Wireless CiNet Network Analysis and Diagnostics Using Neighbourtables

Ismo Hakala and Timo Hongell
University of Jyväskylä / Kokkola University Consortium Chydenius
P.O.Box 567, FI-67701, Kokkola, Finland
Email: {ismo.hakala, timo.hongell}@chydenius.fi

***Abstract*—Reliable communication is crucial for successful deployment of a wireless sensor network. By taking into account communication and other wireless sensor networks constraints, networks can be analyzed efficiently. The diagnosis of the network's operation can be done using the information gathered by the sensor nodes in the network. This paper discusses wireless sensor network diagnostics describes so-called neighbourtables used to collect diagnostic data of the wireless CiNet network and their construction. The information stored in neighbourtables can be used to monitor network behavior and to improve networks' packet routing and fault recognizing for the nodes, in both, nodes' and centralized applications' point of view.**

***Keywords*-neighbourtables, wireless sensor network, diagnostic, management**

I. Introduction

In recent years, study of wireless sensor networks (WSN) has become a rapidly developing research area. A WSN is a set of wireless sensor nodes where each node measures a physical value using selected sensor probes and sends the value to a database through specific sink nodes. Nowadays, WSNs are widely used in civil and industrial applications such as smart home or environment monitoring [1], [2], [3]. Compared to traditional sensing methods, wireless sensor networks technology offers some benefits: wide areas can be covered with inexpensive, energy-efficient battery-powered devices, which make long-term monitoring and real time access to measuring data possible. Often the nodes of WSN also are able to self-configure themselves, which enables quick and easy system deployment.

The use of WSN-applications also reveals many different constraints that can decrease the possible number of real application deployments. These constraints may be defined in different categories based on the constraints, such as the system's memory, processors' limitations and energy consumption. Out-of-date communication equipment and their bandwidth as well as physical environmental and measurement factors related to sensors' location and calibration also may be the cause of different constraints. A WSN can take into account some of the possible constraints by gathering diagnostic information from the network. For example, radio link quality is affected by many internal and external factors, and it can be evaluated by using the Received Signal Strength Indicator (RSSI). RSSI values as well as other diagnostic data, such as battery levels, the number of received packets etc., can be stored in one table, the so-called neighbourtable.

In WSN nodes, the diagnostic data collected to the node's neighbourtables, gives the nodes direct information about the network around them and also information about the sink node. The node can directly use the neighbourtable information to determine its routing and clustering possibilities based on the collected and calculated information about the neighbour nodes. Since the nodes' neighbourtables can be centrally collected to a server database, they can be utilized in different diagnostic applications. For example, we have developed a graphical real time application, CiNetView, to make wireless sensor network deployment and monitoring easier. The application visualizes, in real time, the nodes' relative locations as well as shows the links' quality, which make the deployment of WSN much quicker and easier.

This paper discusses reasons for wireless sensor network diagnostics and describes the so-called neighbourtables used in the wireless CiNet sensor network. The paper is organized as follows: First, we provide a brief description of some related research and then discuss the useful diagnostic data and neighbourtables in WSN. Section IV presents the neighbourtables construction in CiNet network as well as their usage. Other functionalities of the tables are also described in more detail. A survey of the neighbourtables' energy cost is presented. Finally, some experiences of the use of neighbourtables in real wireless sensor network implementations are discussed.

II. Related work

Analysis and monitoring of WSNs are highly evolving research topics in the field of wireless technology. However, real time performance of wireless communication is not that widely studied. In [4], Meier et al. discuss link behavior and metrics that can be used to evaluate link performance. They have used statistical link analysis in their studies. Ferrari et al. have done indoor performance

 ISBN: 978-1-61208-144-1

studies of WSNs [5]. Sensor network diagnostics and visualization are discussed in [3]. However, there has not been much discussion on real time inspections of the changes in link qualities, whether uplink or downlink, in sensor networks. There are some commercial network visualization and diagnostic applications, for example, MOTE-VIEW [6] and Surge View [7], developed by Crossbow Technology. One approach that uses additional messages to construct and utilize neighbourtables is considered in [8]. Routing using neighbourtables is studied in [9] and [10]. Jacuot et al. have discussed indirect diagnosis of the node state with few messages, using an SNMP-like LiveNCM management tool [11].

