First Experiences Using Wireless Sensor Networks for Noise Pollution Monitoring

Silvia Santini*
Institute for Pervasive Computing
ETH Zurich
8092 Zurich, Switzerland
santinis@inf.ethz.ch

Benedikt Ostermaier
Institute for Pervasive Computing
ETH Zurich
8092 Zurich, Switzerland
ostermaier@inf.ethz.ch

Andrea Vitaletti†
Department of Computer Science
University of Rome La Sapienza
00100 Rome, Italy
andrea.vitaletti@dis.uniroma1.it

Abstract

The assessment of environmental pollution levels is a complex and expensive task that public administration and often also private entities are willing or forced to take over. Focusing on the assessment of environmental noise pollution in urban areas, we provide qualitative considerations and experimental results to show the feasibility of wireless sensor networks to be used in this context. We present a prototype for the collection and logging of noise pollution data based on the Tmote invent prototyping platform, using which we performed indoor and outdoor noise pollution measurements. We build upon these first experimental results to depict the potentials and limits of currently available wireless sensor networks prototyping platforms to be used as noise pollution sensors. Furthermore, we present tinyLAB, a Matlab-based tool developed in the context of this work, which enables real-time acquisition, processing and visualization of data collected in wireless sensor networks.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Experimentation, Measurement

Keywords

Wireless sensor networks, environmental monitoring, noise pollution

1 Introduction

While environmental issues keep gaining increasing attention from the public opinion and policy makers, several experiments demonstrated the feasibility of wireless sensor networks to be used in a large variety of environmental monitoring applications. For instance, wireless sensor networks have already been used to monitor bird habitats and habits [11, 17], to investigate the growth model of redwood trees [5], or to study the influence of environmental parameters on the quality of agricultural products [16]. In more recent years, several projects that aim at monitoring environmental pollution parameters in urban areas have been kicked off and are expected to map pollutants distributions with an accuracy that was unimaginable, though desirable, even a few years ago. For instance, the CitySense[1] project will provide a fixed network of 100 line-powered wireless sensors and allow collecting fine-grained air pollution data as well as deliver it in real-time to the users. Fine-grained data collection is essential to foster scientific research and increase the understanding about actual pollutants spreading mechanisms and their influence on human health. Furthermore, any action plan aiming at the reduction of environmental pollutants may fail or be ineffective if no adequate actual data is available for planning and successive validation.

While the above considerations apply to almost any type of environmental pollutant, we focused our attention on the peculiar problems and challenges that arise when considering the assessment of environmental noise pollution in urban areas. Recent studies indeed demonstrated that exposure to environmental noise clearly increases the risk of hypertension, ischaemic heart diseases, hearing losses and sleep disorder, and that it negatively influences productivity and social behavior [8]. Recognizing in noise pollution a serious hazard to health and productivity as well as a source of increasing complaints from the public, the European Commission made the avoidance, prevention, and reduction of environmental noise a prime issue in European policy [1]. Through a recent document, the Commission additionally states that "*more detailed noise modelling/mapping and noise exposure assessment may have to be undertaken in order to produce detailed local action plans*" [9, p.5].

The European Directive 2002/49/EC indeed requires member states to regularly provide accurate mappings of noise levels throughout all agglomerations with more than 250.000 inhabitants and to make this information publicly available through adequate web-interfaces [1, p.15]. However, only few pioneering administrations already provide easily accessible web front-ends exposing noise mapping information to citizens [6, 12]. Furthermore, these maps are actually generated from synthetic data, i.e., approximate noise levels computed by numerical models taking into account typical noise propagation patterns as well as parameters like the (estimated) number of vehicle transits and the actual urban topology. Even though these models allow to gain a first insight into the noise pollution problem, they often provide just inaccurate data. This poses an alarming uncertainty on the effectiveness of noise pollution abatement plans elaborated upon information distilled from this data. To cope with this problem, the European Commission explicitly recommends that *"every effort should be made to obtain accurate real data on noise sources"* [9, p. 6]. As we will detail in the next sections, collecting accurate actual noise pollution

*Partially supported by the Swiss National Science Foundation (NCCR-MICS, grant number 5005-67322).

†Partially supported by the AEOLUS European Integrated Project (IST-015964).

[1] `www.citysense.net`

REALWSN'08, April 1, 2008, Glasgow, United Kingdom.

data relying on the current measurement procedures is costly and cumbersome and does not scale with the demand for higher data granularity. We argue that wireless sensor networks could be successfully used in this context to provide accurate, fine-grained data on noise pollution.

