The data acquisition board is *adaptable*, as it allows up to four different analog sensors to be connected. It can be upgraded to use up to eight. We can also connect digital sensors using SPI or USB. Other devices such as video or photographic cameras can also be connected through USB. The use of the Debian GNU/Linux OS allows software to be easily developed without requiring extensive WSN knowledge. The SBC used is powerful enough to run reasonably complex software, allowing some processing to be performed on the WSN nodes if necessary, thus reducing the required bandwidth.

The WSN is *easy to deploy* by non-specialists, thanks to the network self-configuration abilities. This task is further simplified by the monitoring application that enables local or remote verification of the network's operation.

Maintenance is simplified by the network design and use of the BATMAN routing protocol. Should a node fail, the redundant design will enable data flows to the sink (the unique single point of failure) to be rerouted using a different path. Only data from the sensors connected to the failed node will be lost.

Node *maintenance* is also made easier by the choice of COTS components. Except for the custom data acquisition board, all components can be easily procured and replaced. By accepting 9 to 18 V input, nodes allow mass-made, easy-to-find power supplies, such as car batteries or solar panels, to be used. These options also contribute to the *low cost* of fabrication and maintenance. Our current prototypes, described in the implementation section, cost about EUR 350 each to produce. This value is expected to drop significantly for larger production runs. This cost does not include battery, solar panel or sensor devices such as geophones.

The packaging solution chosen is durable enough for *permanent* installation and robust enough for *temporary deployments*. The case size is compact when compared to the geophones that it will be paired with, while being sufficiently spacious for installing hard disks for long-duration recordings of data where communication with a remote laboratory is not possible, or housing and protecting batteries sufficient for temporary deployments. The use of portable solar panels allows long-duration or even permanent installations to be performed easily.

The use of a GPS receiver in every node ensures *time synchronization* among the different nodes of the WSN, among

different WSN deployments and with other data sources. The CLOWDE protocol enables operation in locations where GPS reception is poor or impossible, such as dense forests, underground or indoors. With CLOWDE, only the sink node is required to have a GPS signal, in order to provide a common time reference with other systems.

Recorded *data are easy to access*. As data are stored in USB mass storage devices, these can simply be removed and plugged into any computer. The myriad of remote connectivity choices enables data to be conveniently transmitted to a remote laboratory. In the field, the sink node builds an ad hoc network for management, enabling any laptop or tablet to be connected, being auto-configured. This, together with the HTTP server and the monitoring application, allows data to be easily and wirelessly visualized or downloaded in the field.

In the rest of this section, we present a set of experiments designed to validate key aspects of our WSN: time synchronization among the nodes, the network's ability to recover from node failures, power supply autonomy, and the ability to record signals in different scenarios.

5.1 GPS synchronization

In our system, each sample is marked with two different time references. Every second, the drift of the internal clock of the SBC is determined in regard to the PPS signal of the GPS receiver. At the same time, the ADC, which is driven by an oscillator, is also reset. Within each second, 50 samples are produced, and these are identified with the two time references. One time reference is provided by the internal clock of the SBC, which is used to timestamp the samples. The other one is provided by the oscillator that is used to provide a sequence number to each packet.

The drift of the oscillator within each second will be lower than $50 \mu\text{s}$, as its tolerance is 50 ppm. We measured the difference between these two time sources in order to evaluate the accuracy of one against the other. Figure 13 shows a histogram of the difference between the two time sources for a set of 38 662 samples. For most packets, the time difference between the two sources is less than $20 \mu\text{s}$, which means that both can be used with high confidence and that our samples are generated at the right times.

We also conducted a different test where three nodes were connected to a single signal generator. All nodes recorded the same signal, a square wave with 2.5 V amplitude and a 50 % duty cycle. For each sample, identified by the sequence number, we verified whether all nodes had recorded the same value: the top or bottom half of the square wave. Using 25 Hz, no two consecutive samples will have the same value, as the wave period will be 40 ms and our sampling interval is 20 ms, causing consecutive samples to oscillate between the top and bottom halves of the wave. Furthermore, as the clock of the signal generator is also not perfect, there are instants where the sample is performed during the transition from top to

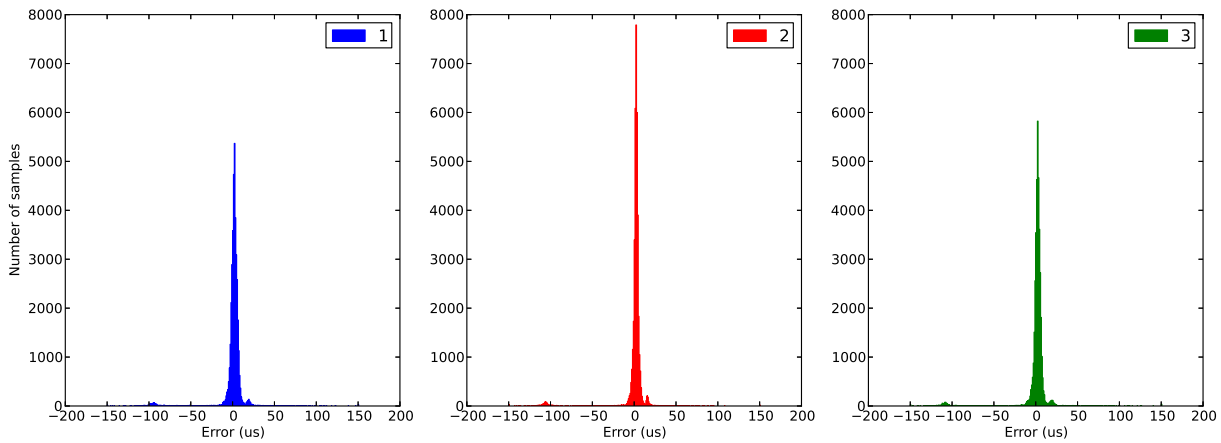


Figure 13. Histogram of differences between clock sources (1 us buckets).

