Vaasan yliopisto
UNIVERSITY OF VAASA

HEIKKI PALOMÄKI

Wireless Distributed Intelligence in Personal Applications

ACTA WASAENSIA 393

COMPUTER SCIENCE

ACADEMIC DISSERTATION

To be presented, with the permission
of the Board of the Faculty of Technology of the University of Vaasa,
for public dissertation in Anvia Lecture room (F141)
on the 4th of December, 2017, at noon.

Reviewers

Dr Smail Menani
Vaasa University of Applied Science
School of Technology/Information Technology
Wolffintie 30
FI-65200 VAASA
FINLAND

Dr Ali Hazmi
Wireless and antenna expert at Huawei
FINLAND

Julkaisija
Vaasan yliopisto

Julkaisupäivämäärä
Marraskuu 2017

Tekijä(t)
Heikki Palomäki

Julkaisun tyyppi
Väitöskirja

Orcid ID

Julkaisusarjan nimi, osan numero
Acta Wasaensia, 393

Yhteystiedot
Vaasan yliopisto
Teknillinen tiedekunta
Tietotekniikka
PL 700
FI-65101 VAASA

ISBN
978-952-476-786-6 (painettu)
978-952-476-787-3 (verkkoaineisto)

ISSN
0355-2667 (Acta Wasaensia 393, painettu)
2323-9123 (Acta Wasaensia 393, verkkoaineisto)

Sivumäärä	Kieli
158	Englanti

Julkaisun nimike
Langaton hajautettu älykkyys eri sovelluksissa

Tiivistelmä
Tietokoneet ovat historian kuluessa kehittyneet keskustietokoneista hajautettujen, langattomasti toimivien järjestelmien suuntaan. Elektroniikalla toteutetut automaattiset toiminnot ympärillämme lisääntyvät kiihtyvällä vauhdilla. Tällaiset sovellukset lisääntyvät tulevaisuudessa, mutta siihen soveltuva tekniikka on vielä kehityksen alla ja vaadittavia ominaisuuksia ei aina löydy. Nykyiset lyhyen kantaman langattoman tekniikan standardit ovat tarkoitettu lähinnä teollisuuden ja multimedian käyttöön, siksi ne ovat vain osittain soveltuvia uudenlaisiin ympäristöälykkäisiin käyttötarkoituksiin.
Ympäristöälykkäät sovellukset palvelevat enimmäkseen jokapäiväistä elämäämme, kuten turvallisuutta, kulunvalvontaa ja elämyspalveluita. Ympäristöälykkäitä ratkaisuja tarvitaan myös hajautetussa automaatiossa ja kohteiden automaattisessa seurannassa.
Tutkimuksen aikana Seinäjoen ammattikorkeakoulussa on tutkittu lyhyen kantaman langatonta tekniikkaa: suunniteltu ja kehitetty pienivirtaisia radionappeja, niitten ohjelmointiympäristöä sekä langattoman verkon synkronointia, tiedonkeruuta ja reititystä. Lisäksi on simuloitu eri reititystapoja, sisäpaikannusta ja kaivinkoneen kalibrointia soveltaen mm. neurolaskentaa. Tekniikkaa on testattu myös käytännön sovelluksissa,
Ympäristöälykkäät sovellusalueet ovat ehkä nopeimmin kasvava lähitulevaisuuden ala tietotekniikassa. Tutkitulla tekniikalla on runsaasti uusia haasteita ihmisten hyvinvointia, terveyttä ja turvallisuutta lisäävissä sovelluksissa, kuten myös teollisuuden uusissa sovelluksissa, esimerkiksi älykkäässä energiansiirtoverkossa.

Asiasanat
Langaton verkko, ympäristöälykkyys, paikannus, reititys, synkronointi, radiokontrolleri, simulointi, tekoäly

Publisher	Date of publication	
Vaasan yliopisto	November 2017	
Author(s) Heikki Palomäki	**Type of publication** Doctoral thesis	
Orcid ID	**Name and number of series** Acta Wasaensia, 393	
Contact information University of Vaasa Faculty of Technology Computer Science P.O. Box 700 FI-65101 Vaasa Finland	**ISBN** 978-952-476-786-6 (print) 978-952-476-787-3 (online) **ISSN** 0355-2667 (Acta Wasaensia 393, print) 2323-9123 (Acta Wasaensia 393, online)	
	Number of pages 158	**Language** English
Title of publication Wireless Distributed Intelligence in Personal Applications		

Abstract

The development of computing is moving from mainframe computers to distributed intelligence with wireless features. The automated functions around us, in the form of small electronic devices, are increasing and the pace is continuously accelerating. The number of these applications will increase in the future, but suitable features needed are lacking and suitable technology development is still ongoing. The existing wireless short-range standards are mostly suitable for use in industry and in multimedia applications, but they are only partly suitable for the new network feature demands of the ambient intelligence applications.

The ambient intelligent applications will serve us in our daily lives: security, access control and exercise services. Ambient intelligence is also adopted by industry in distributed amorphous automation, in access monitoring and the control of machines and devices.

During this research, at Seinäjoki University of Applied Sciences, we have researched, designed and developed short-range wireless technology: low-power radio buttons with a programming environment for them as well as synchronization, data collecting and routing features for the wireless network. We have simulated different routing methods, indoor positioning and excavator calibration using for example neurocomputing. In addition, we have tested the technology in practical applications.

The ambient intelligent applications are perhaps the area growing the most in information technology in the future. There will be many new challenges to face to increase welfare, health, security, as well as industrial applications (for example, at factories and in smart grids) in the future.

Keywords

Wireless Network, Ambient Intelligence, Positioning, Routing, Synchronizing, Radio Controller, Simulation, Artificial Intelligence

ACKNOWLEDGEMENT

The research on distributed intelligence, related to the aim of this thesis, was started at the end of the 1980's, when I developed small user programmable controller logic devices communicating in a wired network. At the beginning of the 2000's, when I started my postgraduate studies, the target of interest was very distributed automation and the possibilities of wireless communication. The electronics laboratories at Seinäjoki University of Applied Sciences with their modern equipment have been a good environment for the research of new technologies. The funding for postgraduate research at the university and the practical courses with students have been the driving force in the research into wireless technology. I am very grateful to my staff and colleagues for this possibility and their understanding about my research work and the new technology.

I am grateful to the advisor Professor Mohammed Elmusrati at Vaasa University. He has been very interested in my research and helped me to complete this doctoral thesis with guidance and worthwhile feedback. Thanks to reviewers Dr Smail Menani and Dr Ali Hazmi for valuable comments. Thanks to Professor Lauri Sydänheimo at Tampere University of Technology: he guided me to write my licentiate thesis. I also wish to thank Professor Markku Kivikoski, whose guidance and support of postgraduate studies in Seinäjoki have made it possible to continue this research. Thanks to Reino Virrankoski at Vaasa University: he organizes resources to develop and test wireless technology. Thanks also to John Pearce: he was willing to proofread and correct my English language.

The contribution of students Marko Huhta, Matti Tassi and Matti Ventä has also been significant in software development and routing simulations. Numerous other students at Seinäjoki University of Applied Sciences helped me while testing the developed technology

Special thanks are due to my family: my wife and children. They have been patient and understanding when I have spent time in my research work. I give the biggest thanks to my God and Saviour who has given me the reason to live and motives to be interested in new possibilities and technologies.

Seinäjoki, Finland, October 2017

Heikki Palomäki

Contents

Figures

Tables

Abbreviations

Ambient intelligence	Intelligent electronic devices in our ambient and in us
Amorphous computing	Very large number of identical computers or tasks
Amorphous automation	Very distributed automation
Distributed intelligence	Computing is divided into small independent units
Exergame	Outdoor exercise game controlled by information technology
Flat mesh	Network without any hierarchy
Positioning	To find coordinates, where an object is located
6LowPAN	Internet over Low power Wireless Personal Area Networks
API	Application interface
ANT	Wireless sensor network technology for fitness applications
DD	Direct Diffusion
DLL	Data Link Layer in communication protocol stack
FFD	Full Function Device
FIFO	First In First Out
GPS	Global Positioning System
ID	Identification number
ISM	Industrial, Scientific and Medical
LNA	Low-Noise Amplifier
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Medium Access layer in communication protocol stack
MEMS	Micro Electro Mechanical Systems

MSDE	Mean value of Square of Distance Errors
NFC	Near Field Communication (mostly RFID technology)
OEM	Own Electronic Manufacturer
PHY	Physical layer in communication protocol stack
PLC	Programmable Logic Controller
PLL	Phase Locked Loop
RF	Radio Frequency
RFD	Reduced Function Device
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indication
SCADA	Supervisory Control and Data Acquisition
SoC	System on Chip
SOM	Self-Organized Map
SPI	Serial Peripheral Interface
TUTWSN	Tampere University of Technology Wireless Sensor Network
UWB	Ultra Wide Band
WLAN	Wireless Local Area Network

1 INTRODUCTION

People are depending on each other and on the environment. Therefore, the connection between people and things around us is necessary for living. The basis for the development of our civilisation is to generate and to share information. The required information is more and more in digital format. Information sustains the structure of society as well as our daily lives. It is becoming more and more important how to connect information processing units with other ones and how intelligent these units are. The computing devices are no longer only data transfer and calculation units, but are also able to make independent decisions and be capable of information filtering.

Information transfer makes it possible to monitor and control processes remotely without having a physical presence. In novel IoT applications, the control of traffic lights, bus timetable displays and other public traffic are all realized in smart cities. The balance between energy producers and users can be controlled remotely in smart grids, especially private energy producers and renewable energy sources. In smart homes domestic appliances, temperatures and ventilation can be controlled via IoT applications without having a physical presence. Summer houses can be safely left unoccupied by remote monitoring their conditions and controlling the lights etc. This all increases security and people's ease of living.

The topic of this thesis is to show new areas and applications that are possible with the development of: wireless technology, distributed computing and artificial intelligence. Traditional technology is safe and well known. Automation and our whole life seems to be more risk-free by using only traditional technology in industry and in household management. Actually, to stay in traditional technology results, in the long run, in a reduced ability to compete and an increase in risks when managing our changeable world. In addition, one aim of this thesis is to open people's vision to new possibilities and to prepare for future automation and management styles that lead to successful and safe living in the future.

Money controls the sharpest tip of technology development. It means that the first winners are those who can invest a lot and have the motives to do so. For this reason, new development serves mostly industry, the military and space technologies. These are good technology drivers, but the results of technology reach the daily life of people much later and only in the form of mass production where companies can gather a lot of profit. Related research groups at universities also perform high scientific analysis, but practical applications are either missing or are too complicated and have too high a cost to be used in household management. This thesis provides a simple but efficient technology and shows application possibilities to use them in the daily lives of people in personal areas as well as in small companies. It is very important that people are winners in new technology development, not only organizations.

The most straightforward way to become familiar with new technology and electronics is to develop and use the most efficient structures first (in terms of simplicity and cost) and thus to discover new possibilities. After studying the technology, the current state of the art and the new information processing methods, your eyes are then opened to see and imagine new ways to use this knowhow in new demanding situations. This thesis describes the existing technology, the physical electronic development needed to have a testing platform, simulations to test the functionality of the planned methods as well as application ideas. Some methods realized in the study case applications are based on the developed platforms. The purpose of the simulations and the study cases is first to show that interesting theoretical methods can also function in practice. Second, the purpose is to prove that smart applications can be made in a simple, low-cost format that can be useful in the everyday environment.

The programming language is standard C for radio controllers. While a PC is collecting data from a wireless network the programming language is mostly C# (C sharp) and in one case Python. As an output, the practical cases generate data tables, which is shown in a visual form using Excel charts. Excel and C# language are the calculation tools in the simulation cases. Excel macros are also useful to control calculations. Simulations generate tables and are visualized in Excel charts.

The contributions to this thesis were realized over several years while I was principal lecturer in electronics at Seinäjoki University of Applied sciences. The development, research and test work for this thesis started in 2004 when studying the simplest wireless technology. The teaching of embedded systems was strongly focused on electronic development; therefore, the aim was to develop all of the electronics by ourselves and to utilise them also in teaching. All circuit board layouts are my own design, including also the embedded system teaching kit. The focus was not on ready-made electronic modules and communication standards, because we had development resources and motives to make them ourselves. The students made a significant contribution to the work by developing a programming environment to edit software to be used for a radio controller. The research flows parallel with some courses, therefore it was important to develop and to use only free software tools without license limits. The first wireless transceiver circuit board was the optional module for the embedded system teaching kit. Later in 2006 the electronics was developed in a project funded by LifeIT Oyj. The goal was to develop wireless game modules for children and this technology was tested for the first time in a real environment. A routing method simulation and some practical tests were carried out in a student's thesis. This simulation showed some new aspects that need to be recognize in networks with a large number of nodes. Connected to a bus technology course, a student class tested radio communication protocol in a larger network. It gave a better understanding of communication range in a real environment.

The university offers teaching-free research periods for study and written work. As a result, I had my licentiate thesis ready in 2008 (Palomäki, 2008). To make a better contribution to research work funded projects are needed. Our university had a cooperation project GENSEN with Vaasa University and Aalto University. The project gave extra resources to develop network protocols. The Vaasa University people organized some study cases to test the developed wireless ideas in practice. It provided the way to get a practical and very useful response to fix radio communication, low-power features and routing methods. These contributions significantly supported my research and development work. The second teaching-free research period made it possible to contribute with additional network tests and thesis writing.

The thesis structure is as follow: Chapter 2 includes the history of distributed computing development and corresponding background and comparisons. Chapter 3 includes existing wireless standards, ways to use them and routing algorithms. Chapter 4 describes information-processing methods of artificial intelligence including fuzzy logic and neural networks. Chapter 5 estimates the challenges of distributed automation in future applications. Chapter 6 describes the development of electronic modules with layouts and features. Chapter 7 includes all simulations of wireless networks, its usage and applications. Chapter 8 includes some test cases based on simulations combining real networks with smart routing, positioning and measuring methods. In addition, it includes some application possibilities not tested in practice. Chapter 9 is the conclusions and discussion about topics section.

