Monitoring of Civil Infrastructures using Wireless Sensor Networks

REINHARD BISCHOFF, JONAS MEYER, GLAUCO FELTRIN, OLGA SAUKH

ABSTRACT:

This paper presents a prototype monitoring system for civil structures based on a wireless sensor network. The sensor network is composed of a root node, representing the data sink, logging and configuration unit and many sensor nodes (motes) distributed over a structure. Each mote is equipped with sensors, a digital signal processing unit, a radio transceiver and a power supply. All motes together form a multi-hop communication network. That is, each mote figures as a data source and a relaying station, forwarding data received from adjacent motes. The routing of the data packets is optimized for maximizing the network's lifetime. Another feature of the network is its capability of self-organization. If a single mote or parts of the network fail, new paths to the root node are discovered and established.

Low power consumption is a key issue in wireless sensor networks because the motes have to operate from batteries for a reasonably long time. Since data communication dissipates remarkable more energy than data processing, it is essential to execute suitable data reduction algorithms allowing for a significant reduction of data items which have to be transmitted to the root node.

The base station of the sensor network is accessible over the internet using wired or wireless connections. A locally or remotely connected terminal visualizes the data and performs a prediagnostic for identifying abnormal states. Moreover it allows for administration of the nodes and configuration of measurement setup. Additionally the collected data are stored in a database.

This paper presents laboratory tests performed with the prototype monitoring system on the scaled cable stayed pedestrian bridge at Empa. The prototype system monitors the stay cable forces. The force is calculated via natural frequency estimation of cable accelerations. This computation is executed on the motes. Only the cable forces are transmitted to the root node.

Reinhard Bischoff, Jonas Meyer, Glauco Feltrin, Structural Engineering Research Laboratory, Swiss Federal Laboratories for Materials Testing and Research, Ueberlandstrasse 129, 8600 Duebendorf, Switzerland, reinhard[dot]bischoff[at]empa[dot]ch

Olga Saukh, University of Stuttgart, Universitaetsstraße 38, 70569 Stuttgart, Germany

INTRODUCTION

Traditional monitoring systems show a star like topology with a data logger unit as center. Various sensors deployed over a structure are connected via long cable runs to that logging unit. These systems are very time consuming to install and therefore cost-intensive. Furthermore, the long cable runs are sensitive to electromagnetic interferences with the effect of reducing the accuracy of the acquired measurements and claiming for expensive high quality cables. Moreover, these cables are susceptible to mechanical damage involving considerable maintenance effort.

The monitoring system presented in this paper aims to solve this problems by introducing wireless data communication techniques. In order to omit the cables, the connection between the sensors and the logging unit are substituted by wireless links. Therefore, each sensor is equipped with a radio device enabling a wireless communication of the measurement data.

Overall structure of the monitoring system

The present monitoring system is mainly composed of three subsystems. Figure 1 gives an overview. The first subsystem is a wireless sensor network (WSN) installed on the structure. A WSN is a network made up of many tiny intercommunicating computers [1]. Each tiny computer represents a node of the network. These nodes are called sensor nodes or motes. The motes are self-contained units typically consisting of a power supply with limited capacity, a radio transceiver, a micro controller and one or more sensors. Each mote measures physical parameters of the structure. These data items are evaluated later on to determine the actual condition of the structure. The second subsystem makes the information originating from the sensor network available to a user or an operator. It implements the data visualization and condition representation. Furthermore, this subsystem is responsible for the long term data storage. The third subsystem forms the link between the sensor network on the structure and the control center. This link is particularly important if the control center is not located on site. It enables the operator to observe, control and configure the sensor network remotely. This link can be established using standard communication technologies like Internet or UMTS and will not be described in more detail.