In this paper, we report our experiences in testing the feasibility of commonly available wireless sensor nodes to be used as noise pollution sensors. After pointing out the peculiar characteristics that make sensor networks a perfect fit for environmental noise monitoring applications, we comment on the prototyping platforms we selected for our experiments. We then introduce tinyLAB, a Matlab-based tool enabling real-time data handling and interaction with tinyOS-based wireless sensor networks. Finally, we present and comment on an excerpt of the noise sensing experiments we performed using the Tmote invent prototyping platform and provide our conclusions.

2 Assessment of Noise Pollution Levels

Through a personal in-depth interview with noise measurement experts of the Department for Environmental Noise Protection of the City of Zurich, we learned about today's currently used noise assessment procedures. In particular, we understood that typical noise measurements in urban areas are mainly carried out by designated officers that collect data at a location of interest for successive analysis and storage, using a sound level meter or similar microphone-equipped device. The measuring sessions take place only at few accessible spots and during short time intervals (e.g., thirty minutes). The collected data is then stored in a land register and possibly used to feed computational models providing extrapolated noise exposure levels for all those areas in which no measurement session took place. Furthermore, we got to know that the inaccuracy inevitably introduced by the lack of actual noise measurements becomes critical when the same data is used to develop and validate urbanization projects or traffic management plans.

An additional issue regarding the specific case of the assessment of road traffic noise, lies in the determination of the number of vehicles passing through a road whose average noise level has to be measured or computed. This number is currently either estimated through numerical models or it is extrapolated from manually gathered data, i.e., data collected by a designated officer standing nearby the road and annotating the type and number of vehicles passing by. This method for estimating the total number of vehicles' transits exhibits the the same drawbacks outlined above with respect to the assessment procedure of noise levels, and could equivalently profit from the adoption of wireless sensor networks technology, as we will detail in the following section.

2.1 Assessment of Noise Pollution Levels Using Wireless Sensor Networks

Collecting fine-grained noise measurements through the manual collection procedure described above is clearly inefficient and expensive. Nevertheless, since the need for higher granularity of noise data in both time and space has been explicitly stated by the European Commission, public administration will likely be required to invest more human and material resources to provide enough actual data on noise sources. In this scenario, wireless sensor networks represent a promising technology that can overcome the drawbacks of the current noise data collection procedure as well as open new monitoring opportunities. Indeed, adequately equipped sensor nodes could be deployed over an area of interest and collect noise pollution readings over longer periods of time, operating unattended and requiring human intervention only for network installation and removal. Noise pollution data reported to a central sink could then be easily stored in a land register and subsequently be used to produce noise maps and validate previously estimated noise levels. Wireless sensor networks could bring significant improvements in particular in the assessment of noise pollution due to vehicular traffic on urban roads, since its fine-grained observation would allow for the design of better traffic management plans aiming at reducing the noise exposure in affected neighborhoods. Furthermore, the assessment of road traffic noise also requires estimating the average number of vehicles passing-by at daytime, evening and night and the average noise level for each vehicle pass-by [4]. We will show in section 5 that these figures could be extracted from collected noise levels after adequate processing and possibly in combination with additional data from magnetometric sensors.

2.2 Noise Indicators

Noise pollution levels can be specified using several different rating methods, provided they comply with international standards and the guidelines defined by the European Commission through its directives and studies. For the preparation of noise maps, however, the *equivalent continuous sound pressure level* L_{eq} has to be used. This indicator is defined as:

$$L_{eq,T} = 10\log_{10}\left(\frac{1}{T}\int_0^T \frac{p(t)^2}{p_0^2}dt\right), \qquad (1)$$

where $p(t)$ represents the rms (root mean square) instantaneous sound pressure produced by an acoustic wave, and p_0 is a standard reference value corresponding to the minimal (human-) audible acoustic signal (i.e., $20\mu Pa$). The period T, over which the L_{eq} indicator is computed, may vary depending on the specific noise source or area of interest and may last from few seconds to weeks or years. The L_{eq} indicator, measured in decibel (dB), captures the sound level of a constant noise source over the time interval T that has the same acoustic energy as the actual varying sound over the same interval.

The equivalent sound level pressure L_{eq} defined above drives the computation of those specific noise indicators that are used for the preparation of noise maps. Indeed, European member states must provide noise pollution data in terms of the L_{night} and L_{den} indicators, which represent the equivalent sound levels averaged over the night only and over the whole day, respectively [2] [1].