III. Network variables and diagnostic

For a customer's point of view, wireless network is fine when the wanted information is received and processed correctly, that is, when the network works properly. On the other hand, wireless networks include large amounts of all kinds of data that can be used to give useful information about the networks' operational performance and reliability, not only for the user or customer, but also for the networks' maintenance crew. Due to the limited amount of memory and space in the sensor nodes and the transmitted data frames, it is reasonable to define the relevant information and metrics that can be used in WSN diagnostics. Fynn [12] has done a collective study of WSN performance analysis methods and metrics, the topics including storage, routing, real time communication, power management and architecture.

The most important information from the node's point of view are its neighbours' addresses, since without them other information cannot be linked to a specific node and it is almost impossible to perform any data routing. The sink node is defined to be the so-called root of the network, and its relative location need also be known. Therefore, a metric called hop count is essential information for the nodes. The bigger the hop count the greater the number of relaying nodes that are needed for sending packets between the specific sensor node and the sink node. This value does not necessarily directly indicate the physical distance, but rather the relative length of a path that a data packet needs to be forwarded to reach the sink node. The hop count value tells the node the direction of the sink that the packet is to be sent to with the minimum number of forwarding retransmissions. Hop count values can also be defined to be calculated from the nearest gateway of the node.

Radio link quality is affected by many factors, which can be divided into a device's internal and external factors. The internal factors are caused by imperfections of the device's hardware or software. E.g., different radio chips do not behave exactly in the same way and each node has its own radiation pattern that is not uniform [2]. The external factors, such as fading, shadowing, multipath propagation, and dynamic environmental factors affect wireless communication and make it difficult to predict the radio performance beforehand. Link quality can be evaluated by using the Received Signal Strength Indicator (RSSI), which indicates the strength of the radio signal between two nodes at the receiver's position.

RSSI values can be used to determine whether the link is acceptable or not. The nodes typically have been programmed to respond to a predefined RSSI lower bound to determine whether the link is strong enough to be useful. In WSNs, radios typically operate in the 2.4GHz ISM band and are based on the IEEE 802.15.4 standard due to which RSSI value -85dBm is considered to be the acceptable lower bound. It is also possible that the link quality may only be suffering from temporary deterioration, for example when people suddenly walk between the nodes. Therefore the neighbour RSSI values also need to be averaged to avoid useless routing changes. Since the data packets' path RSSI value, from node to sink or vice versa, cannot be any larger than the worst link's case, the path RSSI values indicate the lowest RSSI value between the sink and the node. This information can directly be used in packet sending and routing decisions. Meier et al. [4] have used RSSI values too, but also, e.g., number of packets, average packet reception rate (PRR) and link quality indicator (LQI) to perform efficient WSN link diagnostic.

When sent packages are spread from the sink node to the last node in the network, it is possible that some of the packages arrive through multiple paths and are somewhat delayed. Nodes need to be able to recognize the packets that have been sent on the same synchronization period. Therefore, every packet has its own sequence number, which also needs to be stored. Some of the packets may be received from different neighbour nodes, meaning that the packets have traveled through different paths from sink to the receiver node. Packet routing is also one interesting network management related topic that will be discussed later.

Wireless nodes naturally need energy to operate, and they usually are battery powered. The varying shape and utilization of the network cause that the battery levels of the nodes do not consume at the same rate all over the network. Therefore, it is essential to know that how much the batteries have power left.

In addition to these considerations, the nodes can be programmed to have several different counters that can be set to count, for example, the number of received and missed packets. These counters can then be used to calculate, e.g., throughput and reliability values of the sensor or relay nodes. All this information needs to be stored somewhere. One solution is to use neighbourtables.

Typically, when talking about neighbourtables, people are thinking about routing tables. A short comparison of neighbourtables and routing tables is needed. A routing table is a set of rules that is used to decide where data packets traveling over network will be directed. Each packet contains information about its origin and destination. The routing table contains the information necessary to forward a packet along the best path toward its destination. Usually a routing table includes the following information:

- Destination: The address of the packet's final destination
- Next hop: The address to which the packet is to be forwarded
- Metric: Assigns a cost to each available route so that the most cost-effective path can be chosen

Our definition of neighbourtable is that neighbourtables basically include, not only the same information as routing tables, but also some additional information. They can be used to aid routing decisions, but they can also be used in different diagnostic and management solutions as well. For a single node's point of view, a neighbourtable is a multifunctional set of information about the nodes' neighbour nodes and the links between them. Globally speaking, the neighbourtable file, constructed by the server, includes all essential information about the whole network, and the whole network's operation can be diagnosed and monitored in real time from there.

