Nat. Hazards Earth Syst. Sci. Discuss., 1, 4305–4348, 2013
www.nat-hazards-earth-syst-sci-discuss.net/1/4305/2013/
doi:10.5194/nhessd-1-4305-2013

Natural Hazards and Earth System Sciences
Discussions
Open Access

A wireless sensor network for monitoring volcanic tremors

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]

[1]INESC-ID, Av. Prof. Dr. Cavaco Silva, 2744-016 Porto Salvo, Portugal
[2]Instituto Superior Técnico, Universidade de Lisboa, Avenida Rovisco Pais, 1 – 1049-001 Lisboa, Portugal

Received: 6 June 2013 – Accepted: 29 July 2013 – Published: 30 August 2013

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

Published by Copernicus Publications on behalf of the European Geosciences Union.

Abstract

Monitoring of volcanic activity is important to learn about the properties of each volcano and provide 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 maintain, adaptable, low-cost WSN for temporary or permanent monitoring of seismic tremor. The WSN enables the easy installation of a sensor array on an area of tens of thousand of m^2, allowing the location of the magma movements causing the seismic tremor to be calculated. This WSN can be used by recording data locally for latter analysis or by continuously transmitting it in real time to a remote laboratory for real-time analyses.

1 Introduction

Volcanologists often use wired arrays of sensors, usually seismometers, to monitor volcanic eruptions and tremor: a very low frequency seismic signal that precedes a volcanic eruption, caused by the movements of the magma in the interior of the crater. 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 calculated. Most of the equipment used in these systems had the particularity of being extremely heavy, normally the size of a small cabinet, and expensive to purchase or maintain. These properties limit the number of devices which can be feasibly installed on a remote location. Also, from a technical perspective, these devices usually rely on specific non-standardised communication protocols, which constrain the system's maintenance, evolution and integration.

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. With traditional sensor arrays, deployment of complex experimental setups, with a significant number of nodes dispersed over a wide region, is not an easy endeavour as these are difficult to install and maintain and due to the high price tag associated with these devices. Today's typical setup usually comprise just a few sensing devices (usually less than five), distributed over a small sized area (less than 100 m^2), being the data stored locally using hard drives or flash devices. Given these limitations, the physical loss or destruction of a single device, although quite probable in case of an eruption, becomes very critical.

In the past few years there has been an increased research effort in the area of Wireless Sensor Network (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 patients health monitoring (Akyildiz and Wang, 2005; Durisic et al., 2012). One of the possible application field for WSN is volcanic monitoring.

In 2004, an USA research project deployed a small test WSN in the Volcan Tungurahua in central Ecuador (Werner-Allen et al., 2006b). During three days, data from the active volcano was 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 out to design a flexible, easy to deploy and 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 which can easily be procured and

replaced in the field. Our system is based on commercially available equipment, such as 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.

The remainder of this document is organized as follows: Sect. 2 describes the context in which this study was performed and related state of the art. Section 3 presents the requirements that were defined for the seismic waves monitoring system and describe the hardware and software architectures and components of our solution. 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 behaviour 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 of the fluid's pressure. The recorded ground motion is often of the surface-wave type, but it can also be formed of body waves if the source is deep (Faria, 2010). To analyse these events, seismic activity caused by the magma movements inside the magmatic chamber may be measured. If detected in an early stage, this can give an early warning of an eruption, enabling the proper actions 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 intensity in the surroundings of the crater. To have a 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 must be easily and quickly repeated in a different region in the vicinity of the volcano. The sensors must cover a region where the entire 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 in each one of the cardinal points are required.

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

2.2 Related work

Research in 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 Volcán Tungurahua in central Ecuador as a proof of concept on how these type of networks could efficiently replace traditional monitoring equipments (Werner-Allen et al., 2006a, b). Nodes used an event detection algorithm which, 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 scenario as 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 for each node only allowed to record 20 min worth of data and only transmit one event at a time. In one of the deployments at the volcano, these properties resulted on the lost of data recordings of a giant explosion. This was due to a smaller, non interesting eruption which preceded the larger eruption and occupied the network while the larger eruption occurred.

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 on a coal mine to reduce coal mine work related deaths (Wang et al., 2007). Their proposed system detected current levels of methane, temperature, humidity, 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[1]. This option of only transmitting interesting events allowed the reduction of network resources used and saves node battery. A combination of a star and peer-to-peer network topologies was researched, checking which was more adequate for coal mines use case scenarios.