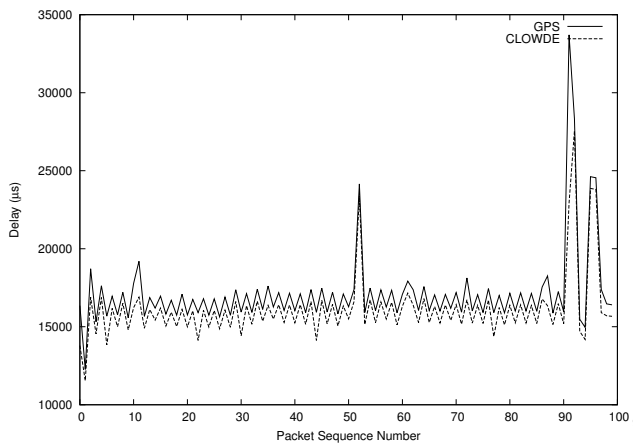


Figure 14. Sample timing delay.

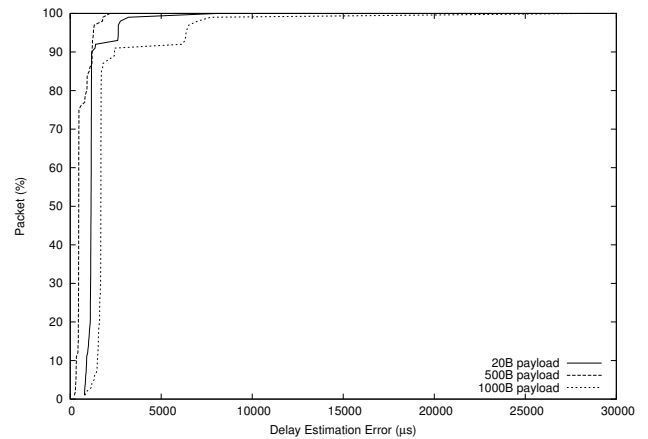


Figure 15. CLOWDE delay correction error.

bottom or vice versa. Even in this more challenging scenario, our three nodes measured the same position of the wave in all but 0.0654 % of the samples, as shown in Table 3. These values exceed the expected accuracy of our design, as the accuracy of 50 ppm of the oscillator could result in a maximum clock drift of $50 \mu s$. Given that at 25 Hz, each half wavelength is 10 ms, our designed maximum error is 0.5 %, 1 order of magnitude above what was actually measured.

5.2 CLOWDE

Among the several tests performed on the components and system as a whole, tests were conducted to evaluate the accuracy of the CLOWDE protocol. We evaluated CLOWDE using three nodes, with data crossing 2 hops from the source to the sink. This is consistent with our network topology presented in Fig. 1, where no packet crosses more than 2 hops.

Figure 14 presents the delay each packet traversing the network suffered. Measurements were taken at each node of the network by a GPS device and then compared with the delay

Table 3. Sampling time accuracy among three nodes.

Frequency	Number of samples	Non-matching samples
1 Hz	18 070	0.0276 %
5 Hz	38 144	0.0419 %
25 Hz	339 342	0.0654 %

estimated by the CLOWDE protocol. The packet delay is represented on the vertical axis, while the horizontal axis represents the received packet sequence numbers. We notice that the difference varies little, as both the CLOWDE and GPS results present similar behavior. We conclude that CLOWDE is able to capture the delay variations in message transmission that occur at each node. These delays are a consequence of CPU concurrent access by other processes running on sensor nodes and media access control (MAC) contention caused by the multiple nodes in the network.

Table 4. Network test–intermediate node failure test.

Metric	Value
Average data loss	2.19 s
Minimum data loss	0 s
Maximum data loss	8.61 s
Standard deviation	3.10 s
Confidence interval (95 %)	± 2.02 s

Figure 15 shows the CDF of the estimated delay error calculated by CLOWDE in the same 2-hop scenario using different payload lengths. The error presented on the horizontal axis is the difference between the estimated value and the time indicated by the GPS device. The vertical axis indicates the percentage of received packets. We notice that a large percentage of packets have similar, low delay errors. Also, the delay error is not significantly impacted by the variation in the payload length. It is also possible to notice that a small percentage of packets suffers from a considerably larger delay error than the majority of the remaining packets. In scenarios where the delay is bounded, such outliers could be easily filtered by the application, as their arrival time clearly differs from that of adjacent packets. In our particular use case, the outliers are filtered out by comparing the obtained creation time with that of the preceding and following packets, as they are created at regular intervals. Their true creation time may then be recalculated using interpolation, enabling us to achieve a precision of under 1 ms. A precision of ~ 50 ms was considered sufficient for re-synchronizing seismic signals (Budi-Santoso et al., 2013).

