Making the Invisible Audible: Acoustic Interfaces for the Management of Wireless Sensor Networks

Jacques Panchard
Laboratory for computer communications and applications (LCA)
EPFL
jacques.panchard@epfl.ch

Abdelraouf Ouadjaout
University of Science and Technology
Algeria

Julien Freudiger
Laboratory for computer communications and applications (LCA)
EPFL

Enrico Costanza
Media and Design Laboratory
EPFL

Behnaz Bostanipour
Laboratory for computer communications and applications (LCA)
EPFL

Jean-Pierre Hubaux
Laboratory for computer communications and applications (LCA)
EPFL

Abstract

Wireless sensor networks (WSNs) represent an enabling technology for a whole range of applications, such as environment monitoring or event detection/alert reporting. Their limited resources, however, make them a challenging tool to handle in the field. In particular, they lack a proper display, which makes them difficult to deploy, and to manage once they are deployed.

In this article, we present Sensor-Tune, a light-weight deployment and maintenance support tool for wireless sensor networks. This tool is based on an auditory user interface using *sonification*. Sonification refers to the use of audio signals (mostly non-speech) to convey information. We explore the potential of this approach, in particular how it allows to overcome the inherent limitations of visual interfaces. We then justify our design choices, and present typical WSN applications where sonification can be particularly useful. Finally, we present the prototype that we built, and describe a user experiment that we conducted in early 2008, which is the first reported attempt to put a multi-hop wireless sensor network deployment in the hands of non-specialists.

1 Introduction

Wireless sensor networks (WSN) are widely regarded as an enabling technology for ubiquitous computing and the precise monitoring of human, urban and natural environments alike. However, this promise will be hard to fulfill as long as WSNs remain as difficult to install and maintain as they are today.

It is an experience commonly reported in the literature of the field that deploying a wireless sensor network can be a cumbersome and labor-intensive task [24], [25], [22]. In particular the influence of the environment on network connectivity is often difficult to diagnose due to the limited display capabilities of wireless sensor nodes. These difficulties are exacerbated when the network topology is sparse (for instance WSNs for agriculture), or when the environment is particulary challenging for the radio channel (indoor environment with metallic walls or pipes, etc.)

We draw on lessons learned from a sparse WSN deployment in a rural setting: one major issue in determining nodes' placement is the connectivity between them and its evolution over time. In an environment characterized by dense vegetation, partial line-of-sight and low network density, deploying sensors requires precisely analyzing the connectivity between nodes while they are being installed in the field.

In the current state of affairs, the tools used are ill adapted. The wireless sensors themselves lack a proper interface that would allow precious connectivity information to be obtained. Usually, the only available feedback to the users is through a series of LEDs. A more sophisticated graphic display would not be practical in most cases, as it would consume too much energy to be adapted to a resource constrained device such as a wireless sensor.

With this in mind, the options left to a deployment team are few. The staff can work blindly in a long and painful trial-and-error process. It can use ad-hoc nodes blinking their LEDs in order to assess one-to-one connectivity. Such a measure involves moving around two nodes that run a Ping-Pong application, and constantly observing the LEDs indicating metrics such as Packet Error Rate (PER). Or it can use more complex network monitoring systems, to the extra cost of relying on extra infrastructure or software services.

Selavo et al. developed a portable graphical display for deployment time validation [25]. However, our experience indicates that traditional displays are usually not convenient during a work-intensive deployment task. Indeed, portable devices generally use LCD displays that are difficult to read outdoors, especially on sunny days. It is also to be noted that the necessity to actively look at a signal or a screen is an important distraction from the work to be accomplished in the field. To this day, a light-weight tool that makes it easy to assess the quality of the radio channel *while performing the necessary deployment tasks* is still lacking.

Designing such a tool for the average user is challenging. Currently, deploying a WSN remains a task requiring a high level of expertise, while end-user installation is crucial for cost reduction, scalability and users' acceptance of the technology [5]. In particular, the deployment-support tools that have been developed so far (see Section **??**) require advanced computer skills and knowledge in networking. If WSNs are to become as ubiquitous as foreseen by many analysts, it will be necessary to develop intuitive interfaces for this technology. In this context, an important issue is the ability of untrained users to deploy a WSN effectively by assessing connectivity and placing nodes appropriately.

A deployment and maintenance support tool for wireless sensor networks should satisfy a basic set of requirements. First of all, the system should interfere minimally with the task to be carried out. In particular, it should not require the installation of an extra infrastructure. Because deployments can take place in challenging environments, it should provide information that is easy to read in all circumstances. Finally, it should not require any special expertise to be interpreted by the person in charge for the deployment, while providing expert users with extended information about the network.

The contribution of this paper is three-fold. First, we introduce the application of *sonification* (the use of non-verbal audio signals to convey information) to wireless sensor networks, discussing advantages and challenges of this approach. Second, we present the design and implementation of *Sensor-Tune* (see Fig. 1), a light-weight deployment and monitoring tool based on sonification. To the best of our knowledge, it is the first tangible example of a sonification-based solution to WSN problems. Third, we report on the field evaluation of our tool, showing the impact of our design choice through the analysis of users performance on a network deployment task. This experiment represents the first reported attempt to put a wireless sensor network deployment in the hands of non-specialists.

The rest of this paper is structured as follows: in the next Section, we begin by presenting the state of the art, both in deployment support systems and in sonification. In Section 3, we explain how we used sonification and justify the choices we made in order to make the sound feedback as intuitive as possible. In Section 4, we explain in detail the scenarios and system design. In Section 5, we present a survey on sonification validating our design choices. We present the implementation of our prototype in Section 6. Results from a field experiment are presented and discussed in Section 7. In Section 8, we summarize the contributions of this work and draw guidelines for future extensions.

2 State of the Art

2.1 WSN deployment and maintenance support

In wireless sensor networking, a traditional way to assess the connectivity between two points is to use a ping-pong application that requires two wireless mobile nodes communicating with each other. Uni- and bi-directional connectivity can be assessed in this way.

The idea of using a PDA for field-inspections was mentioned before in several publications. A concrete example is the TASK project [6]. The TASK field tool provides the ability to ping a single node, issue a command to turn on a LED or a buzzer, or to reset the node. Similarly, Ringwald et al. [24] propose an in-field inspection tool on a compact device that not only simplifies the process of collecting information about the nodes state but also enables the actual users of the WSN to perform routine checks such as displaying the network topology, or uploading new firmware versions.

In both cases, the feedback given to the user is visual, not sound-based. As mentioned earlier, we believe sound to be better adapted to deployment tasks, the real challenge being to provide intuitive feedback in this form.

In the area of deployment-support tools, Ringwald and Römer emphasize the necessity to passively inspect the network in order not to disturb it and modify its state [23]. Consequently, they designed a deployment-support network (DSN) that superposes itself onto the network to be monitored, communicating with it on a back-channel. This approach supposes to deploy a second network in parallel with the monitored network, and it requires the extra-nodes to have dual radios.

In contrast, our approach is resolutely light-weight. The interference caused to the network by the exchange of messages with the PDA is tolerable, because we only want to have a limited snapshot of the node's state and of its connectivity with the rest of the network.

Selavo et al. [25] recognize what they call the deployment time validation (DTV) as an "indispensable part of fielding a real system". They developed a deployment time validation approach, named SeeDTV, based on a simple communication protocol between a master node and a deployed network, and an in-situ user interface device, called SeeMote. The feedback is given to the user through a small screen adaptable to a mote.

Our approach is similarly lightweight, but we explore the use of a different interface paradigm, based on auditory feedback. We emphasize its originality and analyze its specific advantages in our context.