For monitoring environmental condition in petroleum extraction facilities and oil rigs, researchers from Dalhousie and Cape Breton University have implemented and deployed a WSN where a heterogeneous architecture is 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 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 synchronised 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

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

that timestamp as the "local time" on every hop through the network or use a global time service, on a node with Global Positioning System (GPS) and broadcast this time to the rest of the network hop by hop. The first solution provides relative times to the *timestamp of interest*, while the second option make all nodes synchronized to a global clock. In both cases, an overhead for synchronisation messages is imposed on the network.

3 Architecture

This section presents the architecture of the WSN created for monitoring seismic tremor, being divided into four parts. First we will detail the requirements we set off to fulfil. 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 which comprise it. Finally, we will introduce the monitoring application which allows users to visualise the collected recordings and monitor the network state, either remotely or on 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, allow for temporary or permanent deployments, use a common clock reference for timestamping samples and provide 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. It should also be possible to record the collected data locally and/or transmit it to a remote laboratory. There

should be *flexibility* about the network technologies which can be used for transmitting the data.

Although the WSN is 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.

A WSN is made up of a set of nodes networked together. *Ease of deployment* should be targeted, as personnel installing equipment on the field should not be required to have expertise on embedded systems or wireless networks. After deployment, nodes and the network should automatically self configure. Field personnel should have access, on the field, to tools which 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 communication with each other. *Ease of maintenance* is required, namely the existence of mechanisms which 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 which has *low maintenance*, specially 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 available to invest in adequate protection. For a permanent deployment, nodes may be sheltered and more powerful and reliable power sources can be made available. These requirements have to be balanced in order to provide a single, all round solution.

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

Recorded *data should be easily to access*. Either on the field or on a remote laboratory, recorded data should be easy to access, both in real-time or 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

Globally the proposed system is graphically represented in Fig. 1.

Seismic signals are collected at the remote volcanic location by a sensor array. A single special node in the sensor array, the sync node which will be detailed later, is then responsible for gathering all the data and transmit to remote location, e.g. using a satellite gateway. It is from this point that collected samples are sent to the remote volcanological laboratory, which is often located abroad. There, specialized personnel is able to analyse 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 on Fig. 2.

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

1. The sink node is located in the centre of the topology in order to reduce the maximum number of hops which 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 nodes transmit

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

2. Nodes which perform the intermediate role are normal sensor nodes but their specific location is chosen in a manner 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 as being an intermediate, it's wireless communication radius has to encompass at least two other nodes.

3. Finally, nodes can have the role of sensor nodes. These devices only acquire data from their sensing devices and transmit the collect information to the sink node or to another node which is in the path to the sink node. Sensor nodes will relay data towards the sync for other nodes which are unable to reach the sync 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 Node architecture

Each node is composed by the components shown on Fig. 3. Globally there are two types of components, hardware and software. Software components can 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 which run on user space, the monitoring application and the data manager. All remaining software components (controller, time synchronisation, clowde, communication manager and sample acquisition) are executed on 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.

In terms of hardware, each node includes an Universal Serial Bus (USB) WiFi card and an Ethernet port. The Analogto- Digital Converter (ADC) maps the analogue signal received from the multiple sensors into digital values, while the GPS provides the node with the current time and location. The local storage is a disk where data is stored.

3.3.1 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 of the node's role on the network, either sends data samples to the communication manager or to the data manager for persistent storage in the case of the sync 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 reduce the power consumption, each node aggregates groups of data samples. Samples are aggregated by destination address, which for all samples is the sync node. This ensures that aggregated packets are all destined to the same node, which will ease the computation procedure for 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. In order to support synchronisation, each time a PDU or an IP packet is stored in the aggregation buffer, the protocol will compute the difference between the packet being aggregated and the one at the head of the buffer. For this reason, packets are aggregated in such way that the last packet in the buffer is the one with the smallest sequence number.

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 then in the node's network stack.

3.3.2 Data manager

The data manager module is responsible for storing and retrieving samples from a non-volatile medium (in our case an USB pendrive or disk). It can only receive as input samples transmitted by the controller module. Although this happens in all nodes roles, it is more important for the sink node where samples gathered from all the 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 which can be easily read by humans or computers. An open-source exporter capable of transforming both these formats into miniseed has been developed[2] in order to allow interoperability with other systems.

3.3.3 Sample acquisition

Seismic tremors is to be sensed using geophones which produce an analogue signal. A sample acquisition system is required in order to convert the analogue signals into digital data. Samples must be collected with a constant rate, which has to be more accurate that what can be achieved by a software controlled solution. Thus a hardware solution had to be designed using a high precision oscillator driving an ADC. Whenever a sample is available, an interruption is raised, triggering the sample acquisition module to collect the data.

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

3.3.4 Time synchronisation

In order for different WSNs to share a common clock, each sink node was equipped with a GPS device.