2 DISTRIBUTED INTELLIGENCE

A very interesting development line is the roles of computing units that generate, store and use information. The first principle in history was to centralize information and decision making capacity. In this case, connections are in a star format: all data generating and exploiting units must connect themselves with a mainframe computer. This structure is simple to keep control of, but it is vulnerable: one single error in the mainframe computer stops the whole information system and the functions depending on it. In many cases, like in the instance of space research, it is desirable that the remote system continues functioning on its own, making its own decisions and connecting data regardless of the failing connection with the command centre. Actually, in many applications the trend is toward distributed intelligence in which information producers and exploiters communicate with each other directly without mainframe data storage. The communication backbone is today mostly based on the Internet Protocols (IP). In addition, even smaller units have the feature to connect to the network. The Internet of Things (IoT) is one good example of this development line.

Reliable communication between the system entities is crucial for successful system integration. According to CISCO, during last year (2016), the total IP traffic has surpassed 1.1 zettabyte (billion terabytes). This number will further increase to about 2.3 zettabyte per year by 2020. About 70% of the total IP traffic will be handled in the last mile by wireless transceivers (Elmusrati, 2017).

Wireless communication makes it possible to connect mobile objects, like phones, navigators, control tags and vehicles. In this way, it is simpler to realize distributed computing without wiring restrictions and costs. Generally speaking, wireless communication has several challenges when compared to wired communication. For example, wireless communication has higher packet losses because of channel fading, it suffers from higher latency because of competing for limited bandwidth resources, and wireless communication is generally less secure. Hence, it is important that the remote system is able to function with local connections and with a limited central connection. One of the most interesting trends is ambient intelligence applications, where all objects around can communicate wirelessly and function intelligently independently. The personal examples of these are access control, health monitoring, modern games, smart clothes etc. A novel trend seems to be managed chaos, in which all objects have some wireless connections but are intelligent enough to function by themselves. Social insects and cells in tissues function in this managed chaos style. Nature is an important example for new technology: as described in the old wisdom: "Go to the ant, you sluggard; consider her ways and be wise" (Solomon, 1965, s. 705).

This thesis describes wireless technology features suitable for new ambient intelligent applications: ambient intelligence is embedded and invisible computing and communication functions implemented in our neighbourhood: clothes, tools, domestic appliances, buildings, furniture, traffic signs etc. Ambient intelligence is normally distributed into very small units that need to be serviced seldom or not at all. The aim of this research work is to estimate the demands, possibilities and features of short distance wireless technology.

The wireless technology today is mostly multimedia communication, where the main purpose is to carry voice, video and files via a wireless channel. This kind of wireless technology is not discussed in this thesis. The focus is on a Low-Rate Wireless Personal Area Network (LR-WPAN).

2.1 History and future of computing

The direction history has taken must be known to know possible future directions. The current technology is mostly based on history and this can be a disadvantage for technology today. The important question is how to develop wireless technology to fit future demands (Palomäki, 2008, ss. 2-4).

2.1.1 From mainframe to interactive life

One good principle for understanding the development of computers is to estimate how large an area a single computing unit covers in the life of people. Weiser (1996) sees major trends in computing as three phases: mainframe, PC and ubiquitous computing (Figure 1).

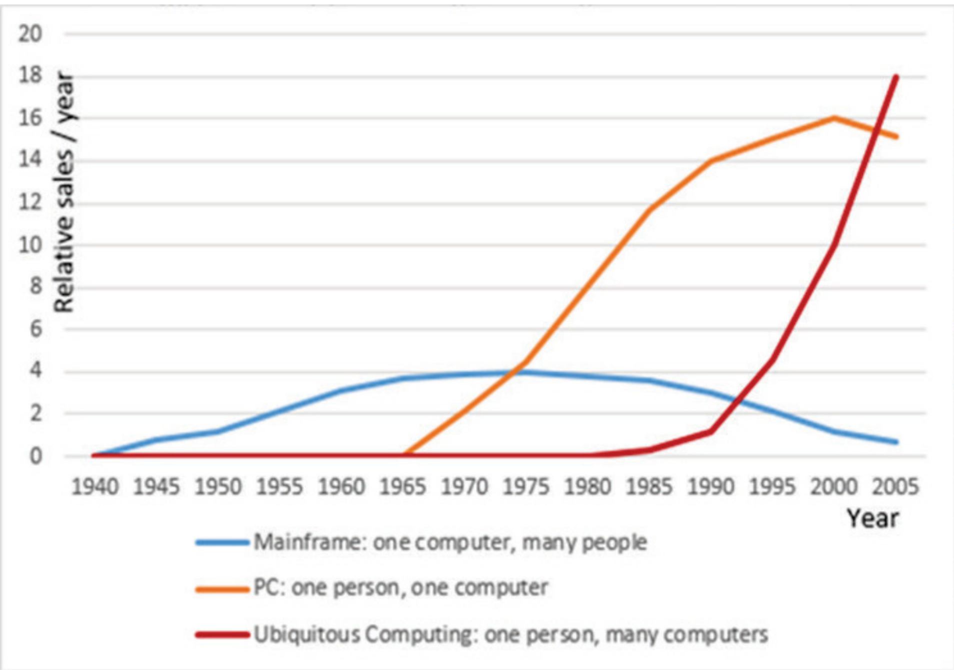


Figure 1. The major trends in computing (Weiser, 1996)

Vertegaal (2003) describes the computing trends in 4 steps based on how many user interfaces one single person uses: 1. many uses one, 2. one uses one, 3. one uses many and 4. many use many. I also see the history of computing in four development phases, but based on the user interface instead; in the first phase, mainframe computers need most knowhow for the user interface. In the last phase are devices that are fully embedded and function independently without any user interface (Figure 2).

In the first development line, one single computer covers or fulfils the needs of one university or one factory. This was the development from programmable electromechanical calculators to mainframe computers. At that time, the developers believed that fewer than ten computers would be enough to fulfil all the computing needs of the world. The mainframe computers were very complicated to use and they needed scientists as operators.

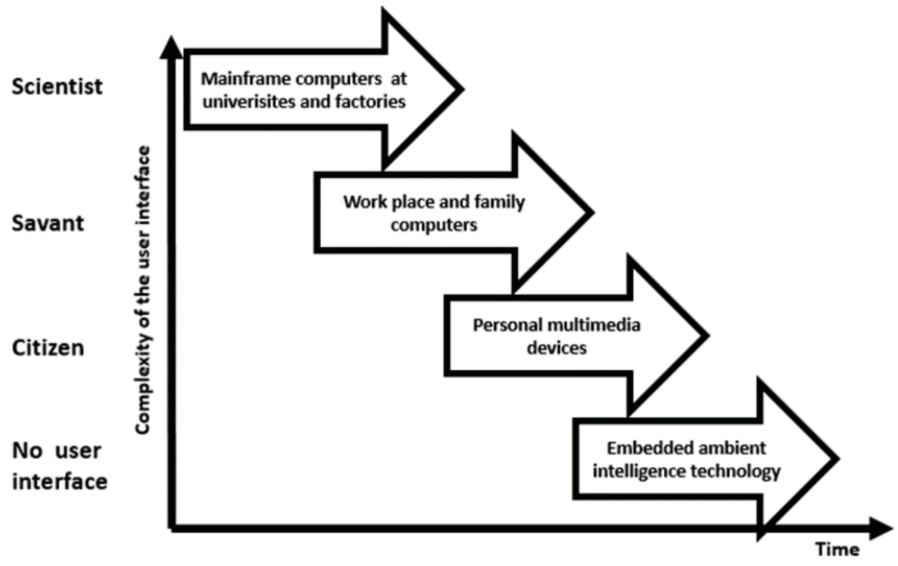


Figure 2. Estimation of computing development lines

The second development line is microprocessor-based technology, starting from the Intel 4004 processor and going on to the personal computers of today. This development line fulfils the need of a single work place or one family. The effective use of a personal computer needs some studying of informatics.

Another microprocessor development line at the same time is automation applications. Multifunctional microcontrollers replace microprocessors to be suitable for small automation. The next development step was single-board computers for universal use in automation and information technology. The most powerful devices in this development line are Scalable programmable logics for industrial automation.

The next development line goes through personal multimedia devices: mobile phones, music and DVD players, tablets and navigators. One person needs one or more devices of this kind. Using a multimedia device is simpler than using a PC. If you can read and write, you can also use the basic features of a multimedia device. However, you must be able to read and understand the user manual and you must be able to use a complicated keyboard. For babies, demented people and disabled persons the use of multimedia devices can be too difficult.

The last development line is ambient intelligence, which is located around us invisibly. The computing intelligence embeds in keys, locks, domestic equipment, wristwatches, traffic lights, cash desks etc. Future applications include for example the monitoring of children, exercise games, health control and condition monitoring in industry. One single person or work place needs numerous embedded computing and communicating devices. This newest development line is still seeking its directions. We expect that in the future ambient intelligence technology will capture new application areas and can soon be the fastest growing area in informatics.

Every technology line has started with simple electronics based on earlier technology and then developed into more powerful devices that perform complex functions. The first line includes some tens of computers. The second line contains some billions of PCs and control computers today. The number of mobile connected devices has been estimated to be 11.6 billion by 2021 (Cisco, 2017). It is perhaps impossible to count the number of ambient intelligent controllers in the future; today it is a very fast growing area.

2.1.2 From centralized to distributed automation

In industrial applications, informatics developed earlier when compared to private and science informatics. The mainframe computers were first used in economy control. In the next step, small controllers grew to powerful control and monitoring systems to control a single production line or a paper machine. In the third development line, programmable logic controllers (PLC) controlled a limited task at a complex production factory, which was then controlled and monitored by PC-level microcomputers. The fourth development line is now going on in the form of distributed automation: wireless sensor networks and intelligent actuators. Because of security and latency problems, wireless technology is not yet in large use in industrial applications.

In any case, distribution gives another kind of security and reliability; if the structure of a distributed system is dynamic, it is allowing nodes to replace each other. The example of a bear and an ant nest shows this feature: when you are hunting with a gun in the forest, you can meet a bear and destroy this creature with a single accurate shot. You can also see an ant nest. With a single shot, you can make a slight disturbance in the life of the ant nest, but afterwards nobody can see any sign of your shot in the nest. The effect of one failure can destroy the whole system if it is centralized. In a distributed system, other resources replace the failed parts and it results in only a small disturbance in the normal operation.

2.1.3 From point-to-point to wireless mesh communication

Each one of the development lines described above has its own types of communication methods. The mainframe computers in universities had a lot of wired ports to interface with teletype terminals. The only way to communicate was to use point-to-point wires.

In the second development line the most important way of communication was the internet realized by worldwide links, local area Ethernet and WLAN. The third development line in multimedia brought NMT, GSM, GPRS, 3G (UMTS, HSPA+), and 4G (LTE-A) technologies. Moreover, the 5G network is still under development for a new cellular network standard. One of the motivations for 5G is to support industrial applications by enhancing reliability and minimizing latency. The standard development of 5G is under way and it has high objectives: the geographical coverage and availability are planned to be 100%. The planned connection speed is up to 10Gbps with 1 ms latency in end-to-end connection. One of the interesting features is the planned energy consumption: The lifetime for a battery powered machine-type device can be up to ten years. This is significant, when planning IoT-based sensor networks in future (Intelligence, 2014).

This second development line includes also multipoint factory buses to connect PLC's and SCADA (Supervisory Control and Data Acquisition) devices. These kinds of systems are developed for industrial purposes.

One part of the fourth development line is the current technology in RFID and charge card applications. More smart applications are seeking their own form and standards for communication. The most common physical communication method is a short-range wireless link. The first network topologies are hierarchical, i.e. sensor networks. The ZigBee standard has some kind of mesh topology and TUTWSN (Tampere University of Technology Wireless Sensor Network) has full mesh topology (Kohvakka;Suhonen;Kuorilehto;Hännikäinen;& Hämäläinen, 2007). Perhaps the new ambient intelligence applications will use non-standard or very scalable standard wireless communication methods. These wireless technologies are seeking their own forms and features, depending on the applications they are to be used in.

2.1.4 From computing units to smart sensors

The structure in traditional automation can be divided into three parts: computing unit, interface and sensors/actuators. These are all separate and sometimes far from each other. The main intelligence and developed software was only in the computing unit. Along with the development of multifunctional microcontrollers, the intelligence is more and more in the sensors and actuators. In embedded ambient intelligence wireless devices, the whole system including sensors and even actuators are on the same small circuit board sensing and acting according to acceleration, temperature, light and/or humidity.

The newest MEMS (Micro Electro Mechanical Sensor) technology makes it possible to integrate the sensors, actuators and user interface in a very small and intelligent form that can be used, for example, in a RF button. The MEMS chip is a silicon chip that has also mechanical structures. Force, rotation, pressure, weight, inclination and acceleration are very simple to measure. Actuators are not easy to realize in very small form; only micro motor and pump structures have been tested and they can be realized. Keyboards and command buttons can be replaced with acceleration, gyro and motion MEMS sensors. The user does not need to enter commands to a simple ambient intelligence device, but the behaviour of the user is the input needed for events. A text or graphical display is not needed in simple applications. Voice and light signs can replace them to give information on events happening in the ambience (Chang;Lee;& Chih-Yung, 2007).

For example, STmicroelectronics has different types of MEMS sensors: acceleration, gyroscope, compass, microphone as well as environmental sensors: pressure, temperature and humidity. The current applications for MEMS sensors are motion tracking, compass and navigation in smartphones and tablets. The gyro and acceleration sensors are used in gaming devices. Many fitness, wellness and home appliance applications use MEMS sensors. Crash detection and many smart sensors are used in car technology. Many robotic and medical applications also use MEMS sensors. Mostly the applications above use sensors to sense movement and position automatically similar to the user interface, but without the user interface. (STMicroelectronics, 2016).