5.3 Network resilience

We have evaluated the ability of the network to handle node failures. Four nodes were placed in the topology depicted in Fig. 16. After the network was completely set up and all nodes were transmitting data to the sink node, we simulated the failure of node B by disconnecting its power supply. This was executed to simulate a complete node failure, without any type of warning, forcing the traffic between the sensor node and the sink node to use the alternative route through intermediate node A. These actions were executed 10 times, and for each repetition, we measured the number of data samples that did not reach the sink node. Results are shown in Table 4, and indicate that the system was able to recover with minimum data loss. The data loss was always below 10 s, with expected values for 95 % confidence values ranging from 0.17 to 4.22 s. This demonstrates that the volcanic monitoring system is able to adapt to topology changes within a short time, minimizing the data loss incurred by this rare event.

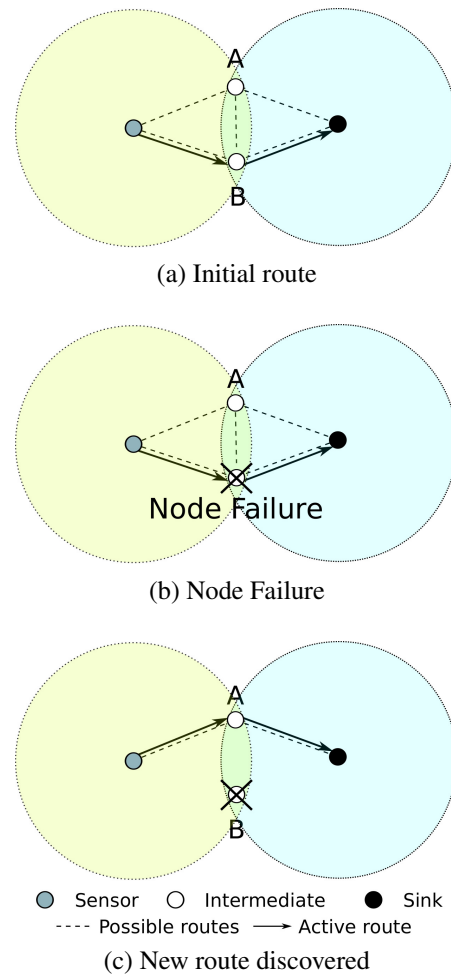


Figure 16. Intermediate node failure test.

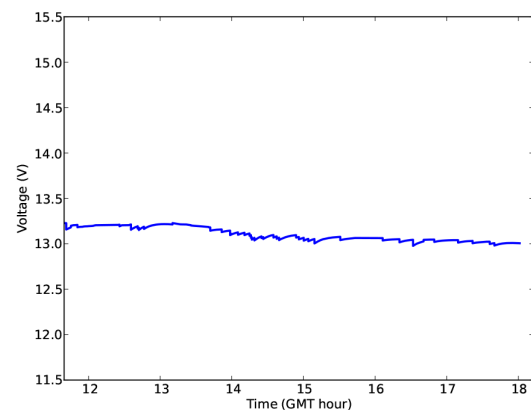


Figure 17. Evolution of internal battery charge using a 10 W solar panel.

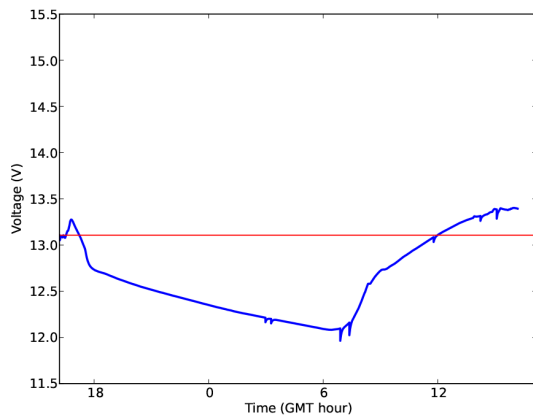


Figure 18. Evolution of external battery charge using a 30 W solar panel on a sunny day.

5.4 Power supply autonomy

We designed our system to be able to operate both in temporary or long-term/permanent deployments. For temporary, single-day deployments, during day time, we rely on the internal battery together with a 10 W peak power solar panel. Weighing only 400 g and having the format of a sheet of paper, the solar panels from NASA Marine are easy to carry. Using this setup, we are able to power a node throughout most of the day while the sun shines. Figure 17 shows the evolution of the battery charge during a deployment that lasted from a little after 11:00 until about 18:00 GMT during a sunny September day in Lisbon. In the beginning, we observe that the battery charge increases. Afterwards, it slowly decreases, as the solar panel is unable to continue to supply enough energy to power the node due to the change in the position and intensity of the sun. When we finished our test, the battery was still far from being fully discharged.