Samples take different lengths of time to arrive at the sync 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. One of our solutions is to equip each sensor node with a GPS device, providing highly accurate time synchronisation within the WSN. When the GPS device is present, it is utilized for two purposes: to provide controller module with a time reference to timestamp samples and for synchronising the ADC itself.

3.3.5 CLOWDE

Given MIAVITA use case requirements, it was decided to develop a delay estimating algorithm capable of approximately calculating the time it takes for data to go from the application that created it, through the various hops on network, up to the application present on the sink node. This algorithm, named Cross-Layer One-Way Delay Estimation (CLOWDE) precludes GPS devices from being necessary on nodes other than the sink, lowering the WSN cost. 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 is created until it is received by the next node; and finally, the sink node uses its reference time 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, the application creates a message at the source node (T_{creation}) containing the data to be transmitted to the sink. After, the application sends the message, which must travel down the IP stack until the Link layer pushes it to the network card. This time

is identified as T_{source}. The network card will then transmit the packet to the next hop. This time, T_{hop_0}, comprises the propagation and transmission delay. Finally the packet reaches the sink node, where it will travel up the IP stack until it is delivered to the destination application, 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, where the forwarding decision is performed ($T_{\text{in}}U_{\text{p}_n}$); second, it is pushed down to the Link layer, taking the value of $T_{\text{in}}D_{\text{n}_n}$; and third it is sent over the air to the next hop (T_{hop_n}).

Considering a path with $N+1$ nodes, T_{creation} is given by:

$$T_{\text{creation}} = T_{\text{reference}} - \sum_{n=0}^{N+1} \left(T_{\text{hop}_n} + T_{\text{in}}U_{\text{p}_n} + T_{\text{in}}D_{\text{n}_n} \right) \tag{1}$$

where:

$$T_{\text{in}}U_{\text{p}_0} = 0, \quad T_{\text{in}}D_{\text{n}_0} = T_{\text{source}}$$
$$T_{\text{in}}U_{\text{p}_{N+1}} = T_{\text{sink}}, T_{\text{in}}D_{\text{n}_{N+1}} = 0$$

The time packets spend inside each node can be measured using that node's local clock. This way, T_{source}, T_{sink}, $T_{\text{in}}U_{\text{p}_n}$ and $T_{\text{in}}D_{\text{n}_n}$ 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 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 synchronised.

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 RTS/CTS. We believe this to be an interesting case, due to the wide range of situations it covers. Thus, in the next section we will describe how time is estimated in 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 ACK 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.

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

where DIFS is the DIFS; T_{tx} is the transmission delay; T_{prop} is the propagation time.

DIFS has a constant value of 50 µs in IEEE 802.11b. The propagation delay is simply the distance between nodes divided by the speed of light ($3 \times 10^8\,\text{m}\,\text{s}^{-1}$). 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 ($1\,\text{Mb}\,\text{s}^{-1}$) speeds.

Although the air time of a first try successful transmission can be easily determined, the same does not hold true for situation with initially busy channel, 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 needed not 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. 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.

3.3.6 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 best routes according to the wireless signal strength. Several surveys regarding this subject have been presented (Mogaibel and Othman, 2009; Akyildiz and Wang, 2005; Al Basset Almamou et al., 2009).

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 main focus is the decentralization of knowledge regarding best routes through the network which results in no single node having all the data. Because of the use of this technique, there is no need for spreading information concerning network changes to every node in the network. BATMAN acts as a distance-vector protocol and does not try to determine the complete route. It relies on the originator-messages to forward a packet's first hop in the right direction. The packet is handed over to the next neighbour in that direction, who uses the same mechanism. This process is repeated until the data reaches its destination. To spread topology information, periodically every node sends out a broadcast with the objective of informing all its neighbours about its existence. The neighbours then relay this message to their immediate neighbours and a cycle is created. It is this simple operation which 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 neighbour the message was received. In

[3]http://www.open-mesh.org

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 (SO) 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 synchronisation 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 GPS device, there is support for replacing the sink automatically, if this node happens to become damaged.

3.4 Monitoring application

The analysis and processing of seismic data requires 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 several hours until data can be visualized. These properties are not an obstacle for specialized analysis performed on historical data where no real-time requirements in visualizing collected data exist. In MIAVITA use case requirements we listed that the monitoring system should enable personal on the terrain to be able to quickly and effectively observe the status of the various nodes collecting data and of the network.