2.2 Sonification

Sonification refers to the use of audio signals (mostly non-speech) to convey information. The use of sound to display information is not new, early examples include alarms, the telephone bell, the Geiger counter and medical instrumentation [14]. However, over the last decade this field has drawn increasing attention, mainly because of the growing amount of scientific data to display and the improved technology capabilities to process audio. A presentation of sonification, its usefulness, approaches and issues, as well as a list of resources can be found in [14, 4].

In general in sonification systems, selected features of the data display – such as power onsets, spectral features, crossing of thresholds – are used to control parameters of a sound synthesis process (such as pitch, amplitude, timbre...). The choice of features and synthesis parameters and their relationship is known as *mapping strategy*.

Sonification research has often investigated applications targeted at expert users: either users expert in the acoustic domain (e.g. people with a music background) [19] or experts in the domain of application [11, 16, 21]. Therefore mapping strategies generally leverage on users' advanced knowledge or ability to detect sound qualities in order to provide a rich output that displays multiple data dimensions and at the same time associates each of them to different audio synthesis parameters such as pitch, loudness, duration and timbre. As discussed below, our approach is targeted at non-expert users, so it favors simplicity at the expense of multi-dimensional display.

Different projects investigate the application of sonification to the monitoring of computer networks. The Peep[11] and NeMoS[16] systems provide a framework for associating different network traffic conditions and events to the generation of sound, while Qi et al.[21] focus on intrusion detection and denial-of-service attacks. All of them differ from what we propose in the present paper in that they are targeted at advanced users - network administrators - rather than non-experts. Moreover, no usability experiments are reported for any of these systems.

2.3 WSNs and their end-users

User studies are still a rarity in the field of wireless sensor networks.

Beckmann et al. [5] explored end-user installations of sensors for domestic use. Based on the results of their study, they proposed five design principles to support this task. Their experiment considered the placement of sensors in the environment from the perspective of the sensor operation (proximity of the phenomenon to monitor), but did not consider communication issues. Some of the derived principles, however, still make sense in our context, in particular the benefits of detecting incorrect installation of sensors, and of providing value for partial installations. The authors also emphasize how important it is to "make appropriate use of user conceptual models for familiar technologies", which is what guided us when designing our audio interface (see Subsection **??**), although in a different sense than meant by Beckmann et al.

Williams et al. [27] ran a user experiment about the impact that augmented objects (such as sensor-equipped appliances) will have on the perception people have of their surrounding space. They equipped toys with sensors generating sounds, in order to understand how people will "encounter and understand these spaces, and how they will interact with each other through the augmented capabilities of such spaces". The authors of the study used an auditory interface; however, unlike our present work, they were not interested in the specificities of sound as a helper for WSN deployments. The feedback provided was not intended to improve user performance with the system.

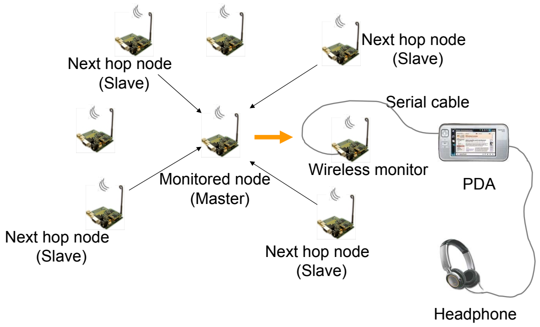

Figure 1. Sensor-Tune system: The monitored node has a wireless connection with the field manager, and queries its immediate neighbors for connectivity information. Both the quality of the local link, and the *distance* to the sink are taken into account.

3 Sonification for Sensor Networks

In this section we outline the advantages that sonification can bring to the field of sensor network deployment and management, compared to the use of graphic displays. We also identify a number of challenges that need to be considered. Later in the paper we will discuss how we specifically addressed some of these in the design and development of our deployment-support tool.

3.1 Applications

The precise description of the system model that we use is provided in Section 4. For the time being, let us only mention that we consider a self-organized multi-hop data collection network where nodes send packets to one or several base stations (sinks) either periodically or as responses to local events. We also suppose that the nodes are installed manually, and not at random.

In this context, we want to design a deployment and maintenance support tool for wireless sensor networks that allows primarily for the monitoring of the connectivity of a node with the rest of the network. This depends on the quality of the radio channel between the monitored node and its neighbors, and on the general topology of the network.

A first application is to inform the user about the appropriateness of nodes' locations. Since the radio channel can vary considerably due to the presence of natural obstacles or interferences with other systems, it is important to get immediate feedback while nodes are being installed. In this way, one can move nodes in order to find a better radio channel whenever possible, or to install efficiently relaying nodes if needed.

A second application is to report on a node's activity after it has been deployed. In a multi-hop data collection network, it can be difficult to identify the points of failure when some nodes start reporting erratically or stop transmitting altogether. In this case, one wants to be able to interrogate the nodes individually about their recent history in order to assess their connectivity status.

There are also other possible use cases of an in-field inspection system using sonification. For instance, to verify the proper functioning of the probes attached to a single node. A sound feedback would allow to check the responsiveness of a given probe.

Also, if security becomes an issue, the detection and localization of possible attacks - such as jamming or malicious modification of the routing topology - are important features of the network. Here again, a sound feedback can inform a maintenance person present in the vicinity of a potential problem.

A precise description of the first two use cases will be provided in Section 4. For the moment, we are building a general case for sonification in the context of WSNs.

3.2 Advantages

The deployment and monitoring of WSNs normally requires users to physically navigate in the environment. In the deployment case, it is necessary to physically place the nodes in suitable locations, both in terms of sensing ability and radio coverage. Nodes' maintenance may also require physical proximity because of the wireless sensors' limited radio range. Navigation is primarily a visual task, which may be particularly demanding in areas that are not easily accessible. The use of auditory displays can be highly advantageous in these situations, because it frees completely the users' visual resources, eliminating the need to switch visual attention between the display and the environment. This visual attention switch is known to be a frequent cause of distraction. For instance, it has been shown that medical students, faced with the concomitant tasks of simulating an operation on patients while monitoring several of their health parameters, performed better when these parameters were represented as sounds rather than graphs [9]. Similar experiments performed on drivers and pilots led to comparable results [3]. Moreover, considering that the most common portable graphic displays are hand-held, the use of audio outputs also frees the users' hands, which can be used to support or balance the body in impervious situations.

From the hardware and physical construction standpoint, an audio display, such as loudspeaker or headphones, presents considerable advantages compared to a graphic display. Audio displays are available at a fraction of the cost of the visual counterparts, and the same is true for the rendering and display driver systems. Because audio requires less processing power, the power consumption is also reduced, which is highly desirable in the context of WSN.

From the ergonomics point of view, visual displays are often problematic for outdoor usage. Under bright daylight, the contrast provided by common LCD screens is often insufficient. Graphic screens also tend to be more fragile than their audio counterparts, which can be problematic in remote or industrial environments, where WSNs are often deployed [15], [12].

Sonification applications in other fields show the potential of the human ear to integrate simultaneously several dimensions of information into a single auditory experience. Experiments with auditory display of scientific data [17] tend to confirm "the effectiveness of auditory display in conveying information and complex structures". Sonification has been demonstrated to be effective for the human recognition of patterns in data, both from experts [19] and non-experts. For example it has been shown [20] how in the context of physiotherapy it can be helpful to create real-time sonifications corresponding to the patients' movements and to let them compare the sounds that they produce to the target sound of a healthy person.

Another appealing aspect is that "sonifications can allow alternative perceptions and new insights into the data"[4]. This can give a certain level of intuitive understanding of specialized data sets to non-specialists. As it is illustrated below, with the design of evaluation of our system, sonification allows for the definition of interface metaphors that can be well understood from novice users but at the same time convey fine details to trained users.

3.3 Challenges

In order to apply sonification to the context of wireless sensor networks, it is important to consider domain-specific constraints and challenges. The interpretation of sound by a user can be decomposed into two parts. First a sound creates a *sensation* - a first contact between the sense organ and the stimulus. Shortly after comes the *perception* of the sound, namely the attempt to identify and classify it [13]. In a system using sonification for extended periods of time, the sound must be designed not to generate fatigue in the sensation part.