For long-term deployments, we rely on a 12 Ah battery, which is able to power a node for longer than 24 h. We validated two different setups in Lisbon during September: one during a mostly sunny day and the other during a sequence of very cloudy, rainy days. Figure 18 shows the evolution of the external battery charge during a 24 h cycle, using one 20 W and one 10 W solar panel. This day was mostly cloud free. The experiment started towards the end of the day, close to 18:00 GMT. The battery was not fully charged. It was still possible to charge the battery further using the last hours of sun. During the night, we observe a continuous discharge of the battery. As the sun comes up, the solar panels are capable of powering the node and recharging the battery. After 24 h, we can observe that the charge is greater than it was initially (horizontal line). Under these weather conditions and daytime duration, it would have been possible to power the node continuously.

We also experienced the system operating under stormy weather conditions for more than a week. Most days had

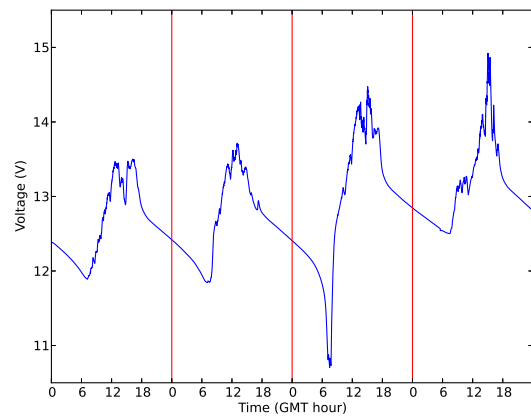


Figure 19. Evolution of external battery charge using a 50 W solar panel during stormy/cloudy weather.

strong rain with little (less than 10 % of the time) or no direct sunlight due to the permanent clouds over Lisbon. To power the node setup in this weather, we decided to increase the solar panel capacity to 50 W. This setup includes 2 kg of solar panels that due to their dimensions are still relatively easy to carry. Figure 19 shows the evolution of the battery charge over 4 days. The vertical lines at midnight signal the change of day. We can observe that the battery charge, which started at a low level, increases over time. Under these conditions, it would have been possible to operate continuously. We can also observe that a sunrise characterized by very dark clouds, on the third day, delayed the action of the solar panels.

The solar panel capacity will have to be adjusted according to the latitude and intended duration of each deployment. Over time, more efficient and affordable solar panels and batteries will become available.

5.5 Bridge vibration

In this test, we deployed three nodes on a 16 m long pedestrian bridge. The bridge has a steel structure and a wooden floor. It is anchored at the extremities and suspended by eight steel rods, four on each side along its length. We deployed two nodes (1 and 3) at the extremities and another one in the middle (node 2).

A pedestrian crossed the bridge 4 times: first, he walked across from the side where node 3 is to the other side and then back; later, he ran from the direction of node 3 to node 1 and then back. The four crossings can be observed in Fig. 20, which shows the signals recorded on the vertical axis of the geophone connected to each node.

The second crossing, framed by two vertical lines, is shown in greater detail in Fig. 21. We can observe the impact of each step and see the amplitude change as the pedestrian walks away from one node and towards another.

A spectrogram of the second crossing is presented in Fig. 22. We can observe that each step causes the bridge to

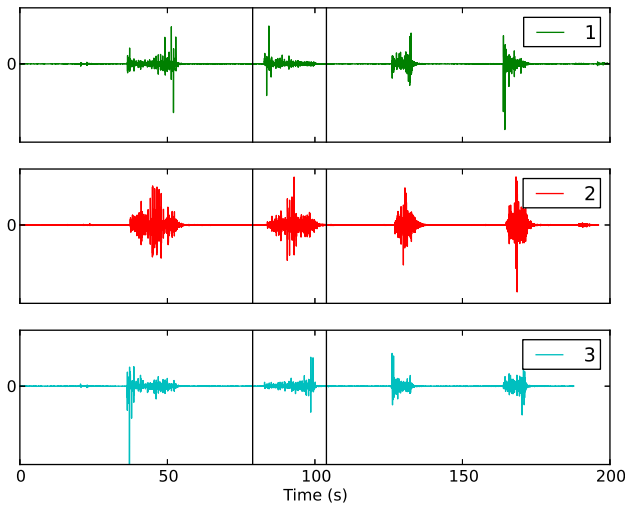


Figure 20. Signal intensity of four bridge crossings.

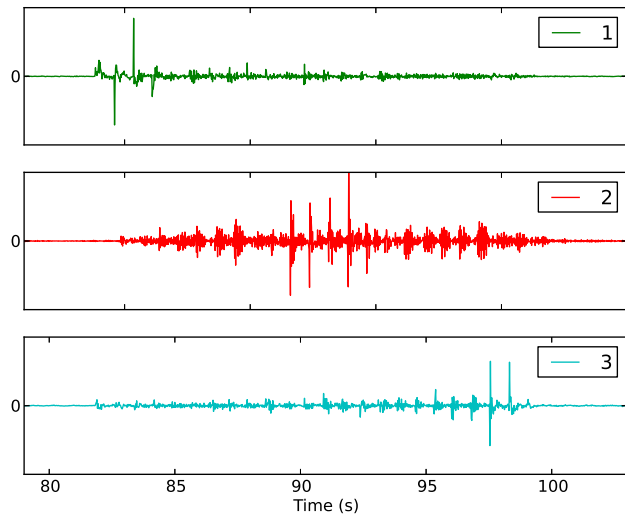


Figure 21. Detail of the second bridge crossing.

vibrate with frequencies across the measured spectrum and that a vibration centred around 20 Hz is present throughout the crossing.