With this requirement in mind, the proposed monitoring application is composed by 4 components as depicted in Fig. 6. The Web Server is the system which delivers the web pages requested by the clients using Hypertext Transfer Protocol (HTTP). The chosen webserver application was lighttpd, due to its optimization for speed critical environments with small memory resources, for being standards-compliant and

historically having few security problems. The monitoring application is responsible for creating the webpages through which users can interact with the system. It consists of a set of javascript functions which, upon client request, are transferred to clients browsers and then locally executed. 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 webbrowser (e.g. Internet Explorer 8, Firefox 3, Google Chrome 16 or more recent) which is able to execute javascript commands. As the generation of the graphical interface was based on standards, it functions in desktops or mobile system, 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 of the most recent data acquired by the multiple sensor in the network. This functionality is shown on Fig. 7a. Figure 7b indicates the location of various sensor on the network over a geographical map of the region. Finally Fig. 7c provides personnel on the terrain 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.

To facilitate the integration of the produced data with existing system, the exporter components transforms both the JSON and binary format into the standardized, widely used Miniseed standard format.

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 Systems[4]. This device is small, measuring 66.6 × 74.3 mm and very light (less then 50 g). It is equipped with a 250 MHz ARM version 4 central processing unit and has 64 MB of RAM available for the SO. Regarding inputs and outputs it has Serial Peripheral Interface (SPI) and an USB buses which are used to connect external hardware. This type of equipment was chosen mainly due to its low power consumption (400 mA and 5 V) and its option to run a vanilla Linux kernel, increasing the standardization of the proposed solution. The complete SO and our software are executed from a micro SD card which allows for easy substitution if an update is required.

4.1.1 Sample acquisition board

The Guralp HS-1 3C Array 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 which provides a reference clock and allows nodes to be located; it provides a high-power USB port; allows four Ligth-Emitting Diodes (LEDs) to be used to convey information to the users; and 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,

[4]http://www.embeddedarm.com/products/board-detail.php?product=TS-7500

producing samples at a rate of 7200 Hz. In order to accommodate the GPS time synchronisation, 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 Synchronisation kernel module.

The GPS device must only be present in the sync 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, highly accurate timestamps are created for each sample at the collector node. Lassen iQ GPS receivers were used. These devices are controlled using a serial port, which provides time with accuracy within a tenth of a millisecond. As we required greater precision, we also used a Pulse per Second (PPS) line, serving a dual purpose: reset the ADC and raise an interruption on the SBC. The PPS signal provides accuracy of 50 ns. The ADC is thus reset every 1 s ensuring that all nodes are sampling 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 interrupt and syscal measurements performed, we estimate the timestamps to be accurate within 100 us. 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 it 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 an 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 Power supply

To power each node we decided to use 12 V lead-acid batteries. This kind of batteries are available worldwide as they are the standard batteries used on automobiles. This is extremely important as the devices are to be installed in foreign countries, where purchasing specialized equipment can be extremely expensive and involve a bureaucratic process. The batteries used on our scenario have 100 A h and weight around 18 kg. One single battery is able to power up a node during 3 continuous days. If an extended autonomy is desired there is the option of installing a solar panel.

4.3 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 to communicate over distances of 1000 m without any packet losses if 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 which correctly supports ad hoc mode operation. This behaviour is due to limitations of the open-source driver used.

4.4 Case

The several components present in each node are installed on a rectangular cuboid case made of aluminium measuring 22 cm × 14 cm × 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 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:

- Geophone proprietary plug to connect the device with external sensors.
- Male USB adapter to connect with external WiFi antenna.
- Power adaptor to connect to battery or solar panel.
- Power on/off switch for the whole device.

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

- 1 Button for enabling/disabling LEDs to save energy if not in use.
- 2 Green LEDs for indicating Power On and if the GPS device has a satellite lock.
- 2 Yellow LEDs, one indicating an internal fault and the other warning that the system is booting.

An image of the interior of the device's case is shown in Fig. 11. In the image the following components are highlighted: (A) button for enabling/disabling LEDs, (B) LEDs, (C) backup/temporary deployment battery, can be replaced by external hard drive for sample storage if user desires, (D) USB connection to external antenna, (E) Ethernet port, (F) TS-7500 embedded arm device, (G) GPS antenna, (H) data acquisition module, (I) power connecter, (J) geophone connector, and (K) power switch.

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

5 Evaluation

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

The communication manager component, with the use of the BATMAN protocol, allows for a flexible, dynamic, number of nodes to participate on the WSN. Connection to external entities, like a remote volcanological laboratory is facilitated by the *flexibility* provided by the use of the Debian GNU/Linux operation system: 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 sample acquisition board is *adaptable* as it allows up to eight different analog sensors to be connected. 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 SO allows software to be easily developed, without requiring extensive WSN knowledge. The SBC used is power enough to run reasonably complex software, allowing some processing, such as event detection, to be run 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 which 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 sync (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 sample 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, costed about 350 € each to produce. This value is expected to drop significantly for larger production runs. This cost does not include battery, solar panel nor sensor devices, such as geophones. The use of CLOWDE provides reasonable clock accuracy, allowing sensing nodes to do without GPS devices, thus lowering their cost (to 290 € in our prototype).