At the same time, it must be complex enough to convey information during the perception phase. In particular the design of the mapping strategy has to take into account:

- Prolonged use: deployment and monitoring sessions can span from a few minutes to several hours. It is hence important that the interface sound be pleasant or at least not annoying over an extended period of time.
- High-level metaphor: the overall impression of the interface should present a sound easy to interpret , if necessary conveying integrated and preprocessed information, making the tool accessible by a non-specialist.
- Low-level details: specific aspects of the sound should allow advanced users to perceive detailed information about the status of the network or the node under exam.
- Ergonomics: The tool must work well in an outdoor environment. Sounds used in the interface need not to be confused with ambient sounds.
- Non-invasiveness: The tool must not disrupt network's operation.
- User acceptance: The tool must be acceptable regardless of the cultural background of the users.

Using audio output in noisy environments (such as construction sites, highways or windy environments) can be problematic. The use of headphones with efficient sonic insulation could be a potential solution. However, the trade-off between sound insulation and an awareness of the environment has yet to be carefully examined.

3.4 Signal and Noise Metaphor

To address the challenges described in the previous subsection, we propose here an audio metaphor adapted to sonification for WSNs. The starting point for the design is the following observation: For the deployment and maintenance of WSNs it is generally possible to define a *good* or *desirable* state in which the network is in a *working* state, all nodes being active and connected; and a *bad* or *undesirable* state in which the system is in a *non-working* state, sensing and communication not functioning properly. These states may just be ideal and conceptual, because in reality the *good* state may correspond to several actual network configurations. However they can be easily understood by even non-expert users. In fact, users will normally maintain *some model* of how the system actually works, which may or may not reflect the reality – depending, among other things, on their technical literacy.

Based on this observation we decided to associate the *desirable* state to a sound that can be immediately identifiable as pleasant and undistorted, and to use a gradual degradation of this sound to signify that the system state moves away from the desirable condition. The proposed mapping strategy can also be interpreted as a metaphor for the tuning of an FM radio, an action that is familiar to most people around the world.

The emphasis is *not* on realistically mimicking the FM tuning effect, but just on providing users with a model easy to interpret. Excessively realistic metaphors are known to be problematic in HCI [26]. The proposed strategy leverages the assumption that even non-expert users of WSNs will have *some* understanding of a system relying on radio transmission.

The use of sound noise or distortion seems not to be very common in the auditory display literature, despite its strong metaphorical valence. This is perhaps due to the concern of confusing degradation generated by the interface with *real* degradation affecting the system. The advent of digital technology, however, allows for an easier control of the presence of noise or distortion, to the extent of completely eliminating analogue noise, as demonstrated by the adoption of *comfort noise* in digital communication systems [18].

3.4.1 What Sound to Play?

The proposed sonification strategy requires the choice of what we defined as a "pleasant and undistorted" sound. One advantage of the proposed mapping is that such a sound can be selected by the end-users according to their taste. However, some consideration needs to be taken into account in this choice. First and foremost, the sound needs to be easily distinguishable from the distortion. Second, it should not generate fatigue in the listener. Finally, in order to optimize the use of memory, it is convenient to use an audio loop of relatively short duration .

Our experience suggests that repeated speech clips can very quickly induce fatigue. Natural sounds (e.g. animals, water, wind) can be considered suitable background sounds, as they do not capture the attention of the auditor and are generally perceived as pleasant, but they are not always easy to distinguish from noise – especially wind and water flow sounds can be quite similar to white noise.

One alternative strategy could have been to translate into audio only the errors or problems in the network, and to have silence signify perfect connectivity. However, while this strategy would have the advantage of limiting the listener's fatigue, as well as reducing energy consumption, we deem it fundamental to give clear feedback about the monitoring tool' status – the use of silence makes it non-trivial to understand if the system is correctly working or the tool is just powered off.

3.4.2 What Degradation?

There are several ways to degrade the quality of a sound, examples include:

- adding noise (including different types);
- reducing the resolution (increasing quantization noise);
- modifying the pitch or playback speed;
- reducing the signal bandwidth (bandpass filtering);
- convolving with another sound;
- adding a delayed copy of the same sound (echo);
- multiplying with a square wave of variable duty cycle (introducing silent gaps for duty cycle $< 100\%$).

Each method has its advantages and disadvantages, in terms of computational complexity and control of perceived degradation. Different types of degradation can be used at the same time, mapping the intensity of each of them to a different variable (e.g. signal power level, packet error rate, SNR, ...). In this way, novice users can perceive the general status of the system from the overall sound quality, even without distinguishing different degradation types, whereas advanced users can get more precise information recognizing what exactly is affecting the network.

As detailed later in this paper, for Sensor-Tune we used two types of additive colored noise to represent local and global properties of the network.

4 System Design

4.1 System Model

The context we consider is a multi-hop data collection network where nodes send packets to one or several base stations (sinks) that are connected to a server either directly or through a bridge (typically GSM or 802.11). The traffic can be either periodic, query-based or event-based. We assume that nodes are capable of organizing into a data collection tree (or forest in the multiple-sink case). A critical issue for each node is to find a suitable parent to route its data towards a sink.

The placement of each node is constrained by the landscape and the data it is supposed to collect. This means that for each node to install, there is a region within which this node must be placed. We do not make any assumption about the size of the region, as this depends on the type of application considered.

We assume that the radio channel is highly unpredictable.

Figure 2. Example of usage of Sensor-Tune, with a sensor difficult to access physically

That can apply to both indoor and outdoor environments, depending on the presence of obstacles and interferences.

We must deploy a total of N sensors. $M < N$ sensors are already deployed. We add nodes one by one and want to place them as well as possible within their allowed region, which is determined by the phenomenon to observe. Extra nodes can be deployed in between the measurement points to insure connectivity, but as these nodes do not provide useful data, their number should be kept at a minimum.

4.2 Tool and Scenarios

4.2.1 Sensor-Tune: A Sonification Toolkit for WSN

We designed and implemented a deployment-support system that we call Sensor-Tune. It consists in a lightweight tool integrating a wireless sensor with a sonification module connected to earphones. This tool can interact with any node present in the network (see Fig. 1). It is designed to be carried easily by any person deploying the network.

A small set of buttons are used to turn Sensor-Tune on and off, and to choose the mode of operation of the tool. Once Sensor-Tune is started in the proper mode of operation, no visual interaction with Sensor-Tune is necessary. In this way, the staff can focus on handling the nodes to deploy or to maintain, possibly in places that are difficult to access and require full physical availability (see Fig. 2).

The acoustic feedback is intended to convey information that cannot be easily retrieved due to the limited interface capabilities of wireless sensors. As a proof of concept of the use of sonification in this framework, we decided to implement the following two use cases:

- Deployment support: Optimization of the placement of new nodes into a multi-hop network,
- Maintenance tool: Retrieval of recent connectivity history of a deployed node

4.2.2 Scenario 1: Live Information

In this scenario, we want to assess the connectivity of nodes as we are deploying them. In order to achieve this, we imagine the following flow of events:

1. The member of the deployment team carrying Sensor-Tune produces a new node from his stock.
2. As he/she turns it on, this node connects itself to Sensor-Tune and probes its neighbors in order to assess their potential as a parent.
3. This information is relayed to Sensor-Tune and displayed in real-time as audio data.
4. The deployment staff positions the node based on the obtained feedback.
5. When the node has been placed, a new node is turned on, which automatically takes over, while the previous node enters its normal mode of operation.

In the event of total loss of connectivity, a continuous tone is played in order to spare the ears of the user.

We implemented this scenario, evaluated it and used it for the experiments that we describe in Section 7.