The computation of noise indicators is actually more complex as it may appear from the definition of the L_{eq} indicator in equation 1. For instance, the raw acoustic signal $p(t)$ typically needs to go through a filtering stage that simulates the frequency response of the human hear (A-weighting). This filtering can be easily delegated to dedicated hardware or standard software packages and represents an optimization that can be added at a later prototyping stage. Therefore, we neglected this and other similar signal processing steps in order to quickly get a working prototype and first experimental results to investigate on. Further fundamental issues, like the spatial distribution of the measurement points, should however always be carefully considered when measuring sound pressure levels, as detailed in several standards and documents regulating the assessment of environmental noise [1, 8].

[2] Accurate definition of these indicators is provided in ISO 1996-2:2007, which recently replaced the withdrawn ISO 1996-2:1987. The standard recommends to consider the *day* period to last 12 hours, starting at 6:00 a.m., the *evening* to be 4 hours, starting at 6:00 p.m., and the *night* to extend for 8 hours, starting at 10:00 p.m..

3 Prototyping Platform

To understand the feasibility of wireless sensor nodes to be used as noise pollution sensing devices, we tested and evaluated three different hardware platforms. At a preliminary stage we considered using the Tmote Sky platform from Moteiv [13] equipped with the SBT80 multi-modality sensor board available from EasySen [7]. As reported in [15], however, we rapidly abandoned this platform due to its highly unsatisfactory performances. We then moved on examining the Tmote invent prototyping platform, also from Moteiv, which provides an extended sensor suite including an omnidirectional electret microphone that we used to measure environmental noise. To this scope, we implemented the *Ennowa* (Environmental Noise Watcher) application, which collects raw acoustic samples, computes the correspondent equivalent noise levels using a remotely settable time period T, and reports the computed values to a central sink at regular time intervals. *Ennowa* runs on top of Boomerang, Moteiv's proprietary distribution of the tinyOS operating system. The centrally collected noise samples are then stored in a database and can be further processed and visualized on common map-based web-interfaces like Google Maps [14]. While the use of the Tmote invent platform allowed for collection of first data sets to investigate on, it also made us come across the computational limits of this resource-poor sensor node. Indeed, computation of noise indicators requires the Tmote invent to sample acoustic signals at rates as high as 32 kHz. This high processing load causes rapid exhaustion of nodes batteries and may be hard to sustain if the CPU must concurrently filter the gathered samples or even send and receive messages over the radio.

To overcome the drawback represented by high sampling rates, we decided to test a third prototyping platform that could outsource the computation of noise indicators to dedicated hardware, thus dramatically reducing the computational load bearing on sensor nodes. To this scope, we built a simple, customized noise level meter and interfaced it with the Tmote Sky platform. The developed circuitry exhibits a nominal error of ± 2 dB and includes some of the signal processing stages mentioned at the end of section 2.2, like band pass filtering and frequency weighting. Measurements obtained using this instrument have been reported in more detail in a related publication [10], while in this paper we focus on the experiments performed using the Tmote invent platform. For both platforms, however, microphones' calibration is a yet-to-solve issue, which partially prevents a direct comparison of data collected by different nodes, as we will show in section 5.

4 tinyLAB: A Deployment Aid Tool

Prototyping wireless sensor network applications often requires to visualize and analyze collected sensor data to identify unexpected behavior or malfunctioning of the nodes as quickly as possible. Even if there exists tools that have been developed to serve this scope (e.g., [5]), they often do not provide satisfactory data processing and visualization features. The Matlab computing environment, on the contrary, has been developed to serve scientists in managing, processing and visualizing their data and appears therefore particularly well-suited to be used in the context of wireless sensor networks. Indeed, the tinyOS1.x software suite allows to use the Matlab environment in conjunction with the tinyOS Javatools, thereby providing basic primitives to interact with a sensor network. However, this solutions requires binding Matlab code to the tinyOS tree and thus limits flexibility and portability.

To enable Matlab-based remote control and interaction with a wireless sensor network, we developed tinyLAB, a simple framework that allows to receive and send messages from and to a sensor network and to visualize and process data as it comes from the network. tinyLAB is implemented relying solely on the Matlab software suite and offers a simple API to receive and send data from and to a tinyOS-based wireless sensor network. The structures of messages tinyLAB can exchange with the network must be entered in an appropriate Matlab file, resembling a simple nesC header file. Furthermore, an appropriate communication channel must be specified. tinyLAB currently supports receiving and sending messages from and to a serial port and/or from a TCP/IP server like the well-known SerialForwarder. Avoiding any cumbersome installation procedure, tinyLAB enables using the full Matlab computing power to manage incoming messages, process, store and visualize data, as well as to send commands to specific nodes or the whole network.