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 which it will be paired with, while being sufficiently spacious for installing hard disks for long duration recordings of data where communications to 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 easily performed.

The use of a GPS receiver at the sync node ensures *time synchronisation* among different WSNs and with other data sources. GPS devices can also be installed in every node when high accuracy among the nodes is required. Otherwise, the CLOWDE protocol enables node cost to be lower and enables operation in location where GPS reception is poor or impossible, such as dense forests, underground or indoor.

Recorded *data is easy to access*. As data is 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 sync node builds an ad hoc network for management, enabling any laptop or tablet to connect, being auto-configured. This, together with the HTTP server and monitoring application, allow data to be easily and wirelessly visualised or downloaded in the field.

5.1 Experimental evaluation

Among the several tests performed on the components and system as a whole, tests were conducted to evaluate the accuracy of the CLOWDE protocol and the operation of the sample acquisition board.

Figure 13 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 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 a similar behaviour. We conclude that CLOWDE is able to accurately capture the delay variations in message transmission which occur at each node. These delays are a consequence of CPU concurrent accesses by other processes running on sensor nodes and Media Access Control (MAC) contention caused by the multiple nodes on the network.

Figure 14 shows the Cumulative Distribution Functionu of the estimated delay error calculated by CLOWDE in the same two hop scenario. The error presented in 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 of 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 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 them be recalculated using interpolation.

Figure 15 presents the raw seismic signals collected for the sink node for a time period of 700 s. Even referring to unprocessed data, it is possible to notice the recording of vibrations by the geological sensors.

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 which 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 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, highly accurate timestamps are created for each sample at the collector node through a sample acquisition component.

As for forwarding packet in the ad hoc network, the routing protocol BATMAN provides a good solution to dynamically generate routes 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 neighbours about their network location. Each node uses these messages to store information about the best route to each other node. BATMAN provides a good routing protocol, while letting us use Netfilter hooks for the CLOWDE time synchronisation protocol implementation. Another advantage of BATMAN is the ease it provides to add nodes to the network. Also, if a node fails it can be automatically replaced by another as BATMAN will proceed to a route reparation process.

In general the presented solution met the pre-established system requirements, being a low-cost WSN 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.

6.1 Future work

A future work concern is related with the overall network's 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 if 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 is sent to a specific sink node address. In the future we will investigate the use of redundancy techniques which enable another node to take the place of the sink, should this fail. This functionality will require the redundant node to also be equipped with a GPS device, persistent storage device and communication to the outside, if used.

A different functionality which can be studied in future work is the inclusion on the system of a component which enables automatic event detection and early warning transmission. Such system would allow the network to avoid the continuously data transmission to a remote volcanological laboratory, only triggering data transmission when an interesting event was detected. Support for this functionality would require all nodes to analyse a time window of received data samples and, if a preconfigured pattern was detected, trigger the transmission of a warning event to the remote laboratory station along with the collected data.

Acknowledgements. This work was partially supported by national funds through FCT – Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2013. The MIAVITA project is financed by the European Commission under the 7th Framework Programme for Research and Technological Development, Area "Environment", Activity 6.1 "Climate Change, Pollution and Risks".

References

Abolhasan, M., Hagelstein, B., and Wang: Real-world performance of current proactive multi-hop mesh protocols, in: 15th Asia-Pacific Conference on Communications (APCC 2009), 44–47, October 2009. 4321

Akyildiz, I. and Wang, X.: A survey on wireless mesh networks, IEEE Commun. Mag., 43, 23–30, 2005. 4307, 4320

Al Basset Almamou, A., Wrede, R., Kumar, P., Labiod, H., and Schiller, J.: Performance evaluation of routing protocols in a Real-World WSN, in: Information Infrastructure Symposium, GIIS '09, Global, June 2009, Hammamet, 1–5, 2009. 4320

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. 4308

Chouet, B.: New methods and future trends in seismological volcano monitoring, in: Monitoring and Mitigation of Volcano Hazards, Springer, Berlin, Heidelberg, 23–97, 1996. 4308

Claerbout, J. F.: Introduction to Seplib and SEP utility software: Stanford Exploration Project 70, 1997. 4321