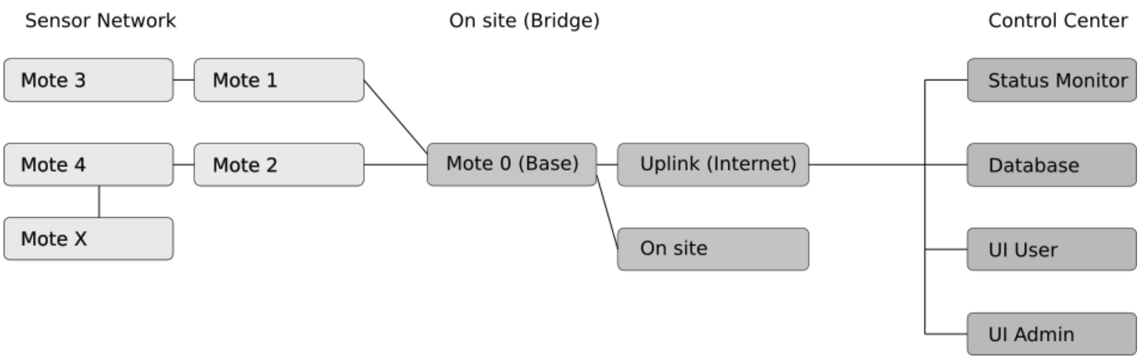

Figure 1: Overview of the present monitoring system on the laboratory bridge at Empa showing the wireless sensor network (left), the control components (right) and the link in between.

Energy constraints

The elimination of the cables connecting the sensors to the central logging unit solves the problems mentioned above but new challenges concerning the power supply of the sensors and communication devices arise. In fact they have to be powered by an autonomous source like batteries or solar cells which are severely limited in capacity. The limited energy resources on each mote present the most restricting factor in designing and implementing such types of monitoring systems. Therefore, monitoring and measurement as well as communication strategies have to be introduced taking into account the limited energy resources.

One option to save energy is to reduce the acquired raw data before transmitting it over the wireless link. This is often feasible since the raw data contains a lot of redundant or even irrelevant information which can be discarded. It is possible to extract a small number of characteristics describing the physical process quite well. Additional energy can be conserved considering the fact that the power needed to transmit data over a distance d is proportional to d^{α}, α ranging from 2 outdoors to approximately 4 indoors. Hence, in order to minimize power consumption, it is preferable to send the data over a chain of several short hops to the target instead of transmitting it directly over the entire distance. Therefore, it is advantageous if each mote acts as data source as well as relaying station, forwarding data of other motes. This way, all motes together form a multi-hop network. The routing of the data is organized in a way to maximize network lifetime, i.e. energy consumption is distributed among the nodes.

THE MOTES

Mote hardware

To provide the defined functionality, a mote has to be composed of one or more sensors, a signal conditioning unit, an analog to digital converter, a processing unit with memory, a radio transceiver and a power supply.

Various mote hardware platforms are available which are optimized in terms of power consumption. The prototype network used at Empa is based on the Tmote Sky platform [2]. It features a 6 channel ADC with a resolution of 12 bit, a 16 bit processor with 10 kB RAM and 48 kB program flash memory, and a radio transceiver operating in the 2.4 GHz ISM band with a raw data rate of 250 kbps. A signal conditioning unit enables to interface various sensing elements like temperature and humidity sensors, strain gages and capacitive and piezo-resistive accelerometers.

Mote software

A wireless sensor network is essentially a distributed computing system. The software running on each mote establishes the network, organizes the communication among the motes, acquires the measurements, performs the data processing and analysis and generates alerts when particular conditions are met. Since each specific monitoring application differs among each other, an open, flexible and scalable software architecture is required.

The present prototype allows collecting information about the structural condition based on temperature, humidity, strain and acceleration measurements. Moreover, it is possible to receive information about the internal state of each mote as well as the communication parameters of the sensor network describing the current condition of the monitoring system.

The energy constraints apply also to the software components. Therefore, all modules are implemented using energy-aware approaches to maximize the lifetime of the monitoring system.