5 Experimental Results

To investigate the suitability of the Tmote invent prototyping platform to be used as noise pollution sensor we performed extensive noise measurements sessions in both indoor and outdoor settings. Due to space constraints, we focus here on two specific experiments that illustrate nodes' responses to both synthetic and real acoustic stimuli. Furthermore, we report about a case of nodes' malfunctioning we happened to come across. Experiments have been performed by letting the *Ennowa* application collect raw acoustic samples at a 8 kHz rate and compute the correspondent equivalent noise level (defined by equation 1) with a temporal granularity T of one second. The experimental settings comprised up to eigth sensor nodes deployed in the field, which used a star topology for communication. Deployed nodes regularly reported noise levels readings to the sink node, which had been in turn physically attached to a powerful computing device running tinyLAB. A simple Matlab script, written using the tinyLAB API, received nodes' messages from the sink node, timestamped, processed and visualized collected noise data (as well as other data of interest) in real-time, and finally stored all reusable information in Matlab-friendly format.

5.1 Calibration

To observe the behavior of different nodes in response to the same acoustic stimuli, we deployed all the 8 Tmote invent platforms we dispose of at close distances from each other. We then used the freely available Audacity tool [3] to produce a chain of five seconds wide white noise pulses of increasing amplitudes. Figure 1 shows the responses to these acoustic events of four different nodes, clearly pointing out a misalignment in the measured equivalent noise levels. This discrepancy is mainly due to mismatches in microphones' sensitivities[3], frequency responses and positions. Adequate standard calibration procedures using pistonphones or anechoic chambers would definitely help in limiting the misalignment in nodes' responses. This would be a necessary precondition to enable comparison of data gathered by different sensor nodes, as actually required for the preparation of noise maps.

5.2 Road Traffic Noise

To demonstrate the performances of the Tmote invent platform in an outdoor setting, we deployed the sensor nodes close to a urban road (about 3 meters distance) and recorded nodes responses. Figure 2 shows a segment of the collected data with the typical rises of the equivalent noise level values caused by vehicles transits. The rises are respectively labeled with the actual type of vehicle passing by, which we manually annotated during the experiment. This data shows that the high noise rise produced by a bus transit extends over a longer period of time if compared to that produced by a car. This characteristic, along with additional information like magnetometric data, could be exploited to design a detector able to count total vehicles transits and possibly differentiate between different vehicles categories [2]. As mentioned in section 2 current vehicle

[3]Microphones' sensitivities may deviate from the nominal value due to flaws in the manufacturing process, experienced mechanical shocks or temperature gradients.

counting procedures are expensive and inefficient, while relying on wireless sensor networks could significantly reduce costs and increase data accuracy and availability. Please note that since tram tracks are about 200 meters away from the measurement point, the produced noise levels do not reach absolute values as high as those of buses or cars, which are as close as few meters. Information about the approximate distance of noise sources from the actual assessment point is an important information that should always be reported when collecting noise pollution data.

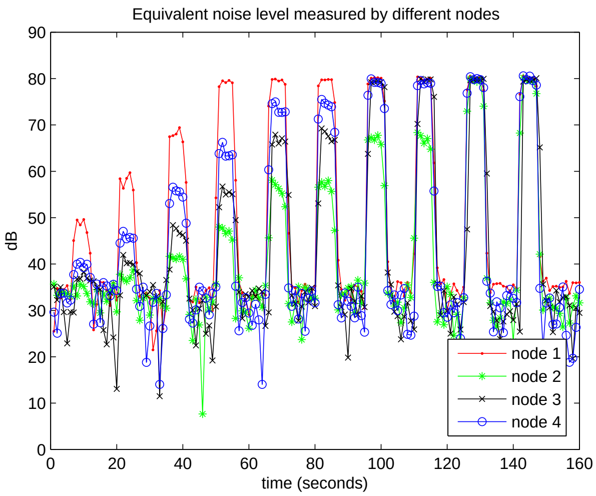

Figure 1. Acoustic responses of four different nodes to a chain of white noise pulses of increasing amplitude.