IV. The CiNet Neighbourtable

We are using CiNet nodes [2][13] in our study. CiNet is a research and development platform for the WSN implemented in Kokkola University Consortium Chydenius. The hardware in the CiNet node is specially designed for WSNs and consists of inexpensive, standard off-the-shelf components. The CiNet node includes all the basic components necessary for WSNs. In our CiNet, the nodes use cross-layer architecture [13].

The main idea of the cross-layer architecture is to implement a wireless sensor network's basic tasks, such as topology management and power saving functionalities, as separate protocols in a cross-layer management entity. Data structures, which are in common use, are in this study implemented in the cross-layer management entity as a neighbourtable data addition. The use of a cross-layer implementation reduces computational and memory requirements - not all the information needs to be transmitted between application interfaces and protocol layers. The architecture also allows the implementation of the application and protocol stacks be as simple as possible, since they are practically free of the tasks related to network management.

In every node, the neighbourtable is stored to a one common data storage in the cross-layer management entity, where all the protocol stack layers can utilize the same information, see Figure 1. This reduces the total amount of memory storage space needed, but the use of cross-layer architecture causes challenges related to maintenance, which have to be considered in the implementation. The problem has been approached by using message multiplexing in the data link layer and modular structure in the cross-layer management entity.

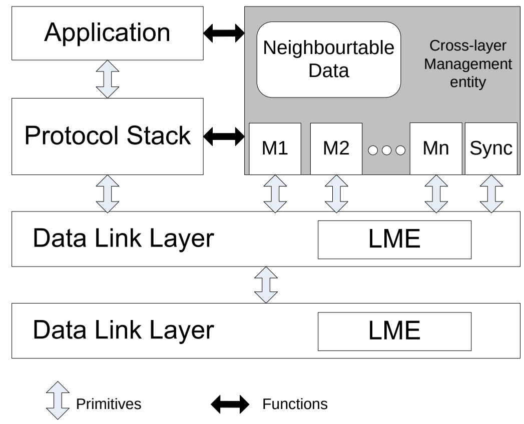


Figure 1. CiNet network's cross-layer architecture.

Basically the neighbourtable of each node consists of d levels with b entries at each level. More precisely, every level d is a different neighbour of the node and each entry b includes stored information, such as node ID, battery level, RSSI and hop count.

All the nodes' neighbourtables are also collected to a single data file on a server, from where all the information can be retrieved. This centralized neighbourtable can be used in different WSN management and diagnostic applications and tools. The format of the file is shown in Table I.

Table I
Format of the server's neighbourtable data file.

Seq No	Node ID 1	Neigh. 1 data	Neigh. 2 data	···
Seq No	Node ID 2	Neigh. 1 data	Neigh. 2 data	···
⋮	⋮	⋮	⋮	⋮
Seq No	Node ID N	Neigh. 1 data	Neigh. 2 data	···

A. Neighbourtable construction

In a CiNet network, each node constructs and maintains its own neighbourtable, as defined in Table II, in which the node stores information about its neighbours, which are the nodes that it hears. The neighbourtable of each node is updated in every synchronization period of the network, see Figure 2. The neighbourtable construction and update is defined to be one part of the synchronization protocol. The sink node broadcasts the

 ISBN: 978-1-61208-144-1

synchronization message isotropically, and every node that hears it broadcasts that message onwards through the network during a predefined time period. In this way the whole network can be synchronized. The synchronization frame structure is shown in Figure 3(a). The SYNC frame also includes additional data that is not used in the neighbourtables, such as the CMD and a command byte that can be used to control the nodes' operation. Before relaying the synchronization message, the nodes update it with their own information. Based on the received synchronization messages and the data included in the synchronization frames, the nodes update their neighbourtables. Note that the synchronization messages are heard by all the nodes' neighbours, including the predecessors, which means that the neighbourtable information can be collected in both directions. To prevent any ping-pong effect, the nodes broadcast the synchronization message only once in every synchronization period. Continuous synchronization of the network is vital to ensure valid operation of the network.