4.2.2.1 Information and Metrics

When deploying one sensor i, we evaluate its connectivity with its neighbors C_{local}, and its "distance" to the sink C_{global}.

Local connectivity: For C_{local}, we use the information about the quality of the radio link between the node and its neighbors. For this information, we use the Packet Error Rate (PER) from this node to all its neighbors. It is defined as follows:

$$PER = \frac{(sf + nr + nack)}{(ack + sf + nr + nack)} \tag{1}$$

where

- sf is the number of packets whose emission failed at the sender
- nr is the number of packets for which the routing layer could not find any route
- $nack$ is the number of packets that were not acknowledged
- ack is the number of acknowledged packets

General topology information: This information, C_{global}, reflects how well the neighbors of the current node are positioned in the network with regard to the base station. For each potential parent, we take into account the metric of the multi-hop protocol used. This metric is customizable. Generally, it is based on the hop count and/or the aggregation of the connectivity levels (packet delivery ratios) of the nodes along the path to the base station. A good metaphor for this metric is the "distance" with the base station.

4.2.3 Scenario 2: Connectivity History

In this scenario, the connection is with an individual node, without local communication with its neighbors. It is assumed that this node has recorded relevant information over the last 24 hours, typically storing parameters in its flash memory for a succession of time steps that last 10 minutes each. It will replay it on-demand. The parameter that can be easily stored in the flash is the PER, information coming from the radio itself. More precisely:

1. The maintenance staff carrying Sensor-Tune walks around the deployment area
2. Automatically, Sensor-Tune beacons the neighboring nodes
3. When a node hears Sensor-Tune's beacon, it answers
4. The nodes that answered are queried sequentially in a FIFO manner
5. The history is downloaded to Sensor-Tune, where it is played a fixed number of times (the total sound sequence should last a few tens of seconds)
6. The user can interrupt the sequence by pressing on a button, either deleting it or saving it in the memory for later retrieval and finer grained analysis.

We implemented this scenario, but have not yet fully evaluated it.

4.2.4 Other Scenarios

4.2.4.1 Local Connectivity

In the use case described above, we only monitor a node's parent. This means that when moving a node, we do not know anything about its connectivity with its potential children. If we want to know what effect the moving of a node will have on its children's connectivity, we need to test the Packet Delivery Ratio, namely the percentage of packets received by this node coming from its children. The sonification technique for this use case can be directly derived from the previous ones.

4.2.4.2 Probes Operation

When deploying a node, the proper operation of the probes can be tested as a sound as well. In this way, the deployment staff can make simple tests such as covering a solar radiation sensor, warming a thermometer, filling a rain gage, etc. In this case, the noise metaphor does not hold anymore. An appropriate sonification would be the synthesis of a sound whose pitch varies as a function of the sensed data.

4.3 Protocols

When monitoring a wireless network, it is important to do so in a minimally invasive way. Ideally, a fully passive system should be used. In our case, however, it is not possible. Most of the time, indeed, the node that we monitor is not part of the network yet. It needs to interrogate its neighbors about their position and to run a decision process to choose its parent. In a normal operation mode, this procedure takes time, and we cannot rely on the regularly exchanged routing messages to send instantaneous feedback to the user as he/she moves the node to find its best placement.

Accordingly, we designed a communication protocol that provides real-time connectivity feedback, and is as minimally invasive as possible. The number of messages exchanged is compatible with a typical environmental monitoring application, even if it might conflict with applications requiring very high data rates and a nearly instantaneous response. We analyze its overhead in Section 4.3.4.

4.3.1 The Actors

We distinguish several actors in the unfolding of the protocol.

1. Sensor-Tune: the monitoring device.
2. Master: the node to deploy, which will query its neighbors for a suitable parent.
3. Slave: any node in the neighborhood of the master node, which is going to answer its queries.

The basic idea behind Sensor-Tune operation is to run a one-hop multicast protocol between the node to deploy (the master) and its neighbors. The radio link between the PDA and the master is used to bootstrap the process, to forward periodically data to the PDA, and to switch nodes.

4.3.2 Live Data

Fig. 3 describes the exchange of messages for this use case.

1. Sensor-Tune receives a message from the PDA as soon as the latter is ready to accept candidates (meaning that the user pressed the *on* button).
2. When a new node is turned on in the vicinity of Sensor-Tune, it sends a INIT message to it, thus applying to become a master.
3. If it does not receive an answer within a given (customizable) time, it enters its normal mode of operation (meaning that Sensor-Tune was *off* or not present).
4. If Sensor-Tune hears the INIT message, it answers with a START, turning the new node into a master.
5. At which point, the master starts a series of rounds that last one second each. During the first 500ms, it sends bursts of INFO_QUERY messages (customizable, but typically 10), and waits for an INFO_RESPONSE from its best potential parents during the next 400ms. In order to reduce collisions, the neighbors use a random back-off timer during this period.
6. The last 100ms of each round are left for normal data traffic to take place.
7. Based on the metrics we defined in the previous sections, the master will select the best potential parent and forward its local and global connectivity parameters to Sensor-Tune. This information will ultimately be translated into sound.
8. When we are happy with the placement of the node, we simply turn a new node on. Upon reception of the new INIT message, Sensor-Tune first stops the previous master (which enters then the normal mode of operation), before starting the new one.

In order to avoid too many answers from the neighbors, the target value for the PER is included in the INFO_QUERY message. We denote it $C_{global}*$. This value depends on the last value received (and increases exponentially if no messages have been received in the last rounds). Nodes only

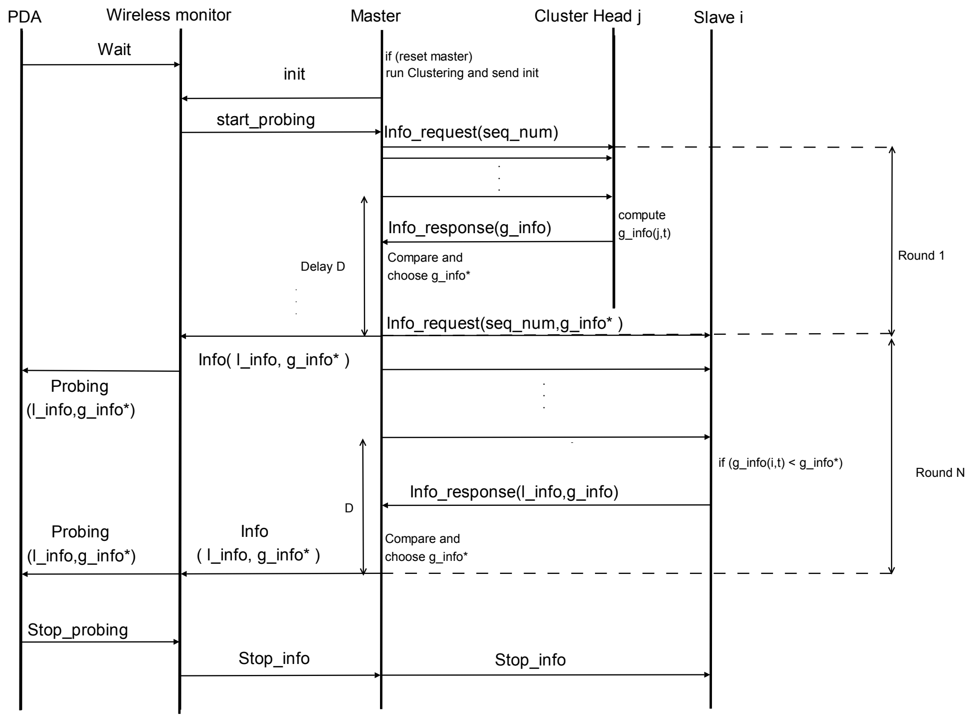

Figure 3. Communication protocol for the *Live Data* use case. The master initiate the session, first querying the cluster heads, then all the nodes in its neighborhood. N requests are sent, then the neighbors have a time slot of duration D to answer if their local and global connectivity metrics are good enough

respond if, based on the INFO_QUERY messages they received, their own PER from the master combined with their own distance to the base station is close enough to $C_{global}*$.