Durisic, M., Tafa, Z., Dimic, G., and Milutinovic, V.: A survey of military applications of wireless sensor networks, in: Embedded Computing (MECO), Mediterranean Conference,June 2012, Montenegro, 196–199, 2012. 4307

Faria, B.: Monitorização geofísica e niveis de alerta do vulcão do Fogo, Ph.D. thesis, Instituto Superior Técnico, Lisboa, 2010 (in Portuguese). 4308

Geoffrey, C. and Welsh, M.: Volcano monitoring: addressing data quality through iterative deployment, in: Deployments and Design Frameworks, Springer Science, US, 2010. 4309

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, October 2006, Boulder, Colorado, ACM, New York, NY, USA, 71–84, 2006. 4310

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K.: System architecture directions for networked sensors, Sigplan Notices, 35, 93–104, 2000. 4309

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, September 2008, Warsaw, 10 pp., 2008. 4320
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, August 2007, ACM, New York, NY, USA, 382–387, 2007. 4310
Kurin, E.: Computers for seismic processing and imaging: a performance study, in: SEG Annual Meeting, San Antonio, Texas, 2451–2454, 2007. 4321
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. 4308
Mogaibel, H. and Othman, M.: Review of routing protocols and it's metrics for wireless mesh networks, in: Computer Science and Information Technology – Spring Conference, 2009, IACSITSC '09, International Association of, 62–70, 2009. 4320
Murillo, A. E. and Bell, J.: Distributed Seismic Unix: a tool for seismic data processing, Concurrency-Pract. Ex., 11, 169–187, 1999. 4321
Rodriguez, I. V. and Sacchi, M. D.: Development of a guided user interface for microseismicity analysis, in: Recovery CSPG CSEG CWLS Convention, Calgary, Alberta, May 2011. 4321
Wang, X., Zhao, X., Liang, Z., and Tan, M.: Deploying a wireless sensor network on the coal mines, in: Networking, Sensing and Control, 2007 IEEE International Conference on, 324–328, 2007. 4310
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, Seattle, Washington, USENIX Association, Berkeley, CA, USA, 381–396, 2006a. 4309
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. 4307, 4309
Yick, J., Mukherjee, B., and Ghosal, D.: Wireless sensor network survey, Comput. Netw., 52, 2292–2330, 2008. 4309

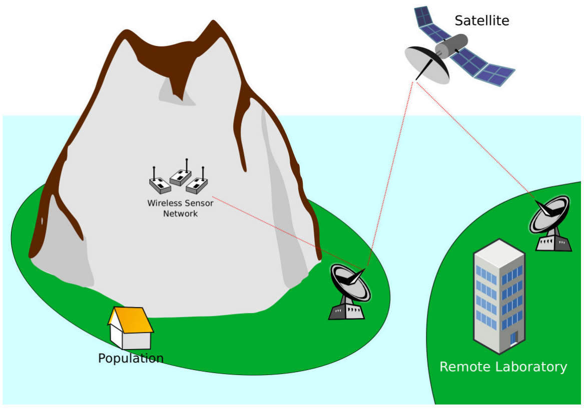


Fig. 1. Global architecture.

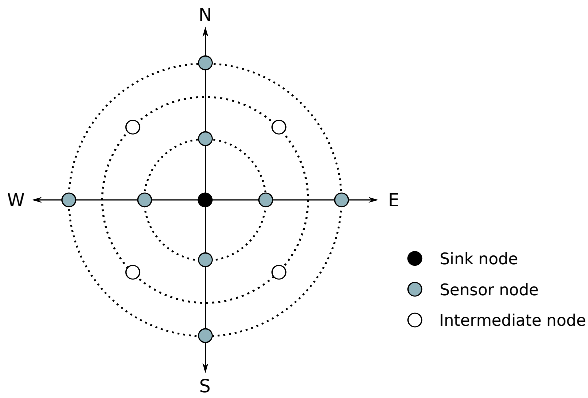


Fig. 2. Sensor/sink node topology.

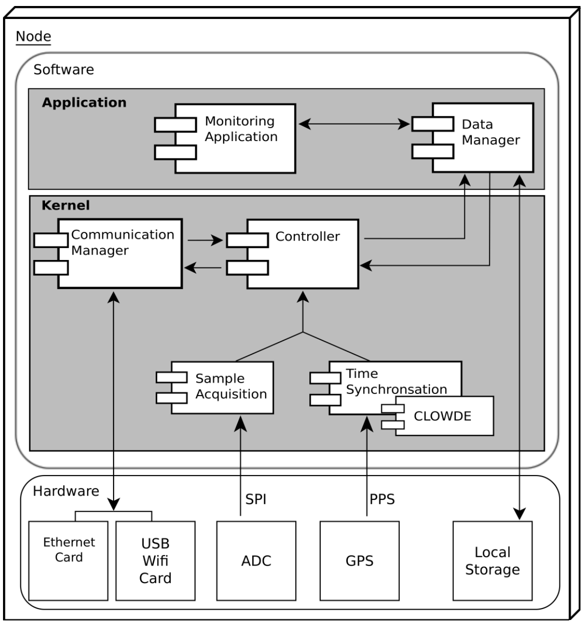


Fig. 3. Node architecture.