5.6 Field deployment

We have executed a field test where nine nodes were deployed in a field, all within a 150 m radius from the sink. Nodes were deployed according to the topology defined in Fig. 1, although no intermediate nodes were used. The relative position of the nodes is presented in Fig. 23. Due to the presence of buildings, roads and other obstacles (which are not yet present in the picture, which is a few years old), node positions are not perfectly regular. Distances between adjacent nodes vary from 40 to 70 m. The line of sight was not available among most nodes. Each node was powered by its

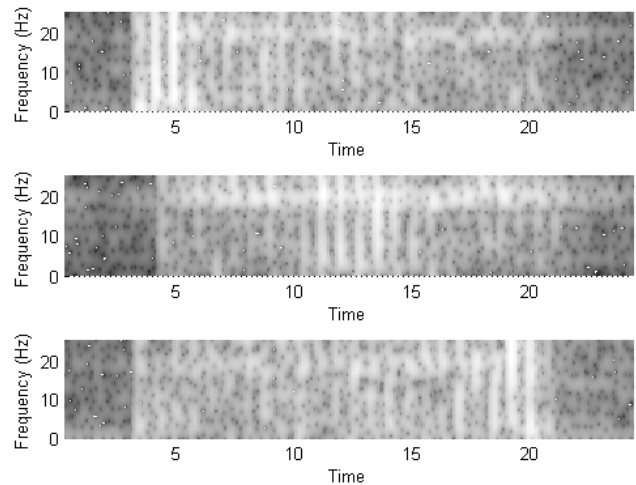


Figure 22. Spectrogram of second bridge crossing.

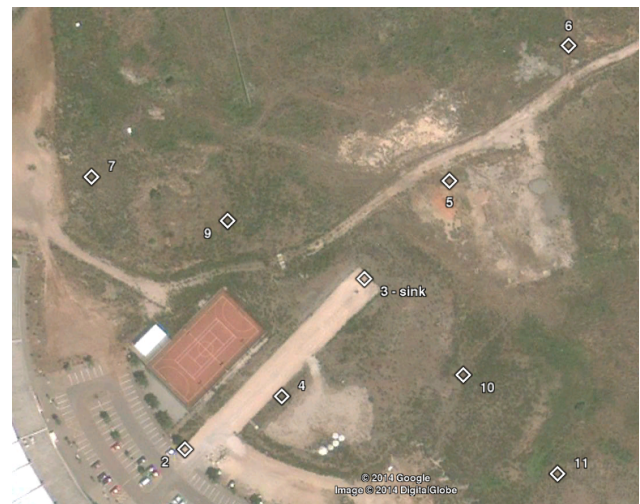


Figure 23. Location of the nine deployed nodes.

internal battery and a 10 W solar panel. This test was carried out during day time.

Transportation of the nodes was easy, as all equipment fitted into a small two-passenger car (a Smart Fortwo). The node cases were easily stacked on top of each other. The remaining equipment (solar panels, network cards and geophones), although not stackable, did not occupy a large space.

Installation of all nodes was executed in under 1 h. Deploying a node is a simple operation consisting of connecting the geophone, the solar panel and the WiFi card to the case and placing this equipment on the ground with the GPS antenna facing up. After a node is turned on, the system boots up, and tries to acquire a GPS lock and connect to the other nodes. The total deployment time for this procedure is less than 2 min. After the boot yellow LED is turned off and the GPS lock LED is on, the node is ready. At this point, connectivity