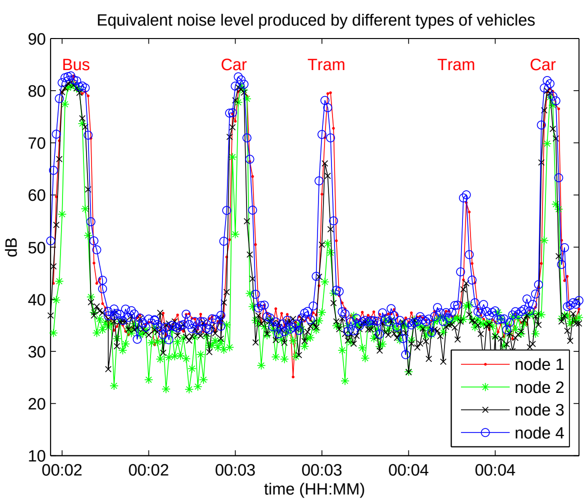

Figure 2. Acoustic responses of four different nodes in correspondence of vehicles transits.

5.3 Reference Voltage

While inspecting collected sensor data using the tinyLAB tool described in section 4, we observed an unexpected behavior of the nodes' reference voltage. In particular, we noticed that under a constant acoustic stimulus the average output voltage of the microphone assumes different values depending on the node being plugged into a power outlet or being draining current from its own batteries. Since we observed a perfectly analogous behavior for all the 8 Tmote invent platforms we dispose of, we report results related to a single, representative, sensor node. Figure 3 helps illustrating the above mentioned malfunctioning by reporting in subplot (a) the development of the total number of samples collected during a single sampling interval, in subplot (b) the average output voltage of the microphone and in subplot (c) the computed equivalent noise level. We annotated different sectors of the plot with letters from *a* to *h*, to identify different phases of our experiment. As observation begins, the node is attached to a power outlet through an adequate usb adapter and the average output voltage of the microphone is 0.8 volts (sector *a*). Once the node is detached from the adapter, this voltage level increases up to 1.1 volts, as shown in sector *b* of figure 3(b) and regularly returns to 0.8 volts if the node is plugged in again (sector *c*). Surprisingly, the effect of plugging/unplugging the node is also visible on the number of samples collected during the interval T, as shown in subplot 3(a). Indeed, as long as the node is plugged in the power outlet it collects about 8200 samples, while this figure increases up to 8600 samples once the node runs on batteries[4]. This oscillation do not (appear to) significantly influence the computed equivalent noise levels, but clearly indicate a malfunctioning in the circuitry regulating the power supply. Things become even more interesting if, instead of unplugging the node from the usb adapter, both the node and the adapter are detached from the power outlet. In this case, the average output level of the microphone shrinks to a very small value hindering proper computation of noise levels, as shown in subplots 3(b) and 3(c) (sector *d*). Instead, the number of samples keeps oscillating as observed above (see subplot 3(a)). Plugging in the node does not help in repairing the malfunctioning microphone (sector *e*) and only a node reboot restores the initial node behavior (sector *f*). Sectors *g* and *h* of figure 3 finally show the reproducibility of the above described behavior in the case the sensor node is plugged in the usb adapter (sector *g*) or not (sector *h*).

We would like to point out that the tinyLAB tool allowed us to identify and easily analyze this unexpected behavior of the nodes' reference voltage. Before deploying the network in an outdoor environment, we indeed tested the hardware and software in our lab to both understand signal dynamics and investigate the issue of calibration. The comfortable and powerful visualization and processing features offered by the Matlab computing environment, made available by the tinyLAB tool, allowed for a fast and effective real-time analysis of the data reported by the sensor nodes. Analyzing the behavior of the Tmote invent platform using a typical approach in which data is collected, stored and analyzed at a later stage, would have surely considerably delayed our prototyping process.

6 Conclusions

Focusing on the assessment of environmental noise pollution in urban areas we provided qualitative considerations and experimental results to understand the feasibility of wireless sensor networks to be used in this context. We reported our experiences in using the Tmote invent prototyping platform for collecting noise pollution data in both indoor and outdoor settings, and pointed out the potentials and limits of our prototype. While our results show the general suitability of wireless sensor nodes to be used as noise pollution sensors, they also illustrate the practical limits of today's commercially available platforms. In particular, we showed that uncalibrated nodes' microphones produce misaligned acoustic responses, hindering a direct comparison of noise readings collected by different nodes. Finally, we demonstrated that tinyLAB, our Matlab-based software suite, enables real-time data collection, processing

[4] Since the interval T extends for one binary second (1024 binary milliseconds), a sampling rate of 8 kHz results in 8192 samples/second.