$C_{global} < C_{global} * \pm\Delta_C$

Where Δ_C is a parameter. A higher value of Δ_C improves the responsiveness to channel variations, but increases the traffic.

$C_{global}*$ is updated at each round with the best value of the last round.

4.3.3 Clustering

Initially, the value of the threshold has to be set arbitrarily, which means that many answers can be expected. In order to avoid a congestion at this point, we use an algorithm that partitions the network in different clusters. Each cluster is composed of a *cluster head* and a subset of its single-hop neighbors, the *cluster members*. At the first round of the *Live-data* protocol, only *cluster heads* can respond.

The clustering protocol takes place at the deployment of each new node (see Fig. 4).

1. The new node sends a *Request* message to the network, and starts a one-shot random timer.
2. A node whose timer is expired sends a packet to all its single-hop neighbors, declaring its type as a *cluster-head*.
3. A node that receives a *Request* message will answer to it by a message that declares its type (*clustered* or *non-clustered*).
4. If the new node receives a message from a *cluster head*, it stops its timer and becomes *member* of the corresponding cluster.

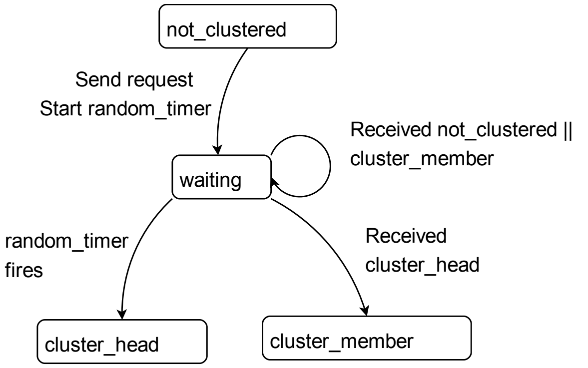

Figure 4. A description of the clustering algorithm

Each cluster is uniquely identified by the ID of its cluster-head.

4.3.4 Overhead

The protocol described above is based on a fast exchange of packets between a new node and its neighbors. During each one-second round, at least 10 messages are received by the neighbors. The number of messages that are sent back depends on the number of neighbors and on the quality of their link with both the new node and the base station. We tested this scheme successfully for up to 5 potential parents. If, above this limit, messages are lost due to collisions, this will not affect significantly the performance of the system, because there will be more than enough parents to choose from. The resort to clustering ensures a successful bootstrapping of the protocol.

Another question is whether this protocol will have an effect on the operation of the WSN main application, because it is communication-intensive, if only for a short period of time. If we are installing a whole network from scratch, the possible disruption is not an issue, as the network is usually not supposed to be fully operational at the moment of deployment. If a node is to be added at a later stage, while the network is operational, we may disrupt the network operation locally during the period of time it takes to install the new wireless sensor. As this task typically takes from a few minutes to up to an hour, this is not a problem for a typical environmental monitoring application, with data rates in the order of the minute or more, without tight response-time constraints, and with some tolerance to errors or missing data. For an alert-based system, the 100 ms window will allow for the delivery of an alert message even in a dense network.

Precise experiments with both high data-rates and tight response-time requirements are still to be performed to find precisely the limits of the system.

4.4 Sonification Mapping Strategy

As outlined in section 3.4, the interface is based on a simple yet powerful model: a pleasant sound indicates that the network is in good condition, whereas additive colored noise indicates a degradation in the network status. The tool allows us to monitor one node at a time. The existence of a connection between the current node and the sink is represented by a piece of music $s_m(t)$, corrupted by an amount of additive colored noise depending on the connection quality $n_g(q,t)$. This can be expressed as follows:

$$s_o(t) = s_m(t) + n_g(q,t) \tag{2}$$

where $s_o(t)$ is the sound output, t is time and q is the connection quality.

The base sound $s_m(t)$ can be selected according to the taste of the end-user, so that different cultural backgrounds can be accommodated. As discussed above, however, it is important that the sound chosen is easily distinguished from noise. For minimizing storage requirements, an audio loop is used for $s_m(t)$, selected to be long enough not to be annoying and to loop in a seamless way (discontinuities could be perceived as signal degradations). In the prototype evaluation described below we used a 16-second clip of classical piano music.

The noise signal is the weighted sum of two distinct colored components: a lower frequency component indicates local connectivity – the PER between the node and its parent – and the other component indicates the global connectivity – the routing metric of the current node's parent.

This can be more precisely expressed as follows:

$$n_g(q,t) = \alpha \cdot n_L(q_{PER},t) + \beta \cdot n_H(q_{route},t) \tag{3}$$

where α is proportional to the packet error rate between the current node and the parent, and β is proportional to the parent routing metric. $n_L(t)$ and $n_H(t)$ are colored noises produced from the same white noise source filtered respectively with a low-pass filter with cut-off frequency 200 Hz and band-pass with center frequency at 2.7kHz and a bandwidth of 20 Hz.

The two components are distinguishable when needed, as shown by the user survey described in the next section. Although normal usage does not rely on users distinguishing the two types of noise, this feature can provide an additional layer of information for advanced users.

We tuned α and β manually, in order to ensure a comfortable level of noise in desirable cases (low PER, small distance to the base station). In particular, β was chosen significantly smaller than α, in order to give more importance to the local connectivity, because this is the parameter of primary importance when placing a node.

The function that we chose to generate noise as a function of PER acts almost linearly for low values of PER and becomes exponential as the PER increases[1]. This is because we want to monitor more closely the low values of PER, as above a certain threshold of packet errors, experience shows a rapid degradation towards total disconnection. At the same time, the human ear functions on a logarithmic scale, so higher intensities of noise become harder to distinguish.

Since the power of the output signal depends on the level at which the user will tune its headphones, we normalized the music waveform and added noise with an increasing envelope. For instance, a 5% PER corresponds to a normalized amplitude of 0.02.

5 Initial Exploration: User Survey

In the previous section, we discussed how sonification techniques, such as altering a sound file with noise, are useful for the deployment of WSNs. We suggested the use of the PER and C_{global} to alter a sound file with high and low frequency noise, respectively. In this section, we describe the results of a *user survey* which explores the perception of noise by users.

5.1 Description

Given the generic nature of this survey on noise perception, we do not require the users to have any prior knowledge of sensor networks, nor musical predispositions. Similarly, we do not constrain the user auditive environment: Sensor-Tune should be usable in any milieu. The survey is thus available online and was advertised in our university via email[2] to users with different academic backgrounds. To stop the users from taking the survey several times, we use persistent cookies[3].

Sonification techniques are usually evaluated by measuring how helpful they are for users to accomplish their *task*. In our case, we wish to know how precisely can users perceive the *intensity* and the *frequency* of noise variations. The survey is composed of two parts. In the first part, the users are given eight sound files containing a sequence of classical piano music altered with noise of low, middle and high intensity. We introduce noise intensity variations Δ and test

---

[1] This function behaves as the ETX value defined in [7]

[2] http://anonymous/sonification

[3] A stronger authentication mechanism could be used but was not deemed necessary because of low risk of attacks.

Intensity	$\|\Delta\|$	Correct	Entropy
Low ($PER < 10\%$)	0	85%	0.60
	1	100%	0
Middle ($PER \simeq 20\%$)	0	92%	0.40
	2	92%	0.40
High ($PER > 30$)	2	64%	0.94

Table 1. Survey results for the first part. Users seem able to recognize noise intensity variations in low and middle intensities (results based on 24 answers).

whether users can perceive these variations by asking: *Do you perceive a change in the noise intensity?* Among the possible answers, users are asked to choose whether they perceive an increasing/decreasing Δ or no change at all. With this question, we evaluate the granularity of noise intensity perception by users. On our normalized scale, a Δ of $+1$ corresponds to an increase of 0.02 in the noise envelope.