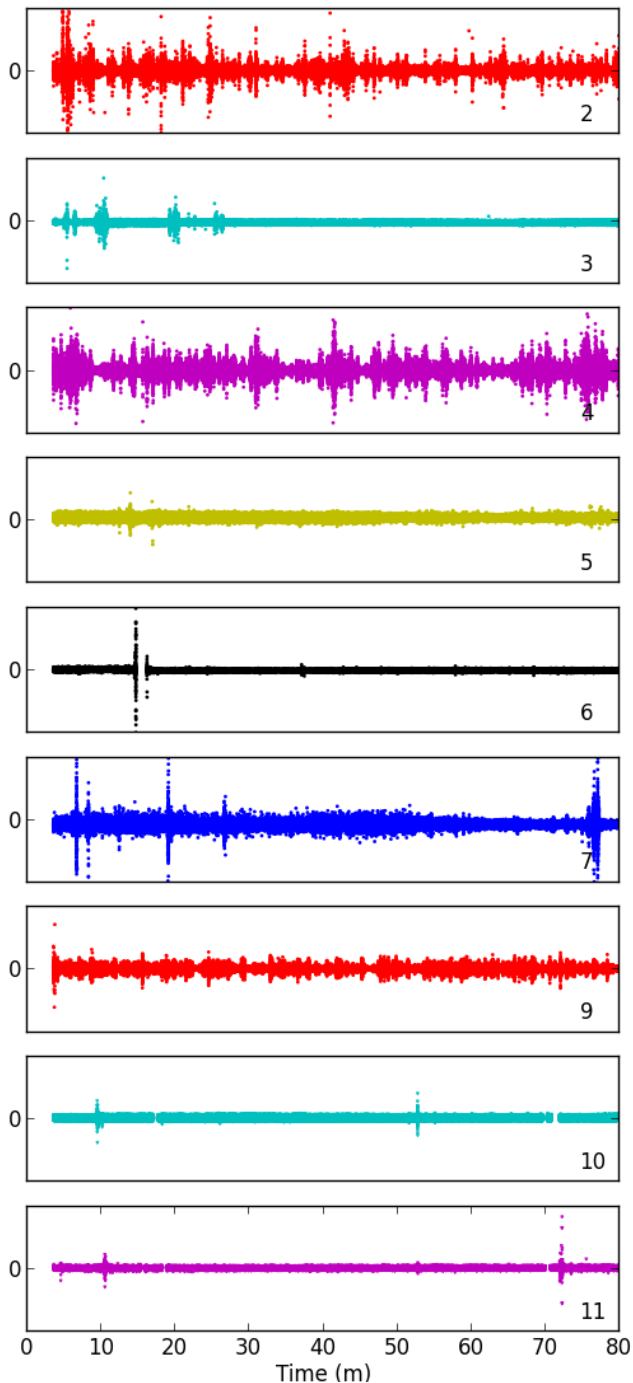


Figure 24. Recorded seismic signals.

was verified, the LED lights were turned off to save the battery, and the deployment proceeded to the next node.

In this field test, we noticed a design fault of the prototype, as the LEDs were difficult to read under direct sunlight. In future versions, this problem has to be taken into account by installing LEDs with a higher lumen output.

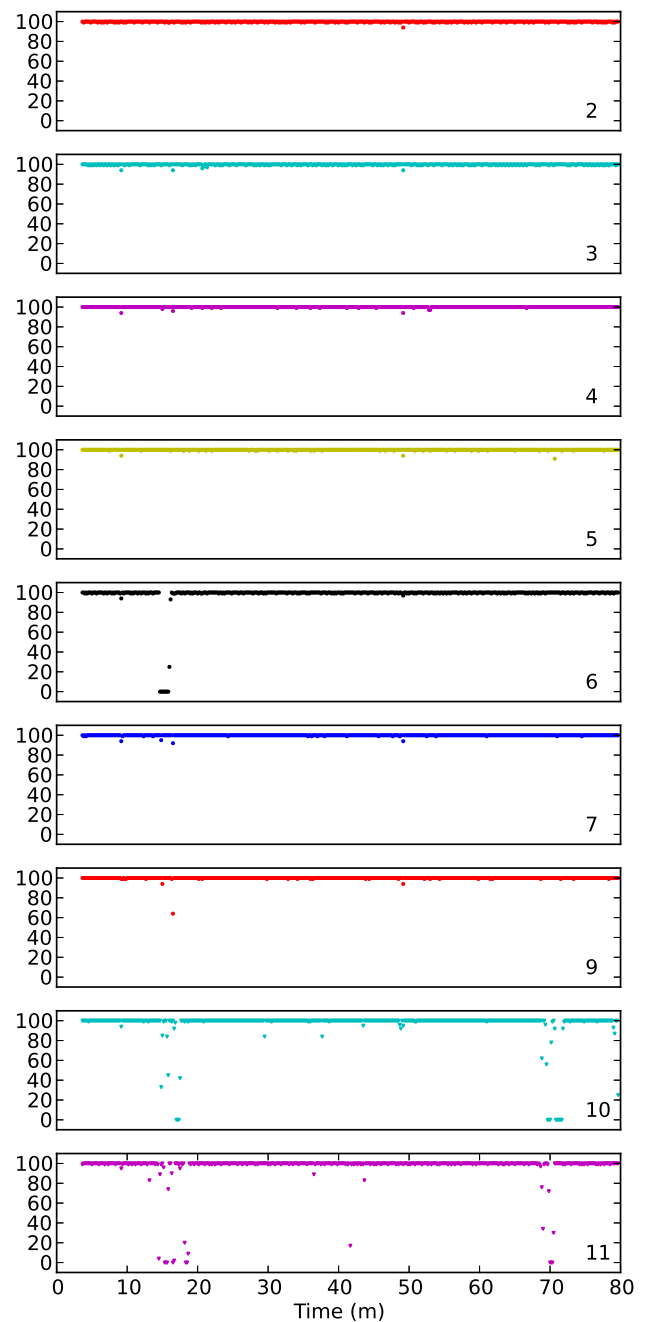


Figure 25. Percentage of received samples over 10 s intervals.

The test lasted a little over 3 h. It was conducted between 15:00 and 18:00 GMT. At the end of the test, all nodes were collected. This procedure required roughly half the time the deployment had taken, when it was necessary to wait for the GPS to acquire a lock. When performing this procedure, we detected another problem in the system design. There is no external button on the exterior of the case to shut down the equipment orderly. For this reason, as the various nodes were

being powered off, the rest of the network nodes considered the node to have failed.

Figure 24 presents the raw data recorded during this test. We only show data from the period after all the nodes were turned on until the first one was turned off. Even when referring to unprocessed data, it is possible to notice the recording of vibrations by the geological sensors. It is beyond the scope of this article to analyze the seismic signal.