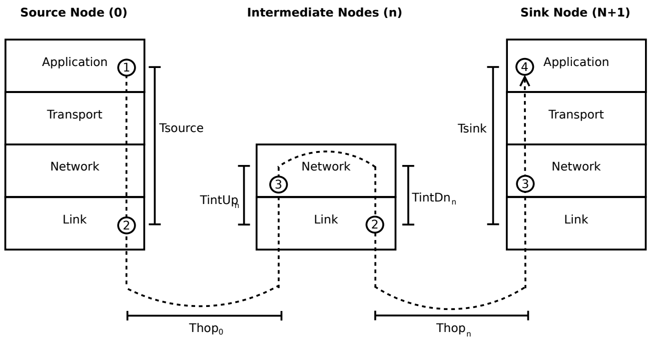


Fig. 4. Time synchronisation time retrieval points.

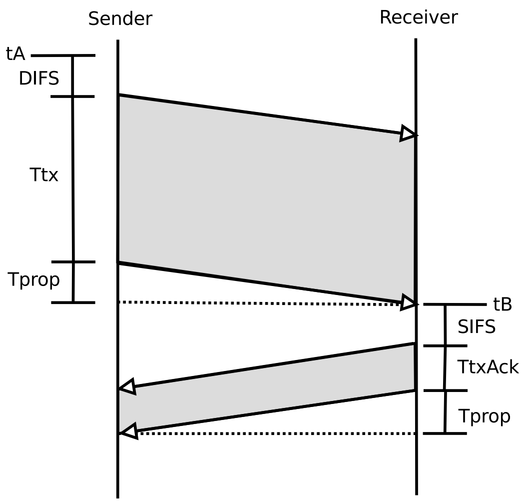


Fig. 5. Successful packet transmission in 802.11 networks.

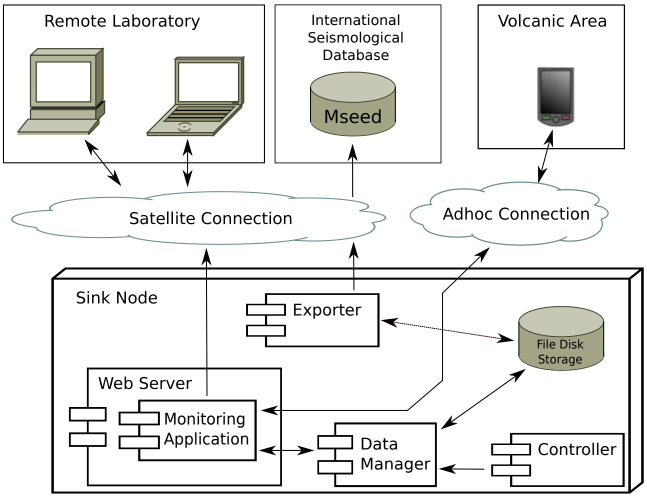
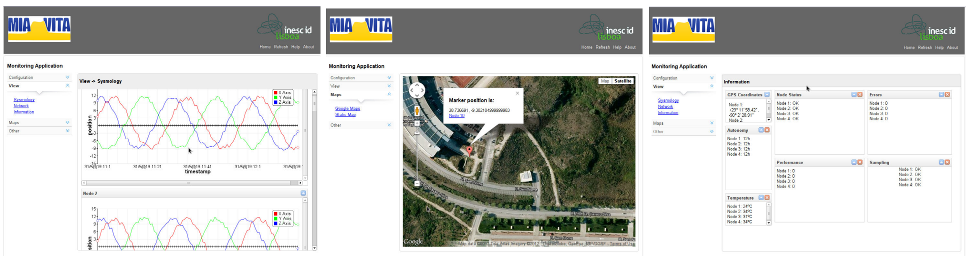


Fig. 6. Monitoring application architecture.



(a) Sample visualization (b) Node location (c) Sensor network status

Fig. 7. Monitoring application screenshots.