In the second part, we generate twelve sound files of the same piece of music, but this time, not only do we alter the files with varying noise intensities, we also use two types of noise: a low frequency and high frequency noise. We examine whether users could recognize the noise types by asking: *Ignoring changes in intensity, do you perceive different types of noise?* Users are asked to say yes or no. We consider various scenarios where both the intensity and frequency vary (DD), where only the intensity varies (DS), and where none varies (SS)[4]. Finally, we ask users for their age and whether they used a headset while taking the survey.

5.2 Results

Over a period of two weeks, 24 users took the survey online. 95% of the users were in their twenties (18-30) and 66% used headphones. Overall, we did not observe any changes in the quality of answers between users with and without headphones. As suggested in [8], we use the entropy H_q to measure the uncertainty of the answer to a question q:

$$H_q = -\sum_{i=1}^{l} p_{i,q} \log_2(p_{i,q}) \tag{4}$$

where $p_{i,q}$ is the probability to answer i to the q, and l is the number of categories for answers to q. H_q is measured in bits and tells how easy it is for users to identify a sound.

In the results of the first question (Table 1), we found that 90% of the users seem to be able to distinguish noise intensity variations at low and middle intensities ($H_q \in [0, 0.60]$). This is much better than for high intensities ($H_q = 0.94$). This result confirms that because the human ear works in a logarithmic manner, users cannot efficiently recognize noise variations at high intensities. Accordingly, we empirically dimension our system with respect to the noise variations that we introduce when the PER and C_{global} vary. For instance, a value of $+1$ of Δ corresponds to a PER of 5% in our final system, $+2$ to 10%, $+5$ to 20%, and $+30$ to 50%, etc.

Variations	Correct	Entropy
SS	4%	0.24
DS	46%	0.99
DD	100%	0

Table 2. Survey results for the second part. Noise and intensity varies (DD), only intensity varies (DS) and none varies (SS) (results based on 24 answers).

With the second question (Table 2), we observe that people tend in SS and DS cases to aggregate both noise types as one. In SS, they are even convinced that there is only one type of noise being played (i.e., note the low entropy). It appears that users could distinguish two noises only when one noise replaces another over time (DD). In other words, when the relative importance of the two noises changes - the dominated noise becomes dominant - users can distinguish the two noise types.

With Sensor-Tune, a user must first optimize the local connectivity (PER). During this operation, while the PER is not good, the low and high frequency noises will not be distinguishable: the user can concentrate on finding a good location for a node. Once a good location is found, the high frequency noise vanishes, and the low frequency noise appears clearly. Thus, users can alternately focus on optimizing the local connectivity and global connectivity to the base station.

The results of the survey allowed us to verify our system design. We realized that we must carefully select noise intensity variations for the users to be able to notice them, and noise types, for the users to recognize them when necessary.

6 Prototype Implementation

6.1 Prototype Description

In this section, we describe the implementation of a prototype of Sensor-Tune, using a PDA running Linux Maemo (Nokia N800) (see Fig. 6). We emphasize the fact that a real commercial system can be implemented in a much less expensive way than with the off-the-shelf components that we used [5].

The PDA is connected through a serial interface with a wireless device compatible with each node that is to be deployed. The wireless device communicates with the monitored node and forwards the received information to the PDA, where this information is analyzed and passed on to a sound-generator. The user can listen to the sonified data via headphones. Once packed, the system is quite compact (see Fig. 7).

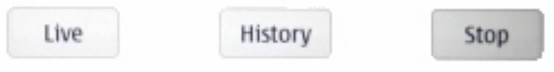

Figure 5. Sensor-Tune simplistic graphical user interface

We used a simple graphical interface to start the tool and

---

[4] D stands for *Dynamic*, S for *Static*

[5] the technical details and software implementation are available on-line at http://anonymous

set it to the desired mode of operation (see Fig. 5). We decided to keep this interface at a bare minimum so that user options can be magnified and be readable in outdoor conditions.

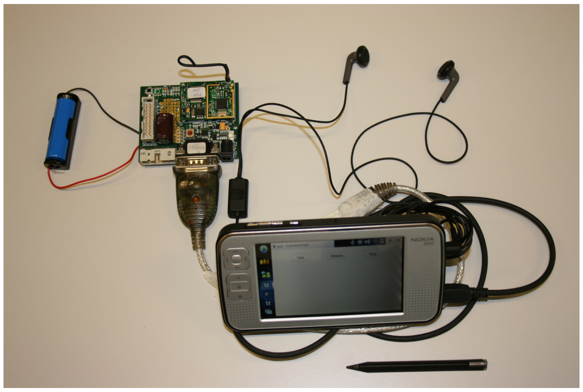

Figure 6. Sensor-Tune prototype: A linux-based PDA connected through a serial port to a wireless sensor

Figure 7. Sensor-Tune prototype once packed

6.1.1 General System Outline

We distinguish the embedded part from the PDA part. On the PDA, a Java subsystem is responsible for the message interface and the data analysis, and a dedicated software, pure data (PD), takes care of the sound generation part.

This section will be broken up in 3 sections: The embedded part, the PDA (with the java data collector/analyzer, and the Pure Data part) and the communication protocols.

6.1.2 Software and Hardware

On the embedded side, we use the TinyOS [2] operating system, as it has evolved to become the preferred choice of the research community to design and implement wireless sensor network systems. We implemented our tool on the tinynode [1] platform.

For the PDA, we used the Nokia N800, which runs Linux Maemo 3.2, thus making it easy to add custom software to it. Java in particular is easy to install. Moreover, it runs PDa, the embedded version of the open source Pure Data sound generator. All different software components communicate through sockets.

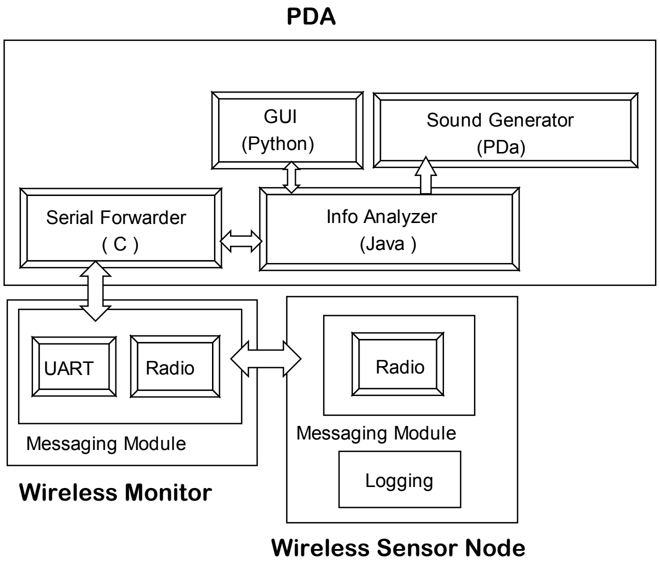

Figure 8. Sensor-Tune functional blocks

6.1.3 Embedded Part

There is minimal change to be brought to any multi-hop application, whose performance we want to monitor. The application comes as a plug-in to be added to the configuration file of the application to be deployed.

The global metric needs to be passed back to Sensor-Tune through an interface, because it depends on the routing protocol to be used. All other metrics are dealt with at a lower layer, so they are independent from the particular context.

6.1.4 PDA

6.1.4.1 Java subsystem

The java subsystem has four tasks:

1. state machine: managing the PDA state machine,in order to keep synchronization with the node to deploy
2. message interface: sending and receiving messages exchanged with the master
3. data processing: analyzing the incoming data, logging them if appropriate, and processing them so that they can be translated into sounds
4. sending the result to the sound generator through a socket.