The native operating system of the Tmote Sky platform is TinyOS [3], a component-based operating system designed for sensor networks and tailored to fit the memory constraints of the motes. TinyOS provides a concurrency model and mechanisms for structuring, naming and linking software components to form a robust network embedded system.

Configuration

Each mote in the network can be equipped with different sets of sensors and, therefore, the software that is loaded on each mote needs to be configured appropriately. The parameters which characterize the mote configuration can be static or dynamic depending if they have to be set at compile time or if they can be modified at runtime. The present prototype allows the user to dynamically configure various parameters like sampling rates, query rates, number of values sent per data packet etc. Furthermore, it is possible to dynamically change the conditions for the triggering of alarm messages.

Data Model

The main characteristic of wireless sensor networks is that they are data-centric and, therefore, the data itself is of great importance. All the data received from the sensors is organized by attributes which are stored in the attributes pool on the mote and can be queried. If several queries have to be answered at the same time, they are merged together. There exist two types of attributes:

- Simple Attributes. These can be raw readings from sensors such as temperature, humidity or strain values which do not need to be acquired with high sampling rates. This group also includes the information describing the internal state of soft- and hardware as well as statistical information about the network configuration, e.g. number and IDs of neighbor motes, battery voltage, etc.
- Complex Attributes. These are physical quantities which need to be sampled with high sampling rates and are predominately used if dynamic processes like vibrations have to be captured.

The core of the mote software is a scheduler. It controls the intervals at which attributes are queried and knows how to configure the attribute correctly. The scheduler is time synchronized, more details can be found below in section Synchronization.

Routing

The routing protocol is responsible for reliable and efficient packet delivery. This prototype uses multi-hop communication which forces several intermediate motes between source and destination to participate in data forwarding. Since radio communication is the most energy consuming task, the efficiency of the routing module influences the lifetime of the whole sensor network considerably. Additionally, there is a trade-off between energy consumption and transport reliability, because if a packet is lost because of a too weak radio signal, it requires additional energy for its retransmission. A promising routing algorithm has been developed accounting for link quality and energy consumption [4]. Since monitoring applications on civil structures typically do not require any mobility of the motes, the routing algorithm has been optimized for static networks. Moreover, it allows using different policies to influence the routing tree structure using application-specific information.

Periodic query answers and event messages are sent using many-to-one routing. The routing module distinguishes the messages to be transmitted depending on their contents and provides more reliable message delivery for events (alarms) by applying more retransmission possibilities in case of a send failure. The one-to-many information flow is required to send commands to multiple motes in the network. Commands manage and modify parameters on a running mote by changing the sampling rates of the attributes, switching them on/off or defining conditions for events.

Synchronization

Time synchronization is of utter importance in various monitoring applications. Furthermore, communication, coordination and power management also depend on the existence of global time. Due to this, the motes are kept synchronized using a Flooding Time Synchronization Protocol [5]. This protocol uses low bandwidth and is robust against node and link failures. The synchronization error per hop is below 0.1 ms.

Data Processing

Signals which need to be sampled at high sampling rates produce a huge amount of data. For such complex attributes, the in-mote preprocessing is essential to reduce the number of data items which have to be transmitted. Often it is possible to extract a few parameters out of the raw measurement data stream which characterize the physical process adequately enough. This approach preserves considerable amounts of energy. Currently, the system performs in-mote data analysis for acceleration sensor readings.

DATA MANAGEMENT AND ACCESS TOOLS

The data obtained from the sensor network has to be accessible in a suitable way for system users or operators. The data items acquired from the sensor network have to be stored appropriately and users have to be able to access the stored data to perform the needed analysis and to get alarmed if necessary. Various software components provide this accessibility. An operating system independent implementation of the software ensures flexibility and portability.

Database

For the long term storage of the collected data and subsequent analysis, a database is used. The applications access the database through Hibernate, a Java-based object-relational mapping and a persistence framework. Using Hibernate provides the flexibility to attach different kind of database systems without changing the application code. Even simple ASCII text files are supported which have no need to install additional database software.