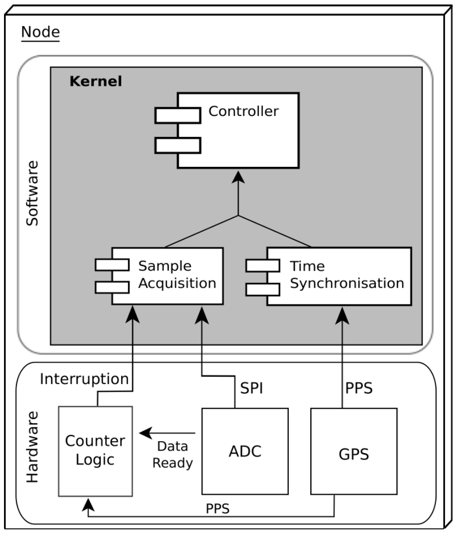


Fig. 8. Sample acquisition module.

Fig. 9. Case front.

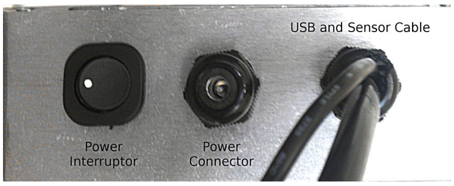


Fig. 10. Case back.

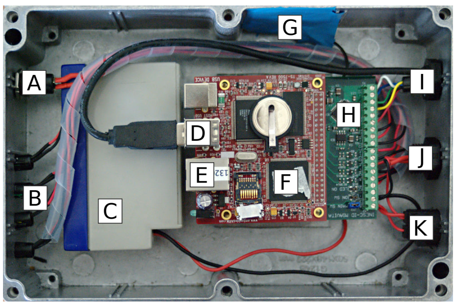


Fig. 11. Open case.

Fig. 12. Node deployed on the terrain with wireless antenna installed on the ground and geophone sensor.

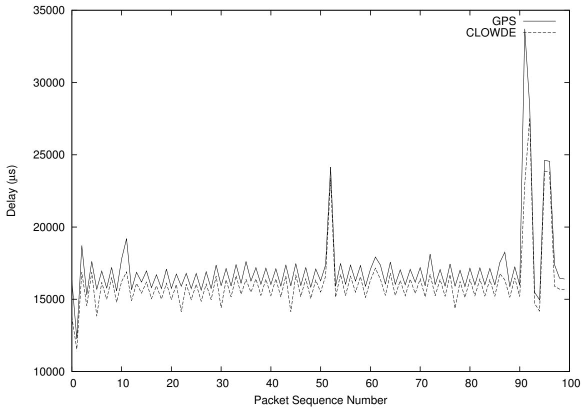


Fig. 13. Sample timing delay.

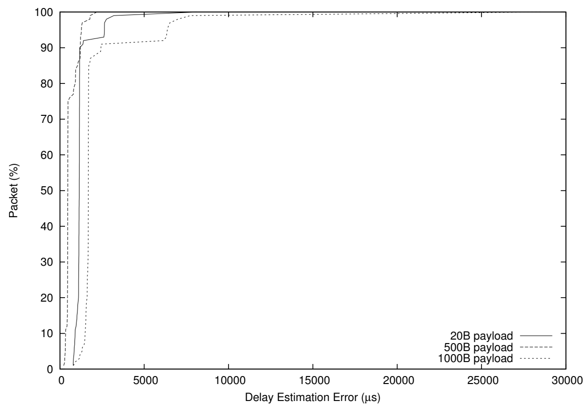


Fig. 14. Clowde delay correction error.

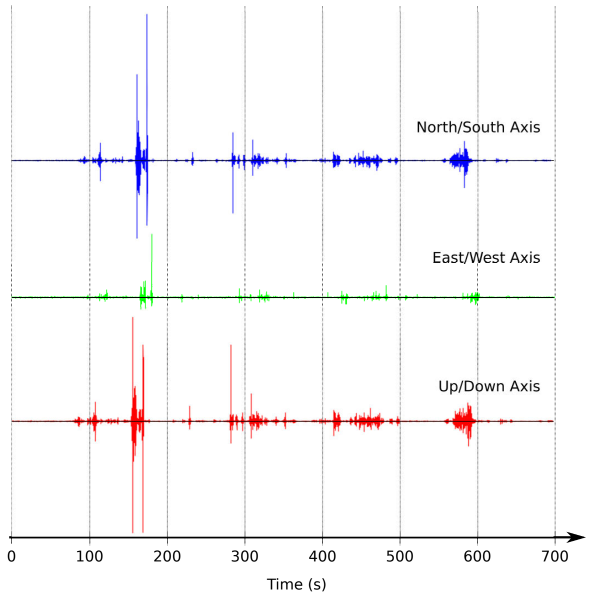


Fig. 15. Recorded seismic signals.