6.1.4.2 Pure Data subsystem

PD (Pure Data) is a real-time graphical programming environment for audio, video, and graphical processing. PD is an example of ”Dataflow programming” languages. In such languages, functions or ”objects” are linked or ”patched” together in a graphical environment which models the flow of the control and audio. PD is an open source project and has a large developer base working on new extensions to the program. This tool, initially designed for desktop computers, has been ported on small handheld devices running Linux, under the name PDa (PD anywhere) [10].

As mentioned earlier, we chose a method consisting in superposing to the background music two noises at different frequencies: a high frequency noise whose volume increases

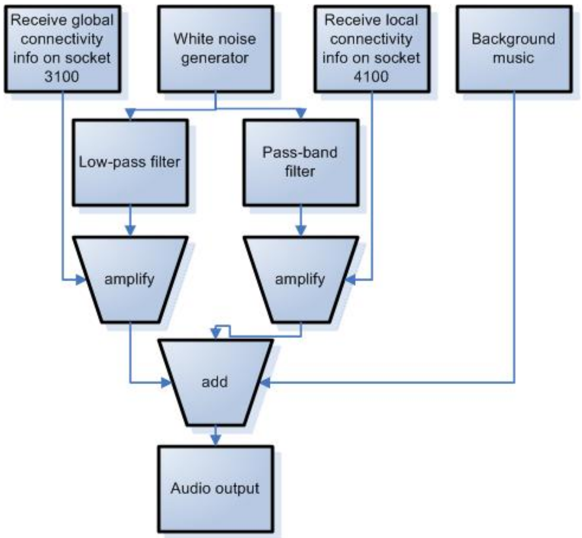

Figure 9. PDA: PD subsystem description, with music and two additive noises

as the packet error rate between the node and its best potential parent increases, and a low-frequency noise (perceptually less annoying) whose volume increases as the distance from the base station in terms of hops increases.

The (simplified) PD subsystem that we designed is described in Fig. 9.

7 Experimental Validation

To validate the proposed design, and in particular the audio based interface, an experiment was designed and performed. The first objective of the experiment was to asses whether, with an appropriate interface, it is possible for non-specialists to deploy a wireless sensor network in a challenging setting with minimal training. We then wanted to evaluate the effects of the auditory presentation independently of the underlying technical system and the actual information presented. For this reason, the audio interface described in Section 4 was compared with a graphical user interface (GUI) that presented the same information on the screen of the Nokia PDA.

7.1 Comparable Graphical Interface

In order to assess the sonification based interface independently of the amount of information provided and of the underlying technical implementation of the system, we decided to compare it with a graphical user interface that would present the same information. Therefore we designed and implemented an interface that displayed two horizontal bars of variable length, as illustrated in Figure 10, one related to the PER and the other to the ETX of the monitored node (see Section 4). We decided to mimic the signal bar common in all mobile phones, so the bars are full when the connection is perfect and become shorter when the connection quality decreases – in other words the length of each bar was inversely proportional to the PER and ETX, respectively.

7.2 Experimental Design

The experiment consisted of 2 network deployment tasks, in each of them subjects had to create a linear multi-hop network that connected specific start and destination points in a building on ...(removed for blind reviewing).

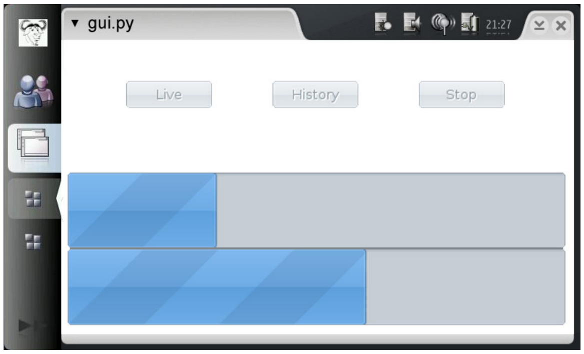

Figure 10. Screen capture of graphical user interface used for the experiment as shown on the Nokia PDA. The horizontal bars convey information about the connection quality.

For both tasks the destination point was the same and it was located in the parking garage in the basement of the building, and marked with an 'x' sign on the floor. The starting points for the two task were on two different ends of the 4th floor of the building (there was a 5 floors distance between start and destination). The building has 5 different stairways and 3 elevator towers and it includes a mix of glass, metal and concrete partitions that attenuate the radio signals of our wireless nodes in different ways (often drastically). A number of movable elements, such as doors and elevators, made the radio path variable with time, which contributed to make the tasks even more challenging, given especially that the experiment took place during business hours, when many people walk around the building.

For each task, subjects had a maximum time of 20 minutes and a maximum of 6 nodes (but emphasis was put on the fact that they could complete the task with less). As a benchmark, both tasks could be completed by experts using only 3 nodes in less then 5 minutes.

Subjects received instructions in written form (to ensure consistency), informing them about the system and the two tasks, asking them to try and complete them as quickly as possible, using the smallest number of nodes as possible, and making the connection quality as good as possible. The instructions were kept concise, with total length of two A4 pages. The instruction simply reported that the audio degradation through noise, or the length of the bars in the GUI indicated the quality of the connection of the current node to the base station, but did not provide any details about the PER nor the ETX. Several participants asked what was the difference between the two bars, but they were answered that they reflected different aspects of the connection quality but that the details were irrelevant to the experiment. After each subject read the instructions, and before the start of each task the experimenter showed the start and destination point, and a specific path between them, even though during the experiment subjects were free to take any path they liked between the two points. A maximum duration of 20 minutes was given for each task, if the subjects did not reach the destination point within this interval, the attempt was considered failed.

All subjects tried both interfaces, each on a different deployment task in alternate order: half of the subjects used the audio interface in the first task and the GUI in the second, while the other half used the GUI for the first task and the audio interface in the second. The two tasks, however, were performed in the same order.

The completion time, the number of nodes needed to achieve the task and the resulting network performance were recorded for all tasks. Participants were shadowed by an experimenter, who took notes about their behaviour and performance. At the end of the experiment subjects were asked to fill a short questionnaire related to their previous experience with computers, with wireless networks and with music as well as their preferences between the audio and graphical interfaces in terms of ease of use, efficiency and overall favour.

7.3 Description

Participants were 14 males, of age between 26 and 48 (avg. 32.6, st. dev. 7.4), all volunteers. All subjects were naive, in that they had not used our system before the experiment and all had no experience in deploying a multi-hop wireless network. Four subjects reported having set-up a home wireless network (802.11).

7.4 Results

Overall, the network deployment was successful in 17 of the 28 trials (60.7%). The first task was completed successfully in 7 of the 14 cases (50%), while the second task was completed sucessfully in 10 of the 14 cases (71.4%). When the audio interface was used the first task was successful in 4 out of 7 cases (57.1%), while with the GUI the first task was successfully completed in 3 out 7 cases (42.9%). For the second task, subjects using the audio based interface were always successful (7 out of 7, 100%) while subjects using the GUI where successful in 3 out of 7 cases (42.9%). The results are summarized in Table 3

Out of the 14 subjects, 5 succeeded in both tasks (2 started with the audio interface, 3 started with the GUI); 2 subjects, who started with the GUI, failed in both tasks; 5 subjects failed in the first task but succeeded in the second (1 of them started with the audio interface and 4 started with the GUI); 2 subjects completed successfully the first task using the audio interface, but failed in the second task using the GUI.