All events and attribute values received from the sensor network are logged to the database. Furthermore, the data sent to the motes is also stored in the database, i.e. commands which contain network configuration data. Therefore, it is always possible to retrace a specific event. The structure of the database is designed to store several WSNs in parallel. This gives the opportunity to store data from several locations into one database and simplify the administration and analysis.

Daemons

Daemons are programs running continuously in the background on the server. In the present prototype implementation, the daemons run on the computer that is connected to the network base station and manages the collection of data packets that arrive at the base station. The daemons store the data packets in the database and handle the requests coming from the network. Another task that is to be implemented is the triggering of alarm messages. For example, a daemon could send an email to an operator if it handles a data packet containing information indicating an exceptional condition.

Configuration tools

The network itself and the motes can be configured and administrated remotely. This allows the operator of the monitoring system to remotely enable or disable the different measurements and change the attribute characteristics on a specific mote, calibrate various mote parameters, set up the detection of exceptional conditions and reset the mote. Furthermore, the motes can be reprogrammed remotely. This mechanism allows for a newly developed analysis algorithm or a completely new firmware to be easily installed on the motes.

User Interfaces

The user interface integrates various highly modularized components. It enables the user to access the database and interact with the wireless sensor network. The basic access components are command line tools. These modules provide the low level access for the user. This level is the most suitable if the tools are used in highly automated monitoring setups and for system administration purposes. The graphical user interface is located one level above the command line tools. It is basically a framework that integrates the graphical interfaces of the various command line tools. A third way to examine the structural behavior is to access the data using a web browser. This interface offers the same functionality like the other ones. The user interface components are mostly written in Java what enables a high interoperability on different platforms and operating systems.

FORCE MONITORING OF CABLE STAYS

Background

A potential application of wireless sensor networks is cable tension force monitoring of stay cable bridges based on vibration measurements and natural frequency estimation. Advanced identification methods include the effect of bending stiffness by simple approximation formulas of the natural frequencies. The natural frequencies are usually estimated by using frequency spectra or output only system identification algorithms. However, these methods are too expensive in terms of memory usage to be applicable to wireless sensor networks. In the present monitoring application, a simple autoregressive model which requires significantly less memory has been used to estimate natural frequencies. The method has been successfully implemented and tested on the cable stays of the laboratory bridge at Empa. A detailed description of the setup and the implemented algorithm can be found in [6].

Algorithm for estimating natural frequencies

Parametric methods of spectral analysis allow estimating natural frequencies with much less data than averaged spectrogram methods. Furthermore, the natural frequencies can be estimated by computing the poles of the associated rational function models and are therefore not subjected to a given frequency resolution. The natural frequencies have been computed using a very simple 2 parameter discrete time AR-model. The use of such a simple model is only possible if the vibration components associated with a natural frequency can be isolated. If the natural frequencies are well separated, the components can be isolated by first filtering the recorded data with a band pass filter.

Laboratory tests and results

The motes with the accelerometers were mounted on six cable stays. The bridge deck was excited with an electro-magnetic shaker driven by a broad band, stochastic signal ranging from 1 to 80 Hz. The accelerations of the cables have been simultaneously recorded with the motes and with a high precision data recorder. The natural frequency of each cable was estimated with a data block of 400 samples. Each data block was subdivided into 10 segments of 40 samples (0.4 seconds). With a clock speed of 4MHz, it took the microprocessor about 8s to complete the computation on a data block of 400 items. Figure 2 shows one natural frequency in a plot window of the graphical user interface. The changes of natural frequency are due to loading and unloading the bridge deck with additional mass. The natural frequency estimations show significant scattering. This scattering is due to the shortness of the data segments (0.4 seconds). The averages of the natural frequency of the three loading states are 8.51, 9.08 and 8.80 Hz. These values fit reasonably well with the results computed using the data recorded with a high precision data acquisition device: 8.47, 9.18 and 8.88 Hz. The latter figures were obtained by evaluating a 10 minutes record what corresponds to an accuracy of approximately 1%.