Table II
CINET NODES' NEIGHBOURTABLE

U16INT u16NbAddr	Neighbor address
S8INT s8RSSI	Neighbor link RSSI value
U8INT u4Bat:4	Battery level of the neighbor
U8INT u4HopCnt:4	Hopcount of the neighbor
U8INT u8NodeType	Sink, Relay, Sensor
U16INT u16Received	Number of received sync packets
U16INT u16Missed	Number of missed sync packets
S8INT s8AvgRSSI	Avg RSSI of the neighbour link
S8INT s8PathRSSI	Path RSSI (path's weakest RSSI)
U8INT u8PrevSeq	Previous sequence number
U8INT u8Ntp	UpLink throughput
U8INT u8UplinkTp	Path throughput
U32INT u32NbLastSeen	Last seen time, for entry maintain
U8INT u8Status	Sync status

After every synchronization period, every node in the network now has real time information about the network around, including information about the sink node. After the synchronization period, the nodes also can send the neighbourtable information through the sink node to the server during the management period. The time interval of this neighbourtable update can be defined to meet the application demands. A minimum update interval is one synchronization period, but it can also be much longer. The management frame includes a section where the neighbourtable information is sent. The management frame structure is shown in Figure 3(b) Diagnostic and other management data is sent (and acknowledged) as a unicast transmission through a selected route to the sink node. If retransmissions are not needed, each management frame is sent once in every synchronization period.

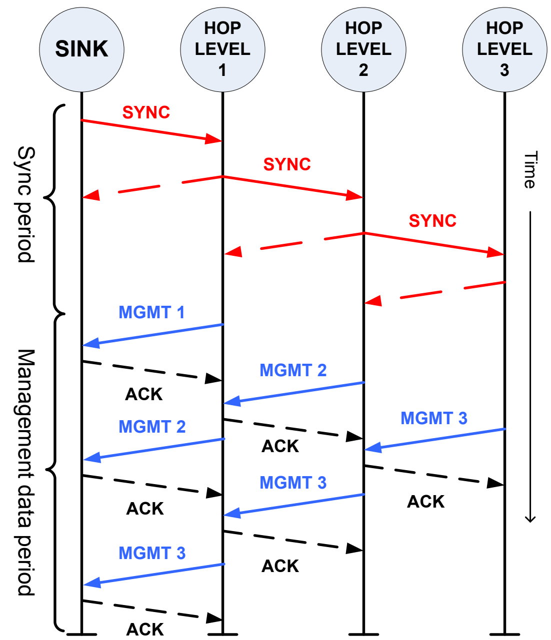
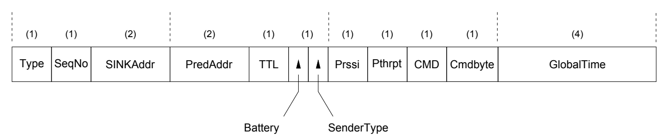
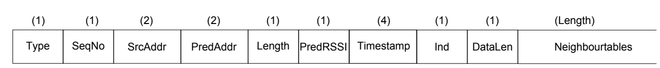


Figure 2. Synchronization, management and ACK messages during one synchronization period.



(a) SYNC frame structure



(b) MGMT frame structure

Figure 3. Synchronization and management frame structure

B. Neighbourtable utilization

Neighbourtables are specially used for collecting information for real time deployment and for monitoring of the WSN. They can be locally used in the nodes or with some management tool. Since every neighbourtable of every node is known, it is possible to count different diagnostic values using the information collected to the neighbourtables.

One of the most useful values that can be calculated from the neighbourtable information is the throughput of one link. The throughput value can directly indicate the reliability and robustness of the sensor node or a link between two nodes. A good throughput values should be near 100%, but in WSNs typically at least some of the packets are missed. Since network synchronization is done periodically, the node knows how many times it should have heard the synch message after having received the first synch message. The sequence number

 ISBN: 978-1-61208-144-1

of the received message is checked, and if the number has increased more than one, then a packet or packets have been missed. The throughput information is calculated based on the nodes' packet counter values.

When a node has been working for a while, the throughput value will settle to some level that will indicate the basic information about the links reliability. If the throughput value suddenly begins to go lower, it will be a clear indication about something having changed or broken in the network, for example, a car might have been parked between two nodes, interfering with the signal.

A node's battery level information can be used to alert the network supervisor to do network maintenance. If a low battery level is noticed before a link stops working, due to power out, it is possible to keep the network topology controlled. Another point is that if the node's battery level is decreasing more rapidly than in the other nodes, it is possible that that the node is not working correctly, or that it is relaying too much data.