At the qualitative level, we noticed a number of frequent behaviors that were detrimental to the task completion or even resulted in failure. First, most participants tried to “let the radio waves follow their same path” – in particular, most participants tried to “bring” the radio signal down the stairways, even though these are interrupted by a number of glass and metal doors which block the radio waves of the nodes. Often, it was noticed that these participants were aware of the fact that the radio waves can go though walls, but simply did not actively use this information. Only 3 of the 14 subjects attempted to let the wireless connection go through the floor, which results in a more efficient solution. All subjects who attempted this alternative strategy were successful in completing the task and used a minimal (3) number of nodes.

	Task 1 Success	Task 2 Success	Total Success
Audio	4 of 7 (57.1%)	7 of 7 (100.0%)	11 of 14 (78.6%)
GUI	3 of 7 (42.9%)	3 of 7 (42.9%)	6 of 14 (42.9%)
Overall	7 of 14 (50.0%)	10 of 14 (71.4%)	17 of 28 (60.7%)

Table 3. User experiment results: successful completion of the deployment tasks by untrained participants.

A second common source of problems was the fact that the very first node was placed in a position where it was not very well connected with the base station, which compromised the connection of the following nodes to the base station. In turn, the bad positioning of the first node was often the result of the two following behaviours: before choosing the position for a node subjects monitored its connection quality for a period that was too short to notice signal drops due to transient events such as other people passing by, doors opening and closing, or elevators moving; subjects monitored the connection quality only when they were very close to the nodes, and with their body somehow influenced the EM field in favour of the connection, while as soon as they walked away the connection dropped.

Regarding the expressed preferences, 8 of the 14 subjects (57%) indicated the audio interface as easier to use, while 9 (64%) indicated that they deemed the GUI let them perform better, and the same number reported it as generally preferable.

8 Conclusion

Throughout the literature, there is a paradox in the fact that wireless sensor networks are envisioned as the ubiquitous communication technology of the near future, while they remain cumbersome to deploy and difficult to maintain. In this paper, we have investigated a novel approach for interfacing the wireless sensing world, relying on acoustic feedback.

We have presented the advantages of such an approach in terms of deployment efficiency, reliability, intuitiveness and cost, and have developed an original metaphor for the analysis of connectivity based on the metaphor of noise. The implementation of a prototype allowed us to confirm that this approach is promising for wireless sensor networks.

The overall success rate of 60.7% in the experiment indicates that the interface is effective in supporting non-expert users deploying a multi-hop wireless network, validating the proposed design for Sensor-Tune. The results indicate no large differences between the performance with the audio interface and with the GUI, suggesting that the two interfaces perform as well as each other. The additional advantages provided by the audio interface, namely eyes-free and hands-free operation, are therefore available without any penalty compared to a graphic counterpart.

The approach needs now to be validated in the field, through a real-life deployment. This validation should include an improvement whose necessity has been unveiled by the user experiment: from the observation that often one specific link between two nodes is the cause of major problems in the entire network, the modification of the interface so that users can easily select which link to monitor, or even monitor several links at the same time, may dramatically increase its performance.

As of future developments, other applications than deployment support may be implemented and tested, such as history of connectivity, on-board sensors validation, etc. Security applications can also be sought. Finally, transposition of the sonification paradigm to other wireless technologies should be envisaged.

9 Acknowledgements

Thanks to (REMOVED) for helping us put our work in context with Sonification research

10 References

[1] Tinynode wireless sensor from Shockfish. http://www.shockfish.com/tinynode/TinyNode

[2] TinyOS operating system. Documentation available at URL http://www.tinyos.net.

[3] J.A. Ballass. Delivery of Information Through Sound. In G. Kramer, editor, *Auditory Display: Sonification, Audification, and Auditory Interfaces*, volume XVII of *SFI Studies in the Sciences of Complexity*. Addison Wesley, 1994.

[4] S. Barrass and G. Kramer. Using sonification. *Multimedia Syst.*, 7(1):23–31, 1999.

[5] C. Beckmann, S. Consolvo, and A. Lamarca. Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments. *UbiComp 2004: Ubiquitous Computing*, pages 107–124, 2004.

[6] P. Buonadonna, D. Gay, J. M. Hellerstein, W. Hong, and S. Madden. Task: Sensor network in a box. In *EWSN*, 2005.

[7] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *MOBICOM*, 2003.

[8] M. Fernstrom, Eoin Brazil, and Liam Bannon. Hci design and interactive sonification for fingers and ears. *IEEE Computer Society*, 2005.

[9] W.T. Fitch and G. Kramer. Sonifying the Body Electric: Superiority of an Auditory over a Visual Display in a Complex Multivariate System. In G. Kramer, editor, *Auditory Display: Sonification, Audification, and Auditory Interfaces*, volume XVII of *SFI Studies in the Sciences of Complexity*. Addison Wesley, 1994.

[10] G. Geiger. Pda - real time signal processing and sound generation on handheld devices. In *International Computer Music Conference (ICMC)*, 2003.

[11] M. Gilfix and A. L. Couch. Peep (the network auralizer): Monitoring your network with sound. In *LISA '00: Proceedings of the 14th USENIX conference on System administration*, pages 109–118, Berkeley, CA, USA, 2000. USENIX Association.

[12] M. Hatler and C. Chi, editors. *Wireless Sensor Networks for the Oil & Gas Industry*. ON World, 2005.

[13] A. A. A. Ibrahim and A. Hunt. An hci model for usability of sonification applications. In Karin Coninx, Kris Luyten, and Kevin A. Schneider, editors, *TAMODIA*, volume 4385 of *Lecture Notes in Computer Science*, pages 245–258. Springer, 2006.

[14] G. Kramer, B. Walker, T. Bonebright, P. Cook, J. Flowers, N. Miner, and Neuhoff. Sonification report: Status of the field and research agenda. *Prepared for the National Science Foundation by members of the International Community for Auditory Display*, 1997.

[15] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless Sensor Networks for Habitat Monitoring. *ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[16] D. Malandrino, D. Mea, and A. Negro et al. Nemos: Network monitoring with sound. ICAD, 2003.

[17] R. Minghim and A.R. Forrest. An Illustrated Analysis of Sonification for Scientific Visualisation. In *6th IEEE Visualization Conference (VISUALIZATION '95)*, 1995.

[18] H. Newton. *Newton's Telecom Dictionary*. CMP Books, 20 edition, 2004.

[19] S. Pauletto and A. Hunt. The sonification of emg data. In *Proceedings of the 12th Meeting of the International Conference on Auditory Display (ICAD), 20-23 June 2006, London*, 2006.

[20] S. Pauletto and A. Hunt. "the sonification of emg data". In *ICAD*, 2006.

[21] L. Qi, M. V. Martin, B. Kapralos, M. Green, and M. García-Ruiz. Toward sound-assisted intrusion detection systems. In *OTM Conferences (2)*, pages 1634–1645, 2007.

[22] N. Ramanathan, L. Balzano, D. Estrin, M. Hansen, T. Harmon, J. Jay, W. Kaiser, and G. Sukhatme. Designing wireless sensor networks as a shared resource for sustainable development. In *International Conference on Information and Communication Technologies and Development*, 2006.

[23] M. Ringwald and K. Römer. Snif: A comprehensive tool for passive inspection of sensor networks. http://www.vs.inf.ethz.ch/publ/papers/mringwal-snif-kuvs.pdf, July 2007.

[24] M. Ringwald, M. Yücel, and K. Römer. Demo abstract:

Interactive in-field inspection of wsns. In *Adjunct Proceedings of the 3rd European Workshop on Wireless Sensor Networks (EWSN 2006)*, Zurich, Switzerland, February 2006.

[25] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter. Luster: wireless sensor network for environmental research. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 103–116, New York, NY, USA, 2007. ACM.

[26] H. Sharp, Y. Rogers, and J. Preece. *Interaction Design: Beyond Human Computer Interaction*. Wiley, March 2007.

[27] A. Williams, E. Kabisch, and P. Dourish. From interaction to participation: Configuring space through embodied interaction. In *Ubicomp*, pages 287–304, 2005.