Figure 25 presents an analysis of the samples produced by each node that were successfully recorded by the sink node. Each point represents the percentage of samples that were received for each 10 s interval (of an expected total of 500 samples). Nodes were configured to create one IP packet for each data sample (50 per second), in order to create the most stressful scenario. We can observe that most packets were delivered; however, there was an event that affected most nodes (6, 10 and 11 most significantly) before minute 20. Another event affected nodes 10 and 11 around minute 70. The area where the test was conducted was an adverse environment with many WiFi traffic sources and vehicle and pedestrian traffic. The distance between the nodes was short enough for several routing possibilities to exist at any given time. These events cause BATMAN to change the routing topology in order to adapt to the different connectivity possibilities. Overall, under these harsh conditions, most nodes delivered more than 99.9 % of their samples, with the most affected delivering more than 96 % of the packets. Table 5 shows the percentage of samples received from each node.

6 Conclusions

This article has presented a solution for a WSN for volcanic tremor monitoring using mainstream COTS components. The proposed design provides a flexible, easy/quick-to-deploy WSN that can be used for temporary or permanent monitoring in remote locations. To guarantee a low-cost and easy-to-maintain solution, we used open-source software and, whenever possible, standard protocols.

We provide users with two options for synchronizing data collection times. When a GPS device is present, highly accurate timestamps are created for each sample at the collector nodes through a data acquisition component. For scenarios where the collector nodes are located in places without GPS signal reception, we propose the CLOWDE algorithm as an alternative timestamping technique, albeit with lower accuracy.

The BATMAN routing protocol is used to generate routes dynamically for forwarding packets over the ad hoc network to the sink node. This protocol does not require nodes to trade information about every network change. Instead, a simple message cycle is generated where nodes inform neighbors about their network location. Each node uses these messages to store information about the best route to every other node. BATMAN provides a good routing protocol, while letting us

Table 5. Samples delivered.

Node	Samples delivered (%)
2	99.99 %
3	99.95 %
4	99.95 %
5	99.95 %
6	98.05 %
7	99.95 %
9	99.88 %
10	96.14 %
11	96.43 %
All	98.92 %

use Netfilter hooks for the CLOWDE time synchronization protocol implementation. Another advantage of BATMAN is the ease it provides in adding nodes to the network. Also, if a node fails, it can be automatically replaced by another, as BATMAN will start a route repair process.

In general, the presented solution met the pre-established system requirements, being low-cost WSN by using COTS components. Also, it is very flexible as, although it was designed with the goal of being used by geophysicists, it can be easily adapted to other purposes as new use cases develop.

Future work

Future work concern is related to the overall network availability. Nodes in the network can fail due to different reasons (e.g., natural hazards or vandalism); however, one of the desired properties of the proposed architecture is that the network should remain functional. This property is respected if the failing node is a sensing node or an intermediate node but not the sink node, as data are sent to a specific sink node address. In the future, we will investigate the use of redundancy techniques that enable another node to take the place of the sink, should this fail. This functionality will also require the redundant node to be equipped with a persistent storage device and communication with the outside, if used.

Recently, new SBC with lower power consumption have become available. The TS-7500 consumes 2 W. Other slightly less powerful SBC, such as the TS-7260, are now available that consume as little as 0.25 W. The use of such SBC would drop the total power envelope from 4.25 to 2.5 W, enabling a considerable reduction in battery and solar panel requirements. The adoption of a new SBC will entail significant changes in both the hardware and software produced.

Acknowledgements. This work was partially supported by national funds through the FCT (Fundação para a Ciência e a Tecnologia) under project PEst-OE/EEI/LA0021/2013. The MIA-VITA project is financed by the European Commission under the 7th Framework Programme for Research and Technological Development, Area “Environment”, and Activity 6.1 “Climate Change, Pollution and Risks”. We would like to thank João Gomes, Simão Silva, Artur Balanuta and Vasco Rato for their participation in this project.