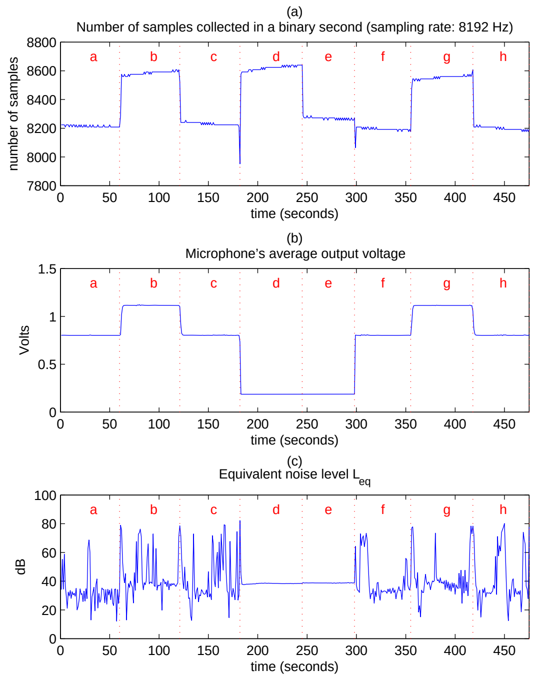

Figure 3. Tmote invent's behavior with different power supplies.

and visualization of wireless sensor network data, thereby revealing as a powerful tool for supporting first prototyping and deployments steps.

7 References

[1] Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 relating to the Assessment and Management of Environmental Noise. Official Journal of the European Communities, July 2002.

[2] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking. *Computer Networks (Military Communications Systems and Technologies)*, 46(5):605–634, December 2004. Military Communications Systems and Technologies.

[3] Audacity. Free, Cross-Platform Sound Editor and Recorder. `http://audacity.sourceforge.net`.

[4] D. A. Bies and C. H. Hansen. *Engineering Noise Control: Theory and Practice.* Spon Press (Taylor &Francis Group), London and New York, 3rd edition, 2003.

[5] P. Buonadonna, D. Gay, J. M. Hellerstein, W. Hong, and S. Madden. TASK: Sensor Network in a Box. In *Proceedings of the 2nd IEEE European Workshop on Wireless Sensor Networks and Applications (EWSN'05)*, Istanbul, Turkey, February 2005.

[6] Department for Health and Environment of the City of Munich (Germany). Noise maps 2007. `http://tinyurl.com/2gg7dl`.

[7] EasySen LLC. `www.easysen.com`.

[8] European Commission. Green Paper on Future Noise Policy. Com (96) 540 final, November 1996.

[9] European Commission Working Group Assessment of Exposure to Noise (WG-AEN). Good Practice Guide for Strategic Noise Mapping and the Production of Associated Data on Noise Exposure, January 2006.

[10] L. Filipponi, S. Santini, and A. Vitaletti. Data Collection in Wireless Sensor Networks for Noise Pollution Monitoring. In *Submitted for publication to the 4th IEEE Intl. Conference on Distributed Computing in Sensor Systems (DCOSS '08)*, Santorini Island, Greece, June 11–14 2008.

[11] B. Greenstein, C. Mar, A. Pesterev, S. Farshchi, E. Kohler, J. Judy, and D. Estrin. Capturing High-Frequency Phenomena Using a Bandwidth-Limited Sensor Network. In *Proceedings of the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys'06)*, Boulder, Colorado, USA, November 1–3 2006.

[12] London Noise Mapping. The London Road Traffic Noise Map. `www.londonnoisemap.com`.

[13] Moteiv Corporation (now Sentilla). `www.sentilla.com`.

[14] D. Rauch. Towards the Sensor Web: A Framework for Acquistion, Storage and Visualization of Wireless Sensor Networks Data. Master's thesis, Distributed Systems Group, Department of Computer Science, ETH Zurich, Zurich, Switzerland, February 2008.

[15] S. Santini and A. Vitaletti. Wireless Sensor Networks for Environmental Noise Monitoring. In *6. GI/ITG Workshop on Sensor Networks*, Aachen, Germany, 2007.

[16] Sensor Network in a Vineyard. GoodFood European Integrated Project: Food Safety and Quality Monitoring with Microsystems. `http://www3.unifi.it/midra/goodfood/`.

[17] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An Analysis of a Large Scale Habitat Monitoring Application. In *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, Baltimore (MD), USA, November 3-5 2004.