For packet routing, nodes typically use basic routing tables. Using the routing table, nodes can efficiently transfer data from each node to a sink, but the routing decisions can be thoroughly justified with the extended information of the neighbourtables. Naturally, the nodes' address information is the most important, but other information can be used to optimize the networks' routing performance and reliability. Almost all the information that the neighbourtable contains can be used to improve the network's data routing.

In our solution, the routing protocol uses hop count and RSSI values. The minimum hop metric is used in many routing protocols due to its simplicity and isotonicity [14]. Hop count value can directly indicate the logically shortest transmission path from a node to a sink. RSSI values also indicate the links' or the whole paths' quality. In order to maximize packet throughput, it is reasonable to choose links that are more likely by the next relay. Throughput values indicate the total amount of packets that have successfully been transmitted through a specific link. If a link has a good throughput value, it is more likely to perform the transmission successfully again. The battery levels of the nodes' neighbours can also be used to define the data routing. If the battery level of one node is getting low, then an alternatively routing should be used, if possible, to avoid unnecessary dropouts.

Transmitting power for the nodes' radios can also be optimized using the RSSI information in the neighbourtables. In some applications and hardware solutions, it is possible to adjust the radio chips' transmitting power in real time. An acceptable lower bound of RSSI can be fixed, and, based on it, the transmitting powers can be lowered or increased when necessary.

V. The Cost of Neighbourtables

In CiNet network, the nodes are synchronized periodically to ensure valid operation. The length of the synchronization period, during which the neighbourtables are collected, depends on the application used and on the WSN measurement solution. Thus, data collection is embedded to the synchronization messages, and no extra messages are needed to be sent to collect the neighbourtable information. The maximum size of the SYNC frame is 16 bytes. Of these, 4 bytes are directly related to the neighbourtable usage. It can be stated that basically almost all information in the SYNC frame would be sent even without the neighbourtable usage, since the information is used for routing protocols in any case. Therefore, it can be said that the neighbourtables are filled with almost free of additional energy cost. Only the size of the synchronization frame is increased.

As every node can store the information of eight neighbours and as 18 bytes of memory have been reserved for each neighbour, this means a maximum memory use of 144 bytes (8 x 18 bytes). From these, the six best are sent in the management phase (one management frame can fit six neighbours).

The main additional cost of the neighbourtables incurs when the tables are also sent to the sink node as a part of the management frame. This increases the number of sent packets and time to spend for sending the data. It also consumes more energy. The total cost of one sent and received management frame is defined as $E_{frame} = \Delta E_{tx} + \Delta E_{rx}$. Our measurements have shown that a management frame transmission takes about 0.216 mJ of energy and receiving takes about 0.142 mJ, so the total consumption is about 0.4 mJ [15], [16].

The energy overhead of the whole network to transmit the neighbourtables to the sink node depends on many different factors. These include, for example, the size of the network used, the number of hops and the application that determines the number of sent and relayed packets. The transmission time of one node is determined by the size of the sent packet. The node's transmission power and battery voltage also affect energy consumption. The size of one transmitted management frame that includes the neighbourtables is between 50 and 128 bytes, depending on the number of the node's neighbours.

VI. CiNetView; A Neighbourtable utilization tool

Neighbourtables can be used to help sensor network diagnostic and visualization. We have been using neighbourtables in our CiNetView application [17]. The CiNetView application is a graphical tool for making the deployment and monitoring of a WSN easier and more assured. CiNetView is based on diagnostic information

 ISBN: 978-1-61208-144-1

that the nodes have collected and stored to neighbourtables. The application is server-centralized and it reads information from the neighbourtable file. The application displays network topology based on relative locations produced by the MDS-algorithm. It can also use real background images and maps, where the user can exactly pinpoint the nodes' true locations. CiNetView displays the essential network diagnostic information and helps the user to see the changes in the network's behaviour. Because of the real time presentation of the network's connections and the quality of these connections, the advantages of this application can be seen most clearly in the network's deployment phase. The application can be used to diagnose and monitor a WSN through the network's lifetime.

VII. Conclusions

In this paper, we have discussed the main topics about wireless sensor network diagnostics and defined the essential metrics for WSN diagnostics. The presented neighbourtables are constructed, without significant additional transmission overhead, using modified network synchronization messages and in every node they are stored to a common data storage in the cross-layer management entity, where all the protocol stack layers can utilize the same information. Neighbourtables can be used in WSN diagnostic and management.

For future work the idea is to improve the utilization of the neighbourtables from the nodes' point of view and in general diagnostic applications. The goal is to make a database that collects historical data from the wireless sensor network and can be used in backtracking errors that have been noticed in the network.

References

[1] Y. Chen, J. Chiang, H. Chu, P. Huang, and A. Tsui, "Sensor-Assisted WI-FI Indoor Location System for Adapting to Environmental Dynamics," in *Proceedings of the 8th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Montreal, Quebec, Canada, October 10-13 2005.

[2] I. Hakala, J. Ihalainen, I. Kivelä, and M. Tikkakoski, "Evaluation of Environmental Wireless Sensor Network - Case Foxhouse," *International Journal on Advances in Networks and Services*, vol. 3, no. 1 and 2, pp. 22 – 32, September 2010.

[3] Y. Hu, D. Li, X. He, T. Sun, and Y. Han, "The Implementation of Wireless Sensor Network Visualization Platform based on Wetland Monitoring," *Second International Conference on Intelligent Networks and Intelligent Systems*, 2009.

[4] A. Meier, T. Rein, J. Beutel, and L. Thiele, "Coping with Unreliable Channels: Efficient Link Estimation for Low-Power Wireless Sensor Networks," *5th International Conference on Networked Sensing Systems, INSS 2008*, pp. 19 – 26, 2008.

[5] G. Ferrari, P. Medagliani, S. Di Piazza, and M. Martalò, "Wireless Sensor Networks: Performance Analysis in Indoor Scenarios," *EURASIP Journal on Wireless Communications and Networking*, 2007.

[6] M. Turon, "MOTE-VIEW: A Sensor Network Monitoring and Management Tool," *The Second IEEE Workshop on Embedded Networked Sensors*, pp. 11 – 18, 2005.

[7] *Wireless Sensor Network: Getting Started Guide*, Crossbow Technology, Inc., Crossbow Technology, Inc. 4145 N. First Street, San Jose, CA 95134, September 2005.

[8] H. Liu and S. S. Lam, "Neighbor Table Construction and Update in a Dynamic Peer-to-Peer Network," *23rd International Conference on Distributed Computing Systems*, pp. 509 – 518, 2003.

[9] Z. Yao, J. Jiang, P. Fan, Z. Cao, and V. O. K. Lit, "A Neighbor-Table-Based Multipath Routing in Ad Hoc Networks," *The 57th IEEE Semiannual Vehicular Technology Conference*, vol. 3, pp. 1739 – 1743, 2003.

[10] C.-S. Nam, H.-Y. Cho, and D.-R. Shin, "Efficient Path Setup and Recovery in Wireless Sensor Networks by using the Routing Table," *2nd International Conference on Education Technology and Computer (ICETC)*, vol. 4, pp. V4–156 – V4–159, 2010.

[11] A. Jacuot, J.-P. Chanet, K. M. Hou, X. Diao, and J.-J. Li, "Livencm : A new wireless management tool," *IEEE AFRICON 2009*, pp. 1–6, September 2009.

[12] A. Fynn, "Performance analysis of wireless sensor networks," Washington University in St. Louis, Available at: http://www1.cse.wustl.edu/j̃ain/cse567-06/sensor_perf.htm, Tech. Rep., 2006.

[13] I. Hakala and M. Tikkakoski, "From vertical to horizontal architecture: a cross-layer implementation in a sensor network node," *Proceedings of the first international conference on Integrated internet ad hoc and sensor networks*, vol. 138, no. 6, 2006.

[14] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Network: Theory and Practice*, ser. Wiley Series on Wireless Communications and Mobile Computing, X. Shen and Y. Pan, Eds. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: John Wiley & Sons Ltd, 2010.

[15] T. Instrumens, *Data Sheet for CC2420 2.4GHz IEEE 802.15.4 RF transceiver*, CHIPCON, Available at: http://focus.ti.com/lit/ds/symlink/cc2420.pdf.

[16] I. Kivelä, C. Gao, J. Luomala, J. Ihalainen, and I. Hakala, "Design of Networked Low-Cost Wireless Noise Measurement Sensors (ISSN 1726-5479)," *International Journal on Sensors & Transducers*, vol. 9, no. Special Issue, pp. 171–190, February 2011.

[17] I. Hakala, T. Hongell, and J. Luomala, "CiNetView - Graphic Interface for Wireless Sensor Network Deployment and Monitoring," *Proceedings of The Fourth International Conference on Sensor Technologies and Applications SENSORCOMM*, pp. 395 – 401, July 2010.