Edited by: P. Jousset

Reviewed by: M. Picozzi and another anonymous referee

References

- Abolhasan, M., Hagelstein, B., and Wang, J. C.-P.: Real-world performance of current proactive multi-hop mesh protocols, in: 15th Asia-Pacific Conference on Communications (APCC 2009), Shanghai, 44–47, 2009.
- Akyildiz, I. and Wang, X.: A survey on wireless mesh networks, *IEEE Commun. Mag.*, 43, 23–30, 2005.
- Al Basset Almamou, A., Wrede, R., Kumar, P., Labiod, H., and Schiller, J.: Performance evaluation of routing protocols in a Real-World WSN, in: Global Information Infrastructure Symposium, 2009, GIIS’09, Hammamet, 1–5, 2009.
- Budi-Santoso, A., Lesage, P., Dwiyono, S., Sumarti, S., Subandriyo, S., Jousset, P., and Metaxian, J.-P.: Analysis of the seismic activity associated with the 2010 eruption of Merapi Volcano, Java, *J. Volcanol. Geoth. Res.*, 261, 153–170, 2013.
- Chouet, B.: A Seismic Model for the Source of Long-Period Events and Harmonic Tremor, in: *Volcanic Seismology*, IAVCEI Proceedings in Volcanology, Springer, Berlin, Heidelberg, 133–156, 1992.
- Chouet, B.: New Methods and Future Trends in Seismological Volcano Monitoring, in: *Monitoring and Mitigation of Volcano Hazards*, Springer, Berlin, Heidelberg, 23–97, 1996.
- Claerbout, J. F.: Introduction to Seplib and SEP utility software: Stanford Exploration Project 70, Stanford Exploration Project, <http://sep.stanford.edu/public/docs/sep70/jon4.pdf> (last access: November 2014), 1997.
- Duriscic, M., Tafa, Z., Dimic, G., and Milutinovic, V.: A survey of military applications of wireless sensor networks, in: *Mediterranean Conference on Embedded Computing (MECO)*, Bar, 196–199, 2012.
- Faria, B.: Monitorização geofísica e níveis de alerta do vulcão do Fogo (Geophysical monitoring and alert levels for Fogo volcano), Ph.D. thesis, Instituto Superior Técnico, Lisboa, 2010.
- Fischer, J., Redlich, J.-P., Zschau, J., Milkereit, C., Picozzi, M., Fleming, K., Brumbull, M., Lichtblau, B., and Eveslage, I.: A Wireless Mesh Sensing Network for Early Warning, *J. Netw. Comput. Appl.*, 35, 538–547, 2012.
- Geoffrey, C. and Welsh, M.: Deployments and Design Frameworks, in: chap. *Volcano Monitoring: Addressing Data Quality Through Iterative Deployment*, Springer Science, 2010.
- Girod, L., Lukac, M., Trifa, V., and Estrin, D.: The design and implementation of a self-calibrating distributed acoustic sensing platform, in: *Proceedings of the 4th international conference on Embedded networked sensor systems*, SenSys’06, ACM, New York, NY, USA, 71–84, 2006.
- Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K.: System architecture directions for networked sensors, *SIGPLAN Not.*, 35, 93–104, 2000.
- Johnson, D., Ntlatlapa, N., and Aichele, C.: Simple pragmatic approach to mesh routing using BATMAN, in: *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries*, Pretoria, South Africa, p. 10, 2008.
- Johnstone, I., Nicholson, J., Shehzad, B., and Slipp, J.: Experiences from a wireless sensor network deployment in a petroleum environment, in: *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, IWCMC’07, ACM, New York, NY, USA, 382–387, 2007.
- Jousset, P., Budi-Santoso, A., Jolly, A. D., Boichu, M., Surono, Dwiyono, S., Sumarti, S., Hidayati, S., and Thierry, P.: Signs of magma ascent in LP and VLP seismic events and link to degassing: An example from the 2010 explosive eruption at Merapi volcano, Indonesia, *J. Volcanol. Geoth. Res.*, 261, 171–192, 2013.
- Kurin, E.: Computers for seismic processing and imaging: a performance study, in: *SEG Annual Meeting*, San Antonio, Texas, 2451–2454, 2007.
- McGuire, W., Solana, M., Kilburn, C., and Sanderson, D.: Improving communication during volcanic crises on small, vulnerable islands, *J. Volcanol. Geoth. Res.*, 183, 63–75, 2009.
- McNutt, S. R.: Volcanic seismology, *Annu. Rev. Earth Planet. Sci.*, 33, 461–491, 2005.
- Mogaibel, H. and Othman, M.: Review of Routing Protocols and It’s Metrics for Wireless Mesh Networks, in: *International Association of Computer Science and Information Technology – Spring Conference*, IACSITSC’09, Singapore, 62–70, 2009.
- Murillo, A. E. and Bell, J.: Distributed Seismic Unix: a tool for seismic data processing, *Concurrency*, 11, 169–187, 1999.
- Rodriguez, I. V. and Sacchi, M. D.: Development of a Guided User Interface for Microseismicity Analysis, in: *Recovery CSPG CSEG CWLS Convention*, Calgary, Canada, 2011.
- Scognamiglio, L., Tinti, E., and Michelini, A.: Real-Time Determination of Seismic Moment Tensor for the Italian Region, *Bull. Seismol. Soc. Am.*, 99, 2223–2242, 2009.
- Taisne, B., Brenguier, F., Shapiro, N. M., and Ferrazzini, V.: Imaging the dynamics of magma propagation using radiated seismic intensity, *Geophys. Res. Lett.*, 38, L04304, doi:10.1029/2010GL046068, 2011.
- Wang, X., Zhao, X., Liang, Z., and Tan, M.: Deploying a Wireless Sensor Network on the Coal Mines, in: *IEEE International Conference on Networking, Sensing and Control*, London, 324–328, 2007.
- Werner-Allen, G., Lorincz, K., Johnson, J., Lees, J., and Welsh, M.: Fidelity and yield in a volcano monitoring sensor network, in: *Proceedings of the 7th symposium on Operating systems design and implementation*, OSDI’06, USENIX Association, Berkeley, CA, USA, 381–396, 2006a.
- Werner-Allen, G., Lorincz, K., Ruiz, M., Marcillo, O., Johnson, J., Lees, J., and Welsh, M.: Deploying a wireless sensor network on an active volcano, *IEEE Internet Comput.*, 10, 18–25, 2006b.
- Yick, J., Mukherjee, B., and Ghosal, D.: Wireless sensor network survey, *Computer Networks*, 52, 2292–2330, 2008.