2.1.5 From computer connections to object connections

Mainframe computers are connected to each other and with terminals; this is required for the normal use of computers. This also means that generated scientific information is more widely spread.

The development of the current multimedia devices is focused on connecting people together with high communication speeds and versatile application software. The current trend seems to be new social media methods, virtualization and augmented reality. It means that people are connected also with a virtual or fictional world. (Palomäki, 2014b). In any case, this technology was left out of this thesis.

In ambient intelligence technology, the objects are connected wirelessly together. There exists a huge set of applications to connect all kinds of objects with each other and with control systems. The objects are either moving or there are so many objects that wiring is not a worthwhile choice. It is a new idea to connect small numerous objects together in a smart way using wireless. This study focus on low-level connections where, as a minimum, an object can tell the wireless network at least: “I’m here” or “I see you”. The set of applications includes monitoring and the control of clothes, animals, children, personal estates, tools, toys, access and identification. Applications of this kind are quite new; excluding access control. Connecting objects makes dumb things in our environment smart.

2.1.6 Smart dust

There are many technologies trying to distribute intelligence in smaller and smaller devices. One technology area is called ‘Smart dust’. The plan is to have wireless devices; which are tiny MEMS units with a set of sensors. The size of these motes can be the size of a grain of sand. They can process information and communicate with other neighbour motes up to a range of as much as 300m. The goal for researchers was to get chips with 1mm sides. The planned structure of a single smart dust mote is in Figure 3. (Hoffman, 2003)

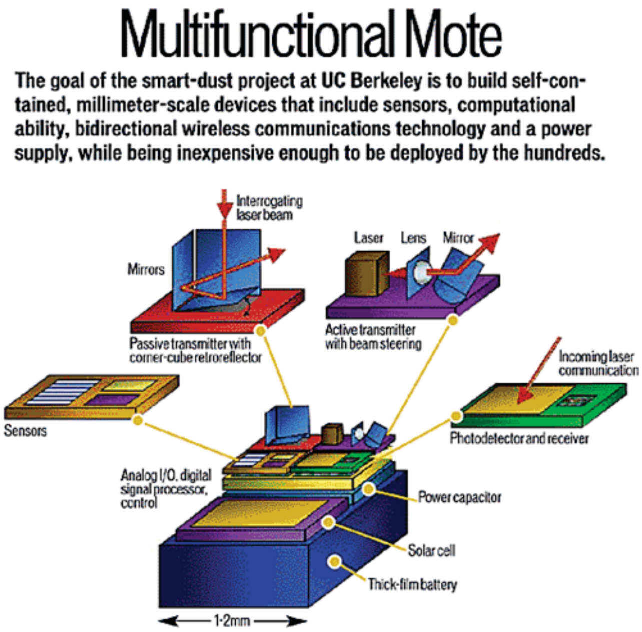


Figure 3. Smart dust mote (Hoffman, 2003)

Many research groups are developing wireless Smart Dust technology around the world. Smart dust is actually a theoretical scenario in wireless technology, in which the intelligence is highly distributed. A single device would be an independent, intelligent measuring and communicating chip with the size of a millimetre cube. The goal of these development projects is that these chips can be seeded in the ambience to sense conditions and to send the data forward wirelessly. There are also plans to develop some kinds of micro actuators for micro robotics. For example, a single smart dust chip can consist of a battery, a solar shell, a power capacitor, a controller with an analogy interface, sensors and optical components; all the size of some cube millimetres. Optical communication is carried out with a MEMS laser diode and a CCR sensor. One example mote has been developed having the size of 63 mm3. (Warneke;Last;Liebowitz;& Pister, 2001)

The smallest independent working device is a RFID μ-Chip developed by Hitachi in Figure 4. The size of the chip is 0.4mm x 0.4mm. However, it is a RFID tag and needs an external magnetic field for power supply, and functions only as an identification chip. (Hitachi ltd, 2003)

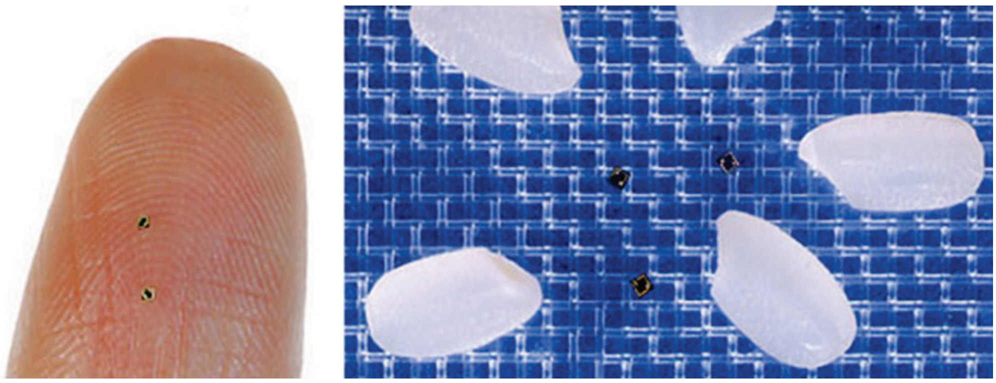


Figure 4. RFID μ-Chip (Hitachi ltd, 2003)

Smart Dust gives the biggest benefit in space technology. Scientists are exploring a space telescope with swarms of particles. The swarms form floating lenses controlled by a laser, which are cheaper and lighter than conventional space telescopes. It is possible to form lenses even thousands of kilometres in diameter. (Gawlowicz, 2014)

The other planned application for space technology is planet exploration. Particles, which are small enough, can fly in the wind. The computer chip controls the shape of a plastic sheath enabling it to steer the particles. Using wireless communication, these particles form networks and include small sensors collecting information about the planet. (Rincon, 2007)

If smart dust scenarios are realized, the ambience can be intelligent. People and devices can have a view of the ambience via a smart dust wireless network and get important data on current or past conditions and events.

3 WIRELESS PROTOCOLS AND TECHNOLOGIES

3.1 Existing wireless technologies

Today's wireless communication standards were developed to fulfil the existing demands of the time and anticipated future demands. However, all of the standards were developed some time ago, so it is uncertain, that they fit the present day demands or future ones. The existing technologies were studied to find out how suitable they are for distributed ambient intelligence. In this study, only wireless technology using free ISM (Industrial, Scientific and Medical) frequency bands was used. These bands can be used freely without license, but with restrictions on the allowed maximum effective transmission power (European Communications Office, 2016, ss. 115-116). The most used band is 2.4 GHz, therefore the focus was on technologies use this band.

3.1.1 Low Energy Bluetooth

L.M. Ericsson in Sweden started in 1994 the development of the first version of Bluetooth. In 1998 five companies formed a Special Interest Group, which grew in four years to a cooperation of 1500 companies. The basic aim was to connect computers wirelessly to peripheral devices. Bluetooth has been designed for high-speed point-to-point communication, but it has the same good features required for a wireless network. (Dursch;Yen;& Shih, 2004)

For small, battery powered control and monitoring applications; Bluetooth has a low-energy alternative BLE (Bluetooth Low Energy) standard developed by Bluetooth SIG (Special Interest Group). The structure of a Bluetooth network can be modified dynamically. One Bluetooth node senses the nearness of other nodes, adapts it to the network structure and starts the communication period. The network structure of Bluetooth is piconet, where one master connects with numerous slaves. Piconet is a basic network topology where multihopping is not possible. The classic Bluetooth enables routing between piconets (Scatternet), but in BLE it is no longer possible. A slave has a 32-bit identification address and theoretically, it can have a huge network size, but in practice in low-energy applications, the number of slaves is between 2 and 11. The bit rate in air is 1 Mbps, but the maximum application layer bit rate is 236.7 kbps. BLE requires memory resources in the controller: 40 kB ROM and 2.5 kB RAM. BLE technology is focused on being used, for example, in smartphones and in similar applications (Gomez;Oller;& Paradells, 2012).

BLE uses 40 RF channels in the ISM band 2.4 GHz, Transmission uses an adaptive frequency hopping mechanism between 37 available channels (Figure 5). Physical channels use a GFSK (Gaussian Frequency Shift Keying) modulation and the bit data rate in air is 1 Mbps. In wireless applications, where nodes are moving and the structure of the network is dynamic, the connection time of a single node is significant. The connection time (the latency delay) in the BLE network is usually from 7.5 ms up to 4000 ms (Gomez;Oller;& Paradells, 2012).

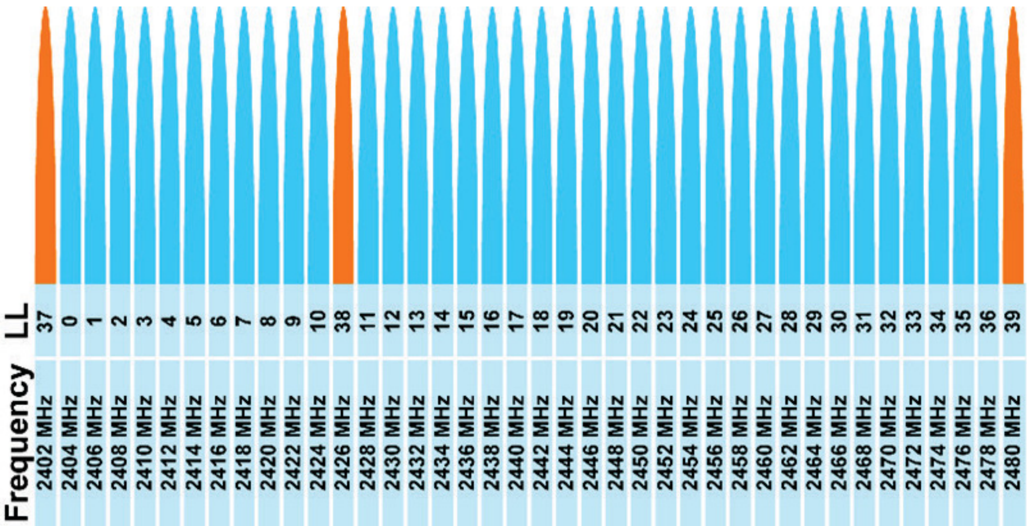


Figure 5. BLE channels (Nilsson, 2013)

The BLE standard was developed to give the sensors a lifetime from several months to several years using a single coin battery cell. The current consumption when transferring 16-64 bytes per second is about 40-90 uA (Kindt;Yunge;Diemer;& Chakraborty, 2014).

BLE is very suitable for simple sensor networks controlled, for example, by a laptop or smartphone. The possible applications are wearable technology and health monitoring devices. The lack of multi-hop limits the use of BLE in most ambient intelligent applications, because the whole network must be located inside the master's range.

3.1.2 ZigBee

ZigBee is a wireless technology for industrial applications, especially in WSN (Wireless Sensor Network) applications. The ZigBee Alliance was established in 2001 the first ZigBee specification enabled more than 65000 nodes in a single network. The standard IEEE 802.15.4 defines the physical layer and medium access control of ZigBee protocol architecture. The ZigBee protocol architecture stack includes the network layer and the application layer as shown in Figure 6 (Gomez & Paradells, 2010).

Physically, ZigBee has two node types: FFD (Full Function Device) and RFD (Reduced Function Device). FFD has normally a continuous power supply and includes all ZigBee features. RFD is low power, possibly a battery powered device having no routing features. The ZigBee network consists of one coordinator, several routers and end devices. A FFD can function as a coordinator or a network root, initializing the network and setting operational parameters as well as function as a router receiving and transmitting data between nodes. A RFD is normally an end device connected to one router or coordinator. Mostly the RFD is a battery powered sensor or a switch. In addition, a FFD can function as an end device with a continuous power supply and is more suitable as an actuator (Somani & Patel, 2012).

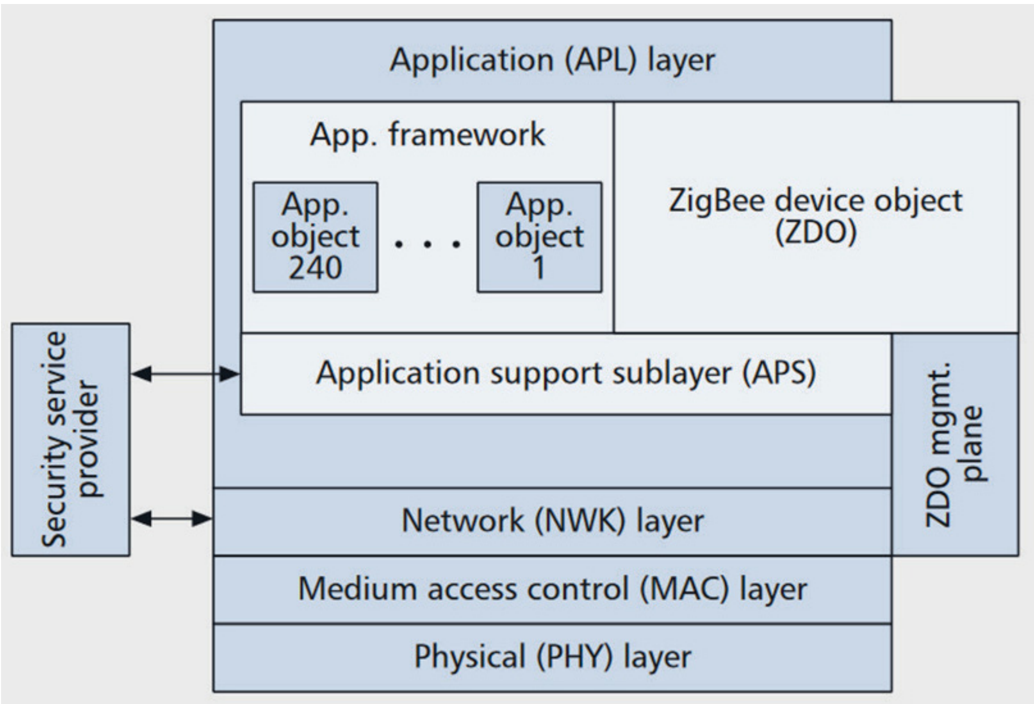


Figure 6. ZibGee protocol architecture (Gomez & Paradells, 2010)

IEEE 802.15.4 standard was defined to use the 2.4 GHz band frequency area between 2405 MHz – 2480 MHz using sixteen 5 MHz channels (Figure 7). The modulation method is DSSS (Direct Sequence Spread Spectrum) using O-QPSK (Offset Quadrature Phase Shift Keying) (MaxStream, 2007).

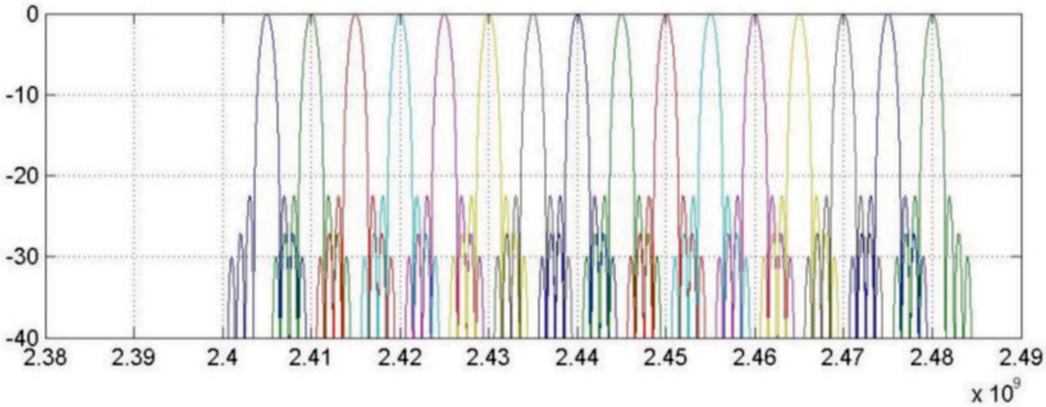


Figure 7. IEEE 802.15.4 channels (MaxStream, 2007)

The ZigBee network is suitable for wireless sensor networks and other applications including sensors and actuators. Multimedia applications are not suitable for ZigBee, because its data rate is quite low. The most important advantages of ZigBee for ambient intelligence are the very low power consumption and the mesh topology of the network. The end nodes in the network can be battery-powered. The data transmission range can be large in the mesh topology because of multihop routing. The size of a ZigBee network can be up to 216 (short address) or 264 (IEEE address) nodes, hence, in practice there are no limits to the number of supported nodes (Callaway, 2003).

ZigBee has been developed mainly for sensor networks, so there are some disadvantages concerning ambient intelligence applications. Only the RFD nodes are really low-power, because they have only one synchronized communication channel with one FFD node. The FFD nodes communicate both with their own RFD nodes and with other FFD nodes. The RFD node can be battery-powered, but the FFD normally has a continuous power supply. The ambient intelligence applications must take into account the power limits of these routing nodes.

A commercial ZigBee manufacturer must join the ZigBee Alliance and pay a membership fee of $4000 - $50000 per year to get the rights to use the 'ZigBee' trademark. The fees depend on, whether the member is an adopter, participant or promotor. This is a significant disadvantage for small and medium sized companies developing wireless technology (ZigBee Alliance, 2016)

3.1.3 6LowPAN

6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks) is some kind of ZigBee extension. Protocol has been developed by the 6LowPAN Working Group to have access to ZigBee nodes via the Internet. This 6-byte Internet addressing mode gives interesting possibilities to control and monitor a personal area network. This technology gives a good way to realize IoT (Internet of Things) applications with ZigBee. 6LoWPAN is focused on ZigBee, but it is possible to take the same idea in use in other PAN area networks to realize a bridge between the Internet and the objects around us (Mulligan, 2007).

If 6LoWPAN network is larger and needs routing, it can function with mesh under routing or route over (Figure 8.). In "mesh under" routing, the routers are following IEEE 802.15.4 standard routing and are not using IP-addresses; the route is a single IP hop. In "route over", every router has an IP address and so every hop is an IP-hop (Gomez & Paradells, 2010).

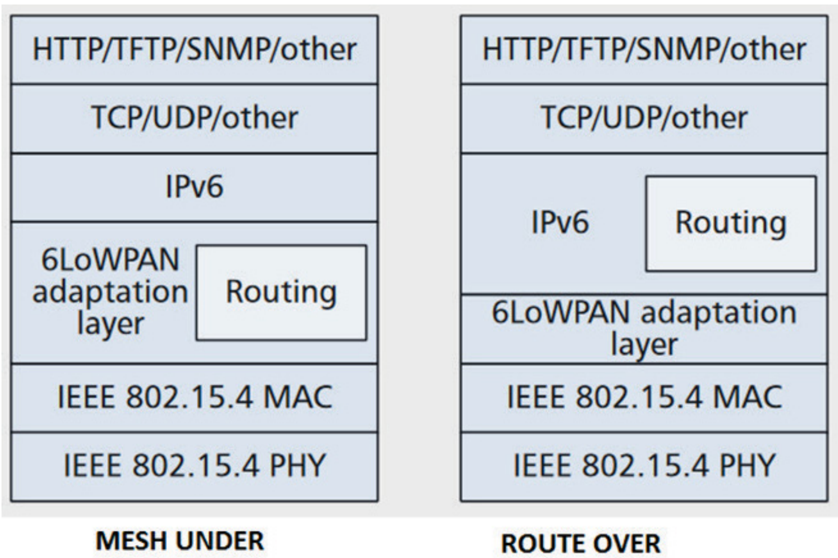


Figure 8. 6LoWPAN protocol architecture (Gomez & Paradells, 2010)

Z-Wave

Wireless Z-Wave technology is suitable for home control systems developed by the company ZenSys (a division of Sigma Design). The Z-Wave alliance controls the use of technology. A member can only be certified as a Z-Wave developer and the annual fees are 250$ to 4000$ depending on the membership level (Z-Wave alliance, 2017).

The Z-wave protocol uses different RF frequency bands: 868MHz (In Europe), 908MHz (In USA) and 2400MHz depending on the chip series. It uses BFSK modulation and the communication bit rates are 9.6kb/s, 40kb/s or 200kb/s (only with 2400MHz band), the range is from 30m (Indoor) to 100m (Outdoor). The Z-Wave protocol stack has 5 layers: physical, MAC, transfer, routing and application layers, as shown in Figure 9 (Gomez & Paradells, 2010).

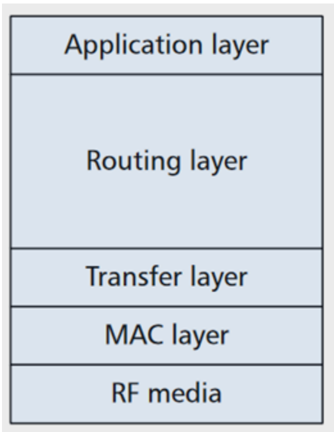


Figure 9. Z-wave protocol architecture (Gomez & Paradells, 2010)

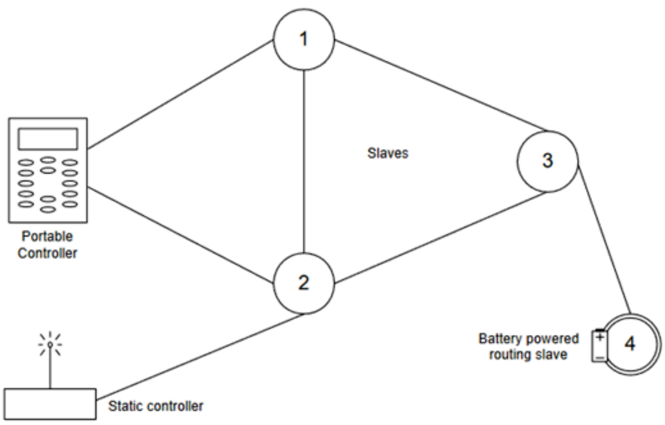


Figure 10. Z-Wave network example (Zensys A/S, 2006)

Z-Wave protocol consists of controlling devices and slave nodes. There are four different controller types: A portable controller, which can function, for example, as a remote controller in the system. A static controller, which can function, for example, as an internet gateway. An installer controller, which is required by the installer to set up the network. A bridge controller, which connects the Z-Wave to other different networks. There are three different Z-Wave slave types: The simplest slave cannot communicate directly with the other slaves but can receive commands, functioning for example as a light dimmer. A routing slave can send messages to other nodes and functions, for example, as a sensor. An enhanced slave, which has added resources compared with the routing slave, has clock and memory and can function, for example, as a weather station. The principal structure and node roles are shown in Figure 10. (Zensys A/S, 2006)

ANT protocol

Dynastream Innovations has developed the wireless communication protocol ANT for short range and very simple applications. ANT handles peer-to-peer, star, tree and fixed mesh topologies. It includes physical, network and transport layers of an OSI stack. In any ANT topology, there is one or more HUB nodes, which serves a host application controller. The network can include RELAY nodes as routers, which connect HUB nodes together. The sensor nodes can be in contact only with a fixed HUB node and they are primary transmitters i.e. masters of the communication channel. The lowest-power sensor has a one-way link and can only transmit, but sensors can also have a bidirectional link to the HUB. The communication channels are fully synchronized: the master can send within a defined time slot its data to the slave and the slave can response immediately if it uses a bidirectional channel (Dynastream Innovation Inc., 2014).

ANT protocol can use the frequency area 2400 MHz to 2524 MHz with 124 RF channels with 1 MHz increments. In an actual ANT chip there are only 78 RF channels using narrow band GFSK (Gaussian Frequency Shift Keying) modulation. The on-air bit rate is 1 Mbps and the maximum data byte rate between nodes is 20 kbps. It is possible to detect logical proximity between nodes using a maximum of 10 steps with the proximity search feature (Dynastream Innovation Inc., 2014) (Nordic Semiconductor, 2010a).

The ANT device can be in the form of a separate host controller (MCU) and ANT module. Some manufacturers have a SoC device (System on Chip), where the controller and ANT protocol stack are on the same chip, as shown in Figure 11 (Dynastream Innovation Inc., 2014).

The primary application of the ANT protocol is in the sport and fitness field, as indicated in Figure 12. It is very suitable for wearable technology to add intelligence to clothes and the body: to monitor health, condition and movement. The ANT protocol was developed for a specific type of application. Therefore, it can include some limitations when using it in some ambient intelligence applications.

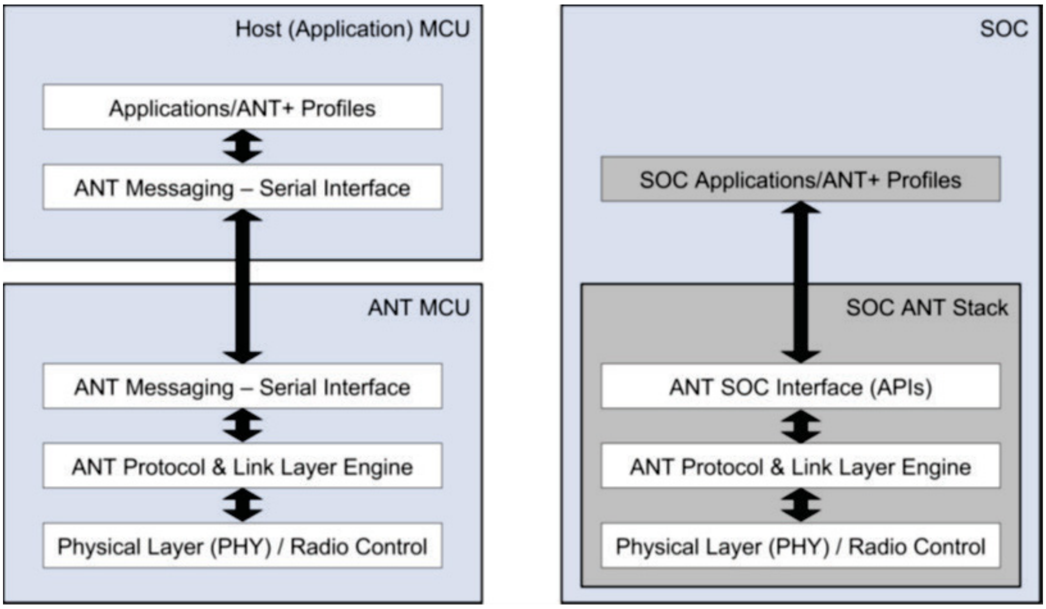


Figure 11. ANT layer alternatives (Dynastream Innovation Inc., 2014)

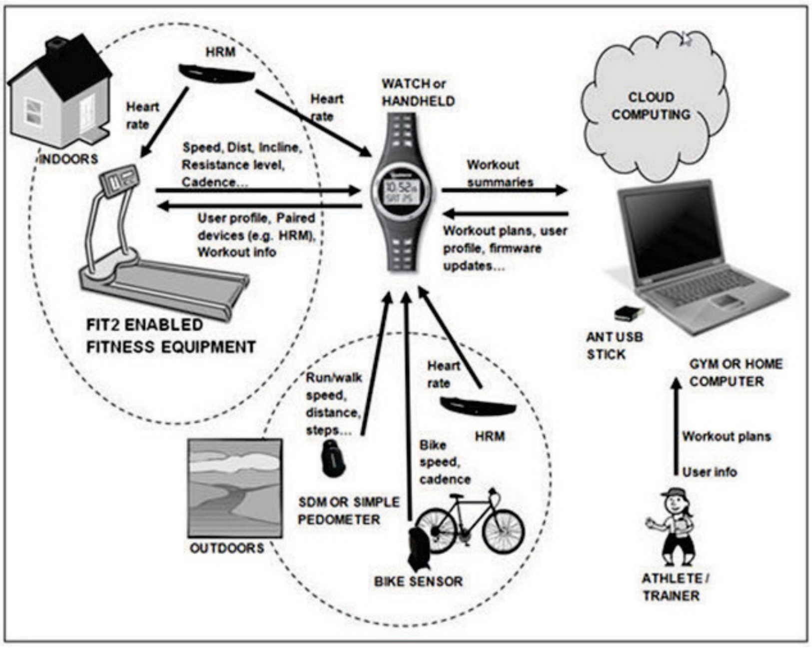


Figure 12. ANT application examples (ANT wireless, 2017)

3.1.4 TUTWSN

Tampere University of Technology has researched wireless sensor network technology in the DACI research group. This group has developed the wireless sensor network TUTWSN (Tampere University of Technology, 2009) with optimal features. The network topology is a full ad-hoc mesh. The number of nodes and multi-hops is unlimited. The network is fully distributed, i.e. it does not need any coordinator or external computing (Figure 13). As the network structure is modified, it repairs and reorganizes itself. TUTWSN has been developed especially to rule dynamic changes in a network, i.e. for mobile applications. This network is very low-powered, for example, the life-time of a badge node with 2 buttons and 3 LED lamps is 3-5 years using 2xAAA batteries. (Kohvakka;Suhonen;Kuorilehto;Hännikäinen;& Hämäläinen, 2007) (Hämäläinen & Hännikäinen, 2007)

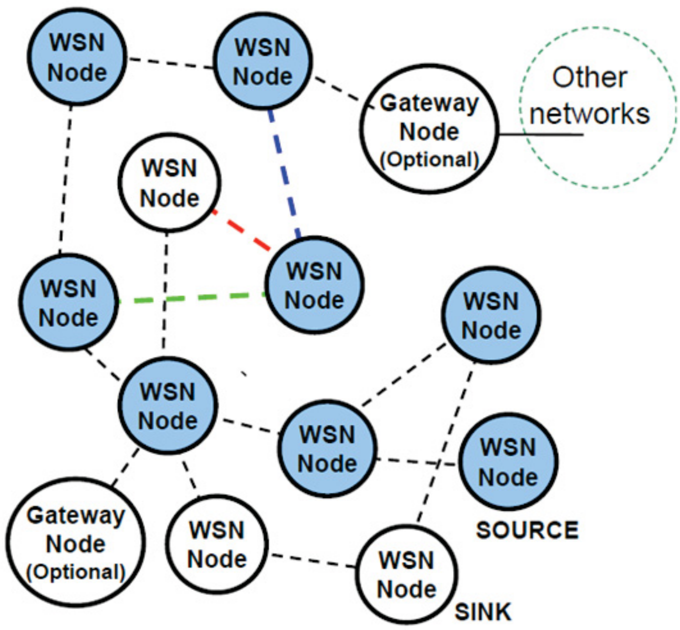


Figure 13. TUTWSN topology (Hämäläinen & Hännikäinen, 2007)

TUTWSN is highly suitable for universal use in ambient intelligent applications. There are more than 40 node types and also gateway nodes for Ethernet, Bluetooth and WLAN. The smallest node is a wrist node, which is suitable for a key chain.

The only disadvantage is that TUTWSN is a customized technology. It is not open source and today all applications are made by the TUTWSN research group.

3.1.5 nRF24 based technology

Nordic semiconductor wireless technology is suitable for customized applications. Especially within hobbyist circles, the controllers nRF24LE1 and nRF24LU1 are widely used, for example, under the name Grazyradio. The older nRFL01 transceiver is still used, but it needs an extra controller. The top-rated application is Nano Quadcopter. (Bitcraze.io, 2016). These same controllers are used for wireless keyboards and mice.

nRF24 series technology uses the 2.4 GHz frequency area. The maximum bit rate in air is 2 Mbps. With 2 MHz wide channels, there are altogether 62 channels; 39 of them are inside the ISM band. With lower bit rates in air, 250 kbps or 1Mbps, there are 125 channels; 79 of them are inside the ISM band. The frequency hopping method is possible with user software. In addition, the controller includes an encryption/decryption accelerator to build safer protocols. RF communication uses narrow band GFSK (Gaussian Frequency Shift Keying) modulation, the same as in the ANT standard, but includes only the physical and data link layers of the OSI protocol stack. Using Enhanced ShockBurst with the auto retransmit feature, the user interface for the programmer is very simply and small applications were realized with quite light programming (Nordic Semiconductor, 2010b).

3.1.6 Other related technologies

3.1.6.1 NFC

Near Field Communication (NFC) is used for payments, using a credit card or mobile phone. The range is very short; mostly for security reasons. NFC technology, based on RFID technology, forms a ‘handshake’ between two devices that are near each other. RFID uses magnetic field induction in the 13.56MHz frequency band over a distance of up to 20 centimetres. NFC is used in passage control and product packaging. When NFC is installed in a mobile phone, it can read passive RFID tags; act as an active RFID tag or exchange data with other NFC devices. The NFC Forum is a non-profit industry association founded in 2004 to promote the standardization of NFC technology (NFC Forum 2007). (NFC Forum, 2016)

The NFC technology is partly suitable for ambient applications. Passive RFID tags have a very low-cost and are small enough to be implanted invisibly in clothes and the ambience. Mobile phones are very common and simple to use to function as a handheld module in a wireless network. However, it is not possible to create a sensor network using NFC technology.

The disadvantage of NFC is its very short range and the missing intelligence of RFID tags. The tags cannot communicate with each other and function independently, so the features of the applications are limited. In its original use, NFC is quite uncertain and misuse is possible. (Hakamäki & Palomäki, 2015)

3.1.6.2 UWB

Ultra Wideband (UWB) is a new interesting wireless technology. This wireless technology principle originates from 1942. The idea of the original UWB was to send very short data pulses as an unmodulated very wideband low-power signal. This seems to be a good technology for a simple short range, battery powered wireless network. However, in reality it is not. From 2003 to 2006 a group of companies tried to define specifications for standard IEEE 802.15.3a. In 2005 some organizations merged into the WiMedia Alliance, which defined complete UWB specifications to develop solutions. The starting point of UWB seemed to be suitable for a simple, low-cost, low-power wireless network. However, UWB has been developed to replace high-speed Bluetooth and wired USB-connections between a PC and its peripherals. The result of UWB development is that it is no longer suitable for ambient intelligence applications (Aiello & Batra, 2006).

3.2 Comparison of wireless controllers

Electronic circuits give the framework for the features in a wireless network. Comparing radio transceiver chips gives some information about how suitable they are for ambient intelligence applications. There are some main criteria for choosing the best possible hardware for applications. The layout of the electronics must be simple to fit into a small button or a wristwatch. In addition, the power source, i.e. the battery must be small. This means a highly integrated, single chip and ultra-low-power RF controller.

In this chapter, four different radio controllers are compared: Bluetooth, ZigBee, ANT and nRF24LE1. The compared Bluetooth radio controller comes from Atmel: ATBTLC1000 (Atmel, 2016). The ZigBee controller comes from Texas Instruments: CC2630 (Texas Instruments, 2016). The other radio chips come from

Nordic Semiconductors nRF24AP2 transceiver (Nordic Semiconductor, 2010a) for the ANT sensor and the nRF24LE1 controller (Nordic Semiconductor, 2010b) for customized usage. All these use the 2.4 GHz ISM band. The basic comparison data is presented in Table 1. All transmit powers are defined at 0 dBm output. Transmit energy includes the transmission of a 32 bytes payload within a 64 bit frame using 3V power voltage.

Table 1. Comparison between RF controllers

Technology	Bluetooth	ZigBee	ANT	nRF
Controller	ATBTLC1000	CC2630-RSM	nRF24AP2	nRF24LE1
Pin count	32	32	32	24
IO-pins	14	10	7	7
AD (bits)	2 (11 bit)	5 (12 bit)	7 (12 bit)	7 (12 bit)
Size / mm	4 x 4	4 x 4	5 x 5	4 x 4
Power down	0.95 µA	0.15 µA	0.5 µA	0.5 µA
Sleep power	1.35 uA	1,0 uA	3.0 uA	1.0 uA
Active power	0.85 mA	1.95 mA	2.5 mA	2.5 mA
Transmit power	3.0 mA	6.1 mA	15 mA	11.1 mA
Receive power	4.0 mA	5.9 mA	17 mA	13.3 mA
Max bit rate	1 Mbps	250kbps	1 Mbps	2 Mbps
Processor core	Cortex M0	Cortex M3	external	8051+
Transmit energy	2.9 µJ	23.4 µJ	14.4 µJ	6.4 µJ
RSSI	yes	yes	no	no

There are some differences between the chips. The pin count tells something about the complexity of the functions, the demands of the electronics assembly work and the price. A current near 1 µA in the power-down state is not significant, because the self-discharge of batteries is sometimes larger. In the Bluetooth and ZigBee controllers, the processor core is very powerful in comparison with the nRF24LE1 and it is needed to control the complete wireless standard. The ANT transceiver needs an external controller, which takes extra space in small layouts. For the research work described in this thesis, the nRF24LE1 radio controller is used, because the ambient intelligence applications need highly customized features to test separate protocols without limits and the laboratories at Seinäjoki University of Applied Sciences have already a programming environment developed for this controller.

3.3 Internet of Things

The Internet of Things (IoT) is the network of objects exchanging data with each other. It allows objects to be monitored and controlled remotely via an infrastructure network. Every object has an individual IP address to be controlled via the Internet. The IDC organization estimated that the market for IoT solutions would reach 7.1 trillion USD in 2020. The biggest part of IoT is M2M (Machine to Machine) communication. D2D (Device to Device) communication defines peer-to-peer communication between devices which belongs also to the IoT technology (Kalyani & Sharma, 2015).

3.3.1 M2M

The term M2M is widely used by industry. It defines all communication between machines using either wired or wireless methods. If M2M is in a wireless form, it uses low power wide-area network connections (LPWAN) to route or to connect machines together. These kinds of protocols are, for example, BLE, ZigBee or Wi-Fi. The IDC organization estimated that the LPWAN connections would grow to 5 billion by 2022 (Kalyani & Sharma, 2015).

M2M communication is very suitable for the industry. All kinds of monitoring, controlling, servicing, firmware updating and software developing is possible remotely via the Internet. It also opens the world-wide market for smaller companies. It gives great possibilities to centralize different types of know-how and combine different specialists from every corner of the world around one single product.

3.3.2 D2D

The idea of IoT and M2M is that the structure of the network is hierarchical: A higher level in the network can control lower level objects and collect data from them via routers. The device to device (D2D) communication includes also peer-to-peer communication. It means that the devices are not dependant on the network structure, but can exchange data directly with each other without routing. As an example, a cellular connection needs a network structure to work inside a network range, but using D2D, the direct connection between phones is possible without any cellular network (Kalyani & Sharma, 2015).

D2D communication is suitable for mobile objects when the connection to the network is working randomly or is changeable. The wireless range can be much more limited, than in M2M, therefore the devices can be very low-powered and small-sized to be better embedded in objects. The objects are not necessarily visible on the Internet. This kind of solution is, for example, traffic vehicles: as two cars pass each other, they can exchange weather information or problems on the route. RF buttons on wild animals can ‘gossip’ information about travel routes via other animals to a collecting device that is located at the feeding grounds and can then be analysed by experts later. A wireless small-size button in a machine or vehicle can save history information in its memory via D2D connections about its manufacture, service and its usage at factories, service stations and gas stations.

3.3.3 User interface of IoT

Some IoT applications seldom or even never need a human connection. The machines exchange a lot of detailed data and use it in their tasks. Nevertheless, the human wants to be the highest controller of all. At the very least, the system can collect an overview of the system's situation and history to output a report. The human also wants to be able to manage critical controls and task modifications. The IoT devices may have very simple user interfaces or none at all. All control and monitoring activity functions are carried out vie the Internet. The most natural user interface, connected to the Internet, is a computer, laptop or smart phone. Internet-based technologies and devices are not the focus of this thesis.

3.4 Network features

In a low-power wireless network, the nodes are for long periods in the ultra-low-power sleep state. The active communication is always high-power action, as shown in Table 1. Therefore, the sequence of communication must be well defined. First, every message should have a free send time slot without any possible conflicts with other messages, or there must be a method to relieve conflicts. Second, every message must have one or more active receivers to accept the message frames.

Wireless communication differs significantly from a wired system. The receiving state in wired communication is a kind of low-power idle state where the receiver always stays active while not receiving data. Therefore, it is logical that low-power slave nodes are normal in the receive state and reply to a high-power master node, which polls questions to every slave.

In wireless communication, the receiving state is a high-power active state, where the receiving filters and PLL (Phase Locked Loop) are active to sense a possible receiving signal. The receive phase takes as much power or even more than the send phase. For example, the active receiver state of the nRF24LE1 chip takes 13.3 mA whereas the full power transmit takes 11.1 mA. If the receiver does not know the exact send time of a message, it must stay in the receiving state, waiting for the message frame. Usually, the receiver's active time is many times longer when compared with the time of the transmitter. The transmitter is in the active state only as long as it takes to physically send the message frame.

With non-synchronized communication in a mesh wireless network, the receiver must be active all the time, and thus it must have a continuous power supply. In synchronized communication, the receiver and the transmitter can be synchronized based on the same time counter or message sequence. The active receive period can be quite short, depending on how accurate the synchronization is.

3.4.1 Hierarchical topology

The hierarchical topology in a network limits the routing possibilities, because direct communication between slave nodes is not possible. Communication can only take place via a higher hierarchical level.

If the wireless network has master-slave topology, synchronizing is simpler. The master node usually has a continuous power supply and it can be in the receive state for a long time listening for communication from slaves. If the master and the slave have the same speed in their time counters, the message can synchronize the counters. If the slave needs a message from the master, the receive-period of the slave can be synchronized by the transmit time, for example at a defined time after transmission. In the radio controller chip nRF24LE1 the synchronization can be made with the hardware: As the slave transmits a message, it receives an acknowledgement frame from the master, which includes the data payload of the master. So the slave does not need to turn the receiver on with software, because it is done automatically by the hardware for a very short time. With this method, the power consumption of a slave is the least possible for two-way communication.

If the slaves do not synchronize their transmission, the most significant problem is message transmit collision. The slaves could have random transmit timing and a retransmit feature in the case of collision. However, this is not a significant problem in a sparse network if data rate demands are not high: in the radio transceiver chip used in the laboratory, the time of a message frame is about 200 μs and the sleep time to the next communication is about 1 to 4 seconds, so collisions are very rare.

Some methods can decrease the collision possibilities. If the radio modem chip has a carrier detect feature, the software can check that the air is free before sending a message frame. If the slaves have about the same sleep timing, synchronization can settle itself: The collision with the retransmit message delays the start of sleep timing, so the timing shifts automatically in the right direction.

3.4.2 Master controlled synchronization

The collision problem is resolved by full synchronization. All nodes must have time counters with the same speed. In master controlled synchronization, the master acts as the communication controller and transmits to every node the absolute value for the time counter and the non-overlapping time window when the node can be active. In its active window, a node exchanges data with the master and adjusts its time counter and active window time.

There are three main possible ways for the master to transmit its synchronization data to slaves. First: if the slave is in the non-synchronized state, it stays in the high-power receive state until it receives the first synchronization data. This is not the best way because of the high-power period. If the slave does not find the master, it loses its power in a short time and has a very short lifetime. Second: the slave is in short-time high-power receiving states randomly or periodically in the hope that the master sends its synchronization message at just those points of time. The power-up can randomly take a lot of time before the slave is synchronized. Third: the slave sends the request message, until the master replies. The master is in the high-power receiving state. As it receives a request from a slave, which is not in the same synchrony, the slave receives the right timing data in the acknowledge message from the master. After this, both nodes are in the master's synchrony. This is the lowest power and fastest way to find synchronization between nodes.

3.4.3 Clustered mesh topology

The previous synchronization methods are based on small networks, where all slave nodes are inside the range of the master. In larger networks, in which the masters are low-power devices, synchronization is also obligatory. All nodes need sleep time to save energy. In most wireless networks, the nodes form clusters. The nodes inside one cluster are inside the range of a cluster master and synchronized to it.

In the wireless network TUTWSN developed at Tampere University of Technology, the cluster masters are called head nodes. On one side, they communicate with slaves in synchrony with them and on the other side, the head nodes communicate with other head nodes in synchrony with them. In TUTWSN, all nodes are equal, so they can change their roles. The role of a head node needs more power than the role of a slave. The role of the head node can change inside the cluster, so the power decreases evenly inside the cluster. The disadvantage in this kind of clustering is that the network must be located in a limited area, so that the cluster head nodes are inside the range. In this case, all communication is made via the cluster head nodes. Communication between subnodes is not possible (Hämäläinen & Hännikäinen, 2007).

In a larger network, where the clusters are bigger, it is possible to have a cluster tree topology. In this network, every node is a child of a parent node except the first cluster head node. Some nodes in the tree are gateways from a parent cluster to a child cluster. These nodes are also heads of the child clusters. The disadvantage of this network is that the head node of the first cluster is the head node of the whole network and it controls the architecture of the network, so the functioning of this head node is very critical for the whole network. Parent nodes make synchronization with a node in this way: when a node joins a network via a parent, its time counter is synchronized with the time counter of the parent. In this case, all communication is made via parent nodes. The nodes at the same hierarchical level cannot communicate with each other; only via the higher tree structure (Callaway, 2003, ss. 85-113).

In the clustered mesh network, the topology is hierarchical, so some nodes are routers, which are more active and therefore need more power than other nodes. The other nodes communicate only via the routers, but they are very low-power, therefore in this kind of a network only the lowest low-power nodes can be mobile with a limited power source.

3.4.4 Flat mesh

A direct data route between a random source and destination needs a flat mesh ad-hoc topology, where every node can function as a router. In this kind of topology, it is very simple to create new applications without hierarchical limits. However, the router must be in an active and high power state to receive routed message frames, but a moving small node has a very limited energy source. The nodes need a very accurate synchronization method in order to keep the active high power communication period as short as possible, when compared with the long ultra-low-power sleep period. When synchronization is resolved, it makes it possible for a node to catch all the resources in the network in a simple way, regardless of where the resource is and through which route it can be reached. In flat mesh topology the transmit delay can be radically higher than in other topologies, because the message must be rerouted quite possibly many times in time slots following each other. (Palomäki, 2010)

In a wireless network with flat mesh topology, the aim is to function without a hierarchy. Every node can communicate with its neighbour and route messages forward. There are no clusters or head nodes. Synchronization is problematic in flat topology where the whole network is like one cluster and in which all nodes are synchronized with each other. In this case, synchronization and duty cycle optimization are very challenging. The main synchronization problems to solve are the following:

- What defines the global synchronizing time, as initially every node has its own time?
- How can all nodes in a large network be in synchronization, or is it even required?
- How can one single node limit the number of communicating neighbours to limit the active period?
- How to inhibit the formation of multiple groups, which have a different synchronizing time and do not observe each other?

To solve these problems, the features of the network should be more limited. However, the flat mesh network is a universal solution in very different applications in ambient intelligence. Normally, the network topology is optimized for some specific applications, like a sensor network or access control. Therefore, the topology must be redefined when implemented into a new application. By contrast, the flat mesh topology has possibilities to be a universal network structure for all applications that have a low data rate and low security demands.

Global synchronizing must have some rules for defining the base of time in conflict cases. There can be a priority using MAC addresses, so that for example the time of the lowest MAC address spreads over the whole network. Only neighbours communicate with each other, so the synchronization of the whole network must only be so accurate that the neighbours have the same time and are therefore synchronized enough to wake on the same active period. This means that the whole network does not need to be accurate to the same timing: the synchronization can drift through the network, but the nearest neighbours must be in synchrony.

Nodes that are not synchronized cannot even sense each other; therefore, it is possible to have several node groups in the same network that are not aware of each other. There must be a working rule for this case as well. For example, based on MAC addresses different groups have a different standby time and so the timing of the active period drifts so that before long, the timing of different groups collides and the groups find each other.

In a flat mesh network, it is more meaningful to use global hardware message addressing, so that all neighbours can receive the message of a single node. Every neighbour must have its own communication window in a limited active period, so that the nodes can have dynamic tour numbers to fit into the active windows of all neighbours. Another possible way to have a unique transmit window is the contention method, whereby every node is first in the receiving state checking whether the air is free before transmitting. However, in radio communication the transmit time to change the data direction is normally longer than the transmission time, so the contention method cannot function correctly.

In a flat mesh network, the nodes need some memory resources to maintain a dynamic neighbourhood table where all MAC addresses, tour numbers and multihop numbers are updated. In a small network, this table can include data about the nodes in the whole network.

3.5 Routing methods

In wireless communication, where data is to be sent from the source to the destination inside the range area, no routing methods are needed. If the distance between the source and the destination is longer than the range distance, a routing method is needed. The routing method is normally more application specific than synchronization. Synchronization is for the network, but data transfer is for the application. There are two main routing types: first, the data route is opened between two nodes for the data streaming purpose and after the transmit is completed the route is closed. Second, small data frames are transferred continuously from random nodes to other random nodes, the data frame includes a single measure or control data and transfer is made in one sequence without open or close operations. The first type is used in Internet and multimedia communication, where the transmitted data includes pictures, voice or data files. The second routing type is useful in sensor networks and distributed automation.

In ambient intelligence applications, the need for data transmit is usually generated by a random event. For example, when a key is pressed or movement is measured, the event code is transmitted over the network to the node needing this data. There are no long-time data links over the network, but many short-time random data transfers. The second type described above is the only usable communication type in ambient intelligence.

Other features than the ones mentioned above can be classified in other router methods. When classified by network structure the types can be flat, hierarchical or location-based. As discussed in Section 2.1.3, the flat structure is most suitable for universal ambient intelligence applications. The hierarchical structure is suitable for limited applications. The location-based routing is most suitable for wireless sensor networks, but it can give extra utility in ambient intelligence applications. When classified by the route finding method, the types can be proactive, reactive and hybrid. In proactive routing, the data for routes is already collected in the nodes. In reactive routing, the route is found when needed. In hybrid routing, both methods are used (Kamal & Al-Karaki, 2004). In all these methods, one message frame has only one destination. If the found route fails while transmitting a message over the network, the message frame can use broadcast addressing, so that every node inside the range can be reached by it and transmit it forward.

One special and traditional type in active routing is flood routing. The message frames have only broadcast addressing, so the frames flow through the whole network via multiple ways to the receive destination. Unnecessary copies are transmitted and bandwidth is wasted. In gossip routing, the message frames are transmitted to a random neighbour, which relays it randomly as well. This causes delay in data propagation (Hedetniemi & Liestman, 1988).

3.5.1 Required dynamics

In ambient intelligent applications, the objects are moving and they have different incidences with each other. The structure of the network is modified dynamically at any point of time. These modifications can control the events in the network. Modifications also set extra demands for routing, because the data exchange can be interrupted at any point. Every data exchange should be cancelled as a whole if both communication points have not accepted the whole communication event.

Normally the state and events of a network are monitored somewhere, so a multihop routing method is suitable for carrying data via the whole network to the monitoring point. The nodes in the ambience can move continuously, so the fixed multihop data routes work for a very short time. The data transfer cannot be secure in a moving node ambient network. Data loss is actually not an error, but it gives information about the network structure and generates new decisions in the nodes.

3.5.2 Proactive routing

In proactive routing, there are many ways to collect route data and many possible structures for the routing data table. The route data is collected by sending test messages (also called agents) through the network. In addition, during normal communication the messages can function as agents, finding routes and writing down link data when routing through the network. In a power aware network, it is important to keep all nodes working, so that the table describing routes can control data routing via those nodes, which have more energy. In practice, it means that every link has a cost value. The routing method seeks the cheapest route from source to destination, bypassing the nodes with too low energy to function as routers (Hac, 2003, ss. 102-108).

Proactive routing is suitable for a fixed network with nodes that have fixed coordinates and some memory resources. The routing tables in the nodes have a long life-time. If the nodes are moving, the routing table has faulty data, producing delays or failures in communication.

3.5.3 Reactive routing

Reactive routing is suitable for mobile wireless networks. It is not possible beforehand to know the shortest or cheapest route. In reactive routing the route is looked for only if needed. No long-time route data is written in node memory. The agent message, finding route links, is very short and does not need much energy. In addition, the message frame with its payload can action itself as an agent finding a route through the network (Hac, 2003, ss. 102-108).

Reactive routing is suitable, when the nodes have very limited memory size, and in a mobile network. A single node must save only one route data, which can be overwritten by the next route seeker. The communication capacity is also low, due to the seek procedure used for every message routing.

3.5.4 Flood routing

In flooding routing protocol, there is no routing data to use and a message frame has broadcast addressing, so every node inside the range can receive and re-route it. In an uncontrolled case, this routing can jam the whole network. To prevent this, there are some methods in order to decrease needless communication:

- Loop: a single node reroutes a specific message frame only once to block loop routing. Every frame includes its own identification number.
- Distance: the message frame is not re-routed, when it travels via enough multi-hops, to reach all nodes in the network. Every frame also includes a hop counter.

- Direction: if the positioning method is in use and the nodes have some distance data, the message frames travel only in the direction, where the distance to the destination is shorter or the same. The frame includes a multi-hop counter, i.e. the distance to the source node.

The flooding routing method needs a lot of communication resources for the network, and therefore it is only suitable for low-speed, simple networks with high-energy nodes or if the network has background communication already for positioning or synchronizing purposes.

4 INTELLIGENT SYSTEMS

4.1 Artificial intelligence

Artificial intelligence (AI) is a large concept including, for example language processing and soft computing. AI copies the information processing methods from nature: spoken languages, nervous system, insect swarm, plant and animal tissue etc. The most technical concept of AI is soft computing (SC), including machine learning, neural networks, fuzzy logic, genetic algorithm, swarm intelligence and many other computing methods. One concept for soft computing is shown in Figure 14.

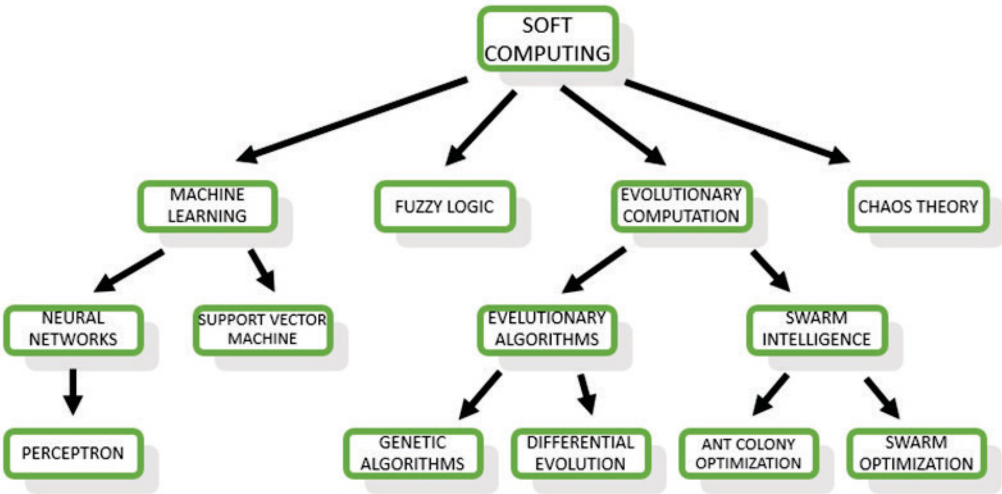


Figure 14. Soft computing concept

One of the most challenging tasks for computing is to model physical systems. The controller system can estimate and predict the behaviour of the physical system by modelling, and can keep it under control in critical situations. For example, it is not reasonable to test the critical behaviour of a nuclear power plant in practice, but by using modelling, it is possible to simulate critical cases without risk.

The dynamics of physical systems could be modelled mathematically by differential (deference) equations. The accuracy of such models depends on the complexity of the physical system. Almost all mathematical models have limited accuracy and no model can catch all physical system behaviour in all operational conditions. Modelling errors usually occur due to the uncertainty and the stochastic nature of real systems. Also, there are some real systems that are very complicated and therefore, it is not possible to build a reliable and accurate mathematical model. Soft computing methods (or black-box modelling) can be more useful in such cases. Depending on the measured data and the model complexity different modelling methods can be used, as presented in Figure 15.

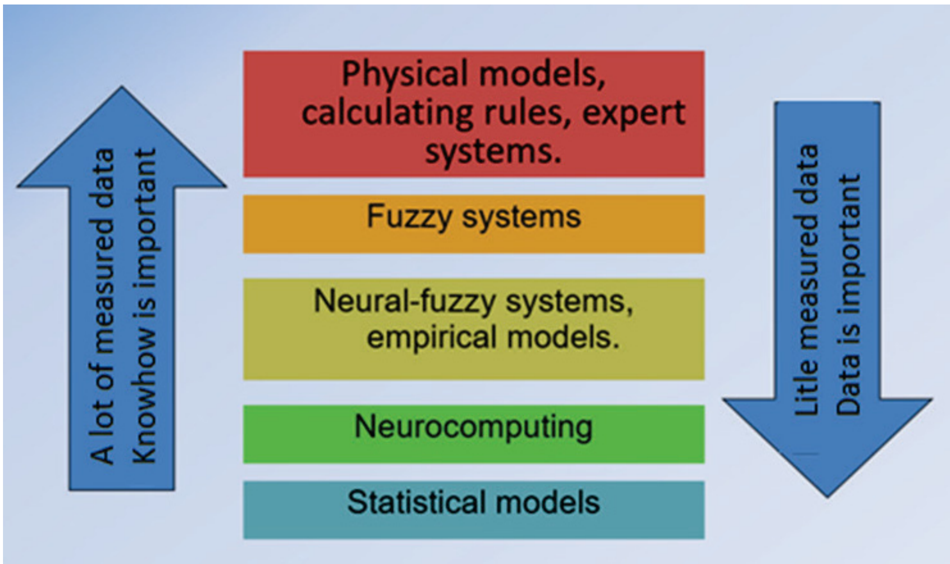


Figure 15. Modelling methods

4.1.1 Neural network

The idea for the neural network comes from the nervous system of animals. The activities of a neural network are divided between many similar neurons. A single neuron reads signals from selected neurons and calculates its own value with simple rules for signalling to other neurons. The first layer of neurons uses input signals measured from the ambience and outputs the results to the next layer. The next hidden layer uses the output of the previous layer as input and calculates values forward to the next layer. The output of the last response layer is the output of the complete neural network. One example is shown in Figure 16. The simplest neural networks include only one response layer (Graupe, 2007).

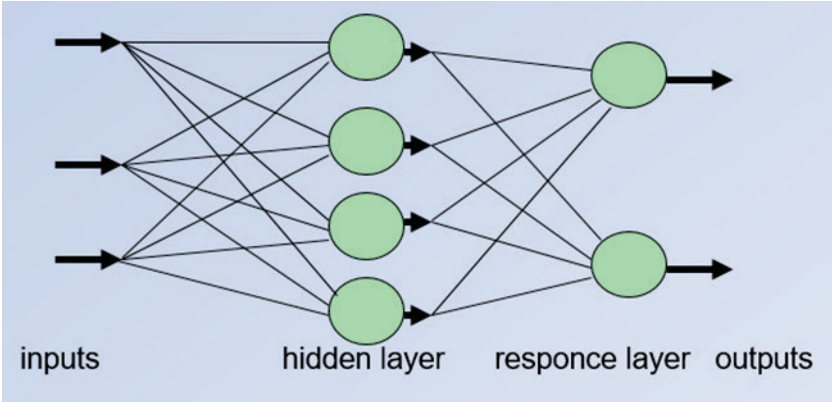


Figure 16. Neural network with one hidden layer

The most interesting feature in a neural network is self-adapting: The learning process uses a large amount of learning material as example input-output pairs measured from the real system. The comparison between the learning material and input-output values of the neural network modifies the connection weights between neurons. After executing this comparison in numerous iteration loops, the neural network finally generates real output signals from the inputs based on the learning material (Figure 17). This way the neural network creates a model of the real system. This is the most usable neural network usage: to create models of physical systems (Graupe, 2007).

The most interesting feature in a neural network is self-adapting: The learning process uses a large amount of learning material as example input-output pairs measured from the real system. The comparison between the learning material and input-output values of the neural network modifies the connection weights between neurons. After executing this comparison in numerous iteration loops, the neural network finally generates real output signals from the inputs based on the learning material (Figure 17). This way the neural network creates a model of the real system. This is the most usable neural network usage: to create models of physical systems (Graupe, 2007).

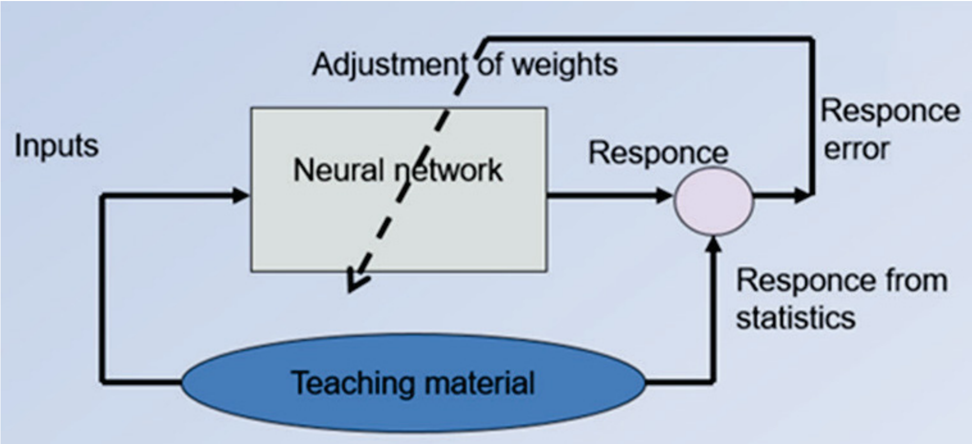


Figure 17. Neural network learning process

One interesting usage of the neural network is the prediction application to predict some value varying with time. In this case, the input of teaching material is historic values and the output the current value. The input of the working neural network is historic values including the current value. The output is the predicted value as seen in Figure 18.

In the prediction application, the learning process from history and predicting can continue in parallel at the same time. Therefore, over time, it learns to predict still better.

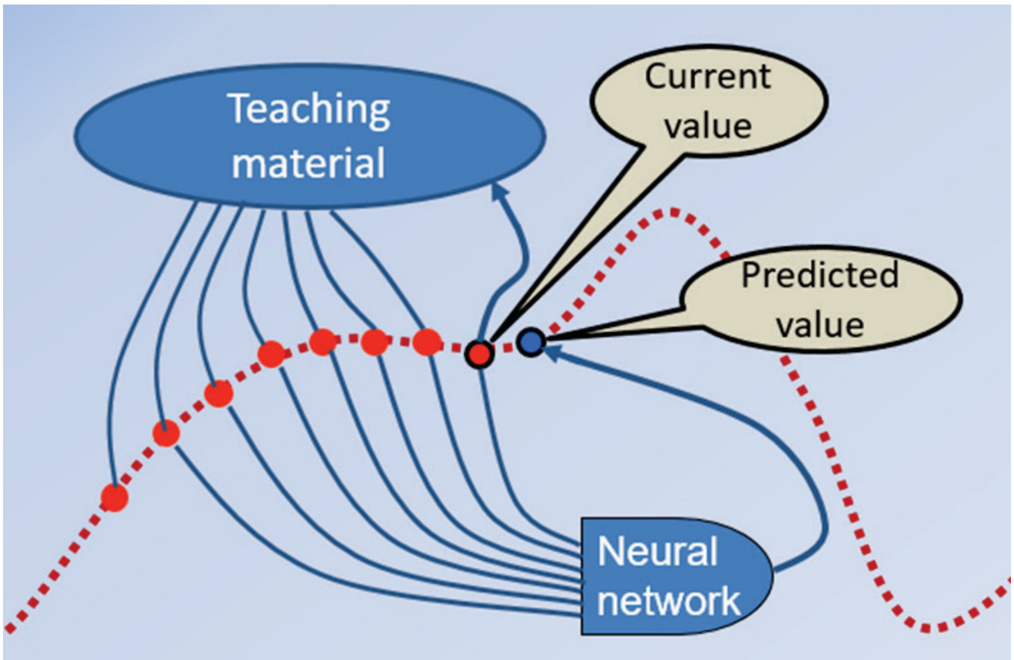


Figure 18. Prediction with neural network

4.1.2 Self-organization

However, a neural network executes the calculation in one direction: from input to output (or visually from left to right). In some applications, a better way is to organize a neural network as a self-organized map (SOM), where the self-adapting or self-organizing depends on the neighbourhood (Kohonen, 2006). It is supposedly possible to organize and control the role of a single node in a wireless network that has a global job or aim. Using SOM, the nodes organize themselves logically and physically.

Self-organization of social insects means global structure modifications using local information. For example, a single bee can direct most bees in the hive to a food source with a higher sugar concentration using positive and negative feedback. Self-organization itself happens though the amplifications of random events (route choosing, task switching and so on) that other individuals choose, thus changing global behaviour. (Bonabeau;Dorigo;& Theraulaz, 1999, ss. 9-14)

The digital solution of self-organization can be fault-tolerant distributed automation, consisting of a group of identical nodes. The task of the whole group contains numerous small tasks, which a single node can execute. The network has a set of reserve nodes, which means that there are more nodes than needed for the global task. Some nodes execute unimportant optional tasks or they are free. These nodes try to find possibilities to change their task to a more important one. If a specific data source node fails, the continuous data questions without a response are a positive response to a free node to get the job of the missing node.

4.1.3 Fuzzy logic

A traditional proportional-integral-derivative (PID) controller is suitable for controlling linear processes and systems. If the process behaves non-linear, it includes many near-linear slices. These slices need different individual controller rules (Figure 19). At the borders of the slices, the controller can use two different rules alternately thus causing instability in control. Fuzzy logic means that borders are not exact. Control can be a combination of two rules weighted by the nearness of the corresponding rule as shown in Figure 20.

Figure 19. Exact control principle of non-linear process

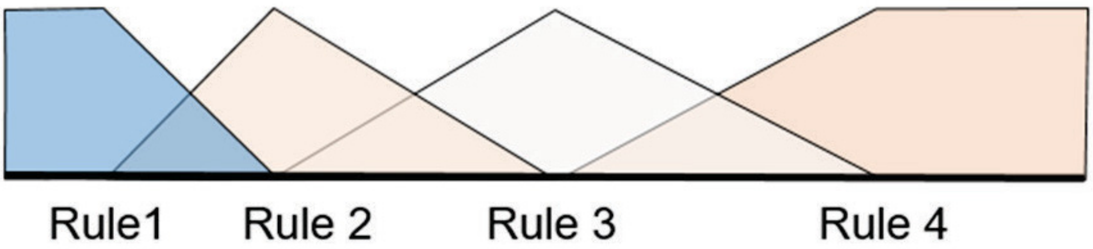


Figure 20. Fuzzy control principle

For controlling a non-linear or even an unknown process or system, the rules can have a verbal form based on experiences. For example, "If I feel cold, the heater must work at maximum power" or "If I feel comfortable, the heater can work at half power". The fuzzy logic needs the words "cold", "comfortable", "maximum" and "half" in numeric format to add corresponding rules in the controller. In more demanding applications, as in aeroplanes, there can be tens or hundreds of rules in one fuzzy controller. In this kind of application, the rules are no longer verbal, but are generated from the measurements of numerous test cases. (Niskanen, 2003) includes more material about fuzzy logic.

4.1.4 Interaction in a social insect swarm

The behaviour of some insects has been researched widely. In addition, some rules and signalling methods have been found to have a mathematical and digital format. The most researched insects are ants and bees, because they live in large social swarms and they have a good order in all functions when interfacing with each other and with the ambience (Bonabeau;Dorigo;& Theraulaz, 1999, ss. 1-23).

Typically, social insects use direct or indirect interaction. The direct interaction is different contacts and material exchanged. The indirect interaction is modifications and chemicals in the environment. These interactions stimulate events and organize the functions of every individual (Bonabeau;Dorigo;& Theraulaz, 1999, ss. 14-16,26).

The digital solution of this interaction could be an exergame with small RF buttons located in the environment. Hand-held radio modules can communicate with these buttons, thus saving marks and loading virtual material to and from the button or between the hand-held modules.

In the ant nest, the ants can change roles depending on, for example, the conditions inside the nest or by noticing missing jobs (Bonabeau;Dorigo;& Theraulaz, 1999, ss. 109-148). In distributed automation, fault tolerance is one feature where role exchange is possible if required. For example, a fault in one sensor node causes sensor data to be missing which activates the alternative sensor nodes to take the role of the missing sensor. In exergames, the role of the handheld module of one player can exchange its role by using data exchange with another by changing its internal resource or by the user's strategy decision.

4.1.5 Ant algorithm

Route optimization is one method copied from ant behaviour. Normally, when ants are carrying food they deposit a chemical called pheromone. Random routes between the nest and food are marked with pheromone. The shortest and fastest route soon has more pheromone than the other routes. Therefore, the ants can choose the most frequently used route shown in Figure 21. Pheromone has a limited lifetime, so only the shortest route has pheromone and it is soon the only way used between the nest and the food source (Bonabeau;Dorigo;& Theraulaz, 1999, ss. 26-31).

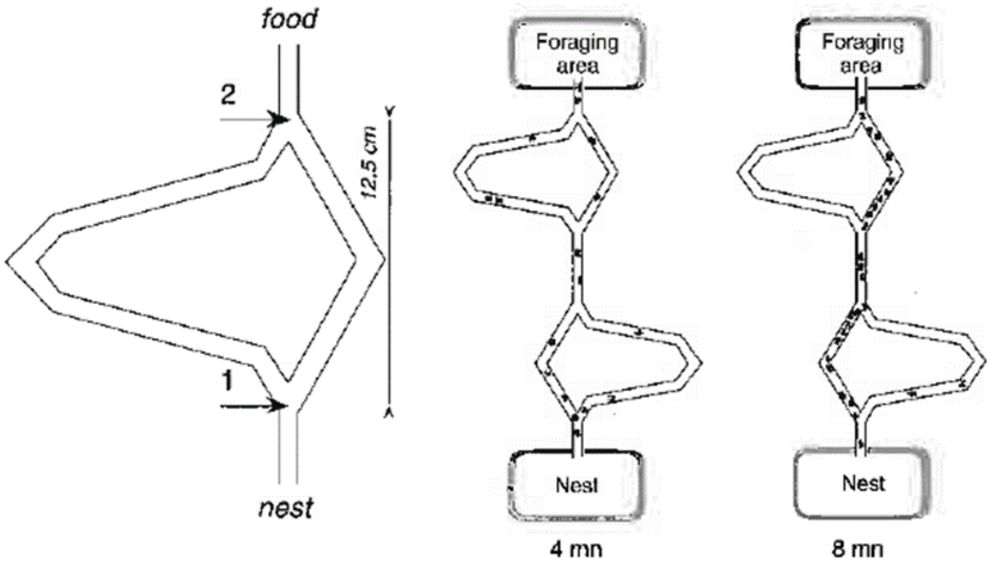


Figure 21. Ant Algorithm (Bonabeau;Dorigo;& Theraulaz, 1999, ss. 26-31)

The digital solution of the ant algorithm could be usable in proactive routing, in which the agent messages travel via a random route between the source and the destination, leaving "digital pheromone" in the nodes. Then the most marked route is the shortest route which the message frames with a payload can use.

4.1.6 Amorphous computing

Amorphous computing is the interaction of a huge number of identical individuals. It is also called cellular computing, and it describes well how the cells in a biological organism interact. A single cell has chemical contacts with its closest neighbours. However, signal wave propagation is possible through the whole organism, because single cells can route chemical signals forward. When simulating biology amorphous computing with software, the program or rules in all the cells are identical and very simple. The results or behaviour of amorphous computing is very interesting and complex. It is usually depicted graphically with pictures, where one dot describes one computing cell (Abelson, ym., 2000).

In ambient intelligence applications, amorphous computing is the base structure of the network. A single node has a limited range to communicate with some of the closest neighbours. The messages flow using multi-hops through the network with the flow routing method. In positioning and passage-control applications, the nodes are identical, measuring distances by the number of multi-hops and by interacting with every node, which are inside the range. Amorphous computing is a very important model when realizing the wireless ambient intelligence network structure.

4.2 Interactive behaviour

The role of a neighbour significantly controls the behaviour of players in a wireless strategy game, depending on whether the neighbour is an enemy or a friend. In addition, the role defines which data exchange is executed; whether a neighbour has data which this node is collecting or sensor data to be relayed forward.

4.2.1 Distributed intelligence

With traditional technology, the cheapest and clearest way is to centralize the computing power. However, it has some disadvantages that can be critical in demanded applications. A single failure can destroy a promising space programme, or a small software error can stop a huge production line. Furthermore, in many applications it is not possible to have centralized computation power due to several different reasons such as cost, lack of resources (like limited data transfer), large latency and huge computation power needed. As an example, assume a certain large distributed system is being controlled to achieve predefined objectives. There are thousands of embedded systems (sensors + actuations) distributed in the system. For centralized computing and control, all of these thousands of sensors should send their data (within a sharp delay limit) to the central processing unit. Furthermore, all actuators should receive the response within the delay limit. It can be very complicated (sometime impossible) to achieve this system technically. Therefore, it is much more desirable (in terms of feasibility, cost, and reliability) to distribute the data processing and intelligence among all embedded systems. It seems that in the future all critical computing and communication can be made with very distributed intelligence.

4.2.2 Reserve resources

The key feature of distributed computing is fault tolerance. If one node in a wireless network (i.e. one unit in a swarm) is destroyed, it can be replaced with another equal unit. The key question is, are there equal reserve units in the network? In a centralized system, the control device is unique and the equal reserve unit must be as large as the original system. In a distributed system, every separate job has its own control system. But the control systems are equal having unique software because of the unique job they execute. Therefore, the number of reserve systems can be lower when repairing a single failure; however, the service man must download the unique software into the replacement unit. This means that the smaller the units are, the smaller the reserve resources can be. In bigger units, the software must be loaded manually, but in smaller units, new software can be loaded automatically via a wireless network; therefore, the system can repair itself. Some units connected to the ambience via sensors or actuators can be exceptions; they must be replaced by connecting manually or there is a sensor and actuator reserve as well.

One interesting question is, how small must the units be, to be equal? A single small reserve unit is enough, if preparing to repair a single failure. A single destroyed unit can be automatically replaced by an equal reserve unit, which downloads its software via the network, i.e. it takes over the role of the destroyed unit.

Every system normally has interface signals to the ambience, so a single unit must have at least one input or output. One interface signal is enough if the direction is software selectable. One single unit in a distributed intelligence system also includes a power supply, a controller, a radio transceiver and an interface signal defined by software. The power supply can be a battery charged by a solar cell or by some other ambient energy source. Energy can also be loaded from a control signal from the sensor or the actuator.

In an ambient intelligence application, the nodes are interfacing with each other and with the ambience. The communication is interactive but does not include much data. The main purpose of communication is to start events and to fix the direction of functions. The same functions are in use in natural insect swarms, so the best rules and examples are found in nature. The example of the social behaviour of insects has produced some soft computing methods, such as self-organization, amorphous computing and neurocomputing.

Another type of modification takes place if an active node with a significant role in the whole network is destroyed. The job of the network is then executed incompletely and it is really an error event. In automation, these failures are critical, because every sensor and actuator has a significant meaning in the control loops. In natural cell tissue, or in social insect nets, there are a lot of reserve resources to replace damaged elements. In ambient intelligence and in distributed automation, it is also a big advantage if a network has reserve resources to replace the destroyed nodes and their functions.

4.3 Positioning methods

The functions and behaviour of a single node can depend on its position. The positioning can be physical or logical. In physical positioning, the node knows its own coordinate values. In automation, the node can identify the measured data or actuator values in a specific room or area. In logical positioning, the node knows its neighbourhood, which nodes it is interfacing with and can behave accordingly. (Palomäki, 2007b) (Palomäki & Kivikoski, 2008)

The two main types of positioning are physical and logical positioning. In both positioning types, the most important function is the distance measuring method between the nodes. In physical positioning, the distance is measured by dimension numerical values. In logical positioning, the distance is measured by the neighbourhood. In the hybrid positioning method, the neighbourhood estimates the numerical distance values. The absolute positioning method needs fixed anchor nodes with constant coordinates as reference points for the coordinates. Without reference points, the coordinates are relative and the positions can be mirrored related to the physical positions. A working 2-dimensional positioning method needs three or more anchor nodes in fixed positions to fix the positions of nodes.

4.3.1 GPS

GPS (Global Positioning System) is the most commonly used satellite based positioning system. In this thesis, the target is universal ambient intelligence application technology, so positioning must work both outdoors and indoors. However, indoor node positioning is not possible with GPS (Mäkelä, 2008, ss. 73-78). In addition, GPS positioning uses an extra module, which increases the node size and power consumption.

4.3.2 Acoustic

The distances between nodes in a wireless network can be measured using ultrasound propagation delay (Savvides;Han;& Strivastava, 2001). The idea is based on switching on a RF (or light) signal and ultrasound signal at the same time. The timer in the receiver starts counting the time required from when the RF (or light) signal has been received until the ultrasound signal is received. The speed of sound is known, therefore, the distance can be estimated. Since RF (or light) travels at the speed of light we can ignore the propagation delay. However, this method needs acoustic components and driver hardware, so it increases the node size and power consumption.

4.3.3 RF strength and transmission time

A simple way to measure distances in positioning is to use RSSI - Received Signal Strength Indicator (Reghelin & Fröhlich, 2006) (Patwari & Hero, 2003) (Liu;Ning;& Du, 2005). This method needs extra hardware, but many radio transceiver chips have this integrated, as Table 1 shows. The smallest transceiver used in this research does not have this feature, but it is an alternative method in the future. Another radio signal based method is measuring the transmission time (Nasipuri & Li, 2002). This method needs extra very specific hardware and calculation power, which are costly and power demanding technologies and are therefore not usable in a very light wireless network.

4.3.4 Neighbourhood

The simplest way to measure the position data of a single node is for it to identify its neighbourhood and estimate its own position coordinates within it (Patwari & Hero, 2003) (Costa;Patwari;& Hero, 2006). The closest neighbour distance is estimated using the range of one transceiver. The distances to farther nodes are range multiplied by the number of multi-hops. In addition, the number of multi-hops can be based on the network topology, not the real distance (Ray;Lai;& Paschalidis, 2006) (Wu;Wang;& Tzeng, 2005). In this case, the positioning is more logical than physical and depends on the routing method. These are the most suitable existing positioning methods for a limited resource wireless network.

Neighbourhood positioning is suitable in dense networks, where the ranges are quite short. In this case, the number of multi-hops is high and the positioning method has good source data for position estimation calculations. In coarse networks, the positioning accuracy is not very good; however, depending on the application, it can be good enough to identify the individual object carrying the node in the network. (Palomäki, 2007b) (Palomäki & Kivikoski, 2008)

4.3.5 Other positioning methods

Plenty of other methods for positioning nodes in a wireless network are available. The Star-Dust framework (Stoleru;Vicaire;He;& Stankovic, 2006) uses image processing and optical reflectors to position wireless nodes. This can function in a limited indoor area, but not outdoors, where dirty ambience can inhibit the functions. In the Spotlight system, the nodes must be visible to a spotlight device (Stoleru;He;Stankovic;& Luebke, 2005). The system is not suitable for exergame (Outdoor exercise games controlled by information technology) for example in a forest.

5 CHALLENGES OF DISTRIBUTED AUTOMATION

5.1 Ambient intelligence

Ambient intelligence applications differ from industrial automation in information technology. The ambient intelligence devices are usually embedded in the ambience and do not need any user interfaces.

5.1.1 Small size

A radio button installed in a wristwatch can for example monitor the behaviour of demented people without a user interface. The physical presence of wireless devices should be as invisible as possible. Therefore, the nodes should have a very small size without any connecting wires, thereby allowing freedom of movement. In many applications, the wireless button is in hard external conditions, such as in garments or in animals, so the button must also be waterproof and have a shockproof box.

5.1.2 Low power

In connection with small size and compact structure, the power supply must also be small and as service free as possible. This means that the radio button should have very low power consumption and use a small battery or a micro power energy-harvesting feature. In addition, communication protocol must support the low-power feature. The receiving state in wireless communication is a very active state, so the radio button cannot spend a long time listening for others. The button must be the active part in communication deciding when to transmit a message and waiting a very short time for a response message. When comparing wireless electronic power consumption, two criteria are important: the current at low-power or deep-sleep time and the energy per bit of a transmitted message.

5.1.3 Interfacing

The nodes in ambient intelligence interface with each other in one way or another. They can sense the nearness of others, identify the role of a neighbour and behave accordingly. The nodes can move randomly, and so the structure of a network can be modified dynamically. Traditionally, modifications in the network structure are exceptions needing a special setup period. In ambient intelligence, fast modifications in the network is a normal working method and include specific information about events going on.

A single node requires the defined interfacing features described above. The radio modem should have Received Signal Strength Indication (RSSI) to measure the distance and to select its relationship to others, or the range should be short enough to sense the nearest neighbours directly as well as the next neighbours using the intelligent multi-hopping routing method. For relative positioning purposes in a network RSSI or an equivalent feature is needed to measure distance between nodes.

5.1.4 Open source

For connecting small objects wirelessly together and for monitoring them the applications used should be as simple as possible. In addition, developers should be able to make very special solutions not included in current standards. For these reasons, all electronic connections and software sources should be open. The best user interface is simple receive and transmit routines so that one node can tell to the network: 'I'm here' needing no any complicated functions. If the creation of a new application does not need deeper knowledge about protocol stacks etc., there are huge application developer resources, including students, hobbyist and small companies.

5.2 Standard versus non-standard

In developing, a new wireless mass product with demands for new features and a small size the best tool for this application is not necessarily a standard wireless solution. It can be tempting to implement new features into existing standards but the solution is not optimized, because the standard has many extra features that are not needed. The best result is reached by fully customized technology developed only for this one application. A big disadvantage is the costly development work made only for this project. The best solution lies between these two (Palomäki, 2008, ss. 10-11).

5.2.1 Need – toolbox – running fees

The alternatives become clearer by a parable with Figure 22. If you need to fix a screw with a screwdriver, you can ask a service man to give you a single screwdriver. Perhaps your organization does not have single screwdrivers, but whole toolboxes with all the necessary tools for standard jobs. However, the toolbox is too heavy to carry, so you need a truck to take it to the work place. A large amount of work has been performed for a small need.

If these objects have real names in a wireless application, the aim of this parable becomes clearer. If you need to use wireless technology in a simple new application, the screwdriver means the technology needed in your application. The toolbox means extra investments: you need to start the development with a standard technology with some kind of a development kit and software. It takes extra costs to fit this standard into the hardware in your product. The truck means running fees: license, support and update fees, which you need to get a right to use for the newest standard of your product. As shown, in many applications standard wireless technology is too troublesome to use and to fulfil real needs.