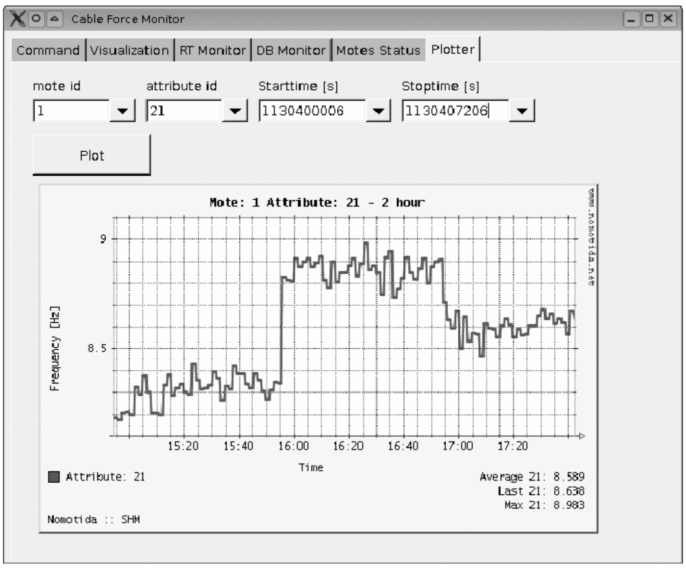

Figure 2: Screenshot of the graphical user interface. The graph window shows a plot of the estimated natural frequency of one cable stay. The visible steps in the plot correspond to three different loading states of the bridge deck.

CONCLUSIONS AND FURTHER WORK

The presented prototype monitoring system has been successfully implemented and tested on the scaled laboratory bridge at Empa. The prototype shows that monitoring based on wireless sensor networks is feasible and that appropriate algorithms and strategies can be implemented which fit the limited memory and computation resources of the motes and provide reasonably accurate results. Additionally, various software tools have been presented which allow the monitoring system operator to access the aggregated data in a suitable way.

In the next period the wireless sensor network prototype will be further tested in the laboratory as well as in the field to analyze the robustness of the wireless sensor network software and hardware as well as measurement, analysis algorithms and methodologies.

ACKNOWLEDGEMENTS

Part of this work was performed within the FP6 Integrated Project “Sustainable Bridges: Assessment for Future Traffic Demands and Longer Lives”. The authors acknowledge the European Commission and the Swiss Federal Office for Education and Science for their financial support.

REFERENCES

1. Culler, D., Estrin, D., Srivastava, M. 2004. Overview of Sensor Networks. *IEEE Computer, Special Issue in Sensor Networks.*
2. Polastre, J., Szewczyk, R., Culler, D. 2005. “Telos: Enabling ultra-low power research,” *Information Processing in Sensor Networks/SPOTS*, Berkeley, April 2005.
3. Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., and Culler, D. 2005 “TinyOS: An operating system for wireless sensor networks,” *Ambient Intelligence*, New York, Springer-Verlag
4. Saukh, O., Marron, P. J., Lachenmann, A., Gauger, M., Minder, D., Rothermel, K. 2006. “Generic Routing Metric and Policies for WSNs,” *Proceedings of the European Workshop on Sensor Networks, EWSN*, Zurich, Switzerland, 13-15 February 2006.
5. Maróti, M., Kusy, B., Simon, G., and Lédeczi, Á. 2004. “The flooding time synchronization protocol” *In Proceedings of the 2nd international Conference on Embedded Networked Sensor Systems* (Baltimore, MD, USA, November 03 - 05, 2004). SenSys '04. ACM Press, New York, NY, 39-49.
6. Feltrin, G., Meyer, J., Bischoff, R., Saukh, O. 2006. “A Wireless Sensor Network for Force Monitoring of Cable Stays,” *Third International Conference on Bridge Maintenance, Safety and Management, IABMAS 06*, Porto (Portugal), 16-19 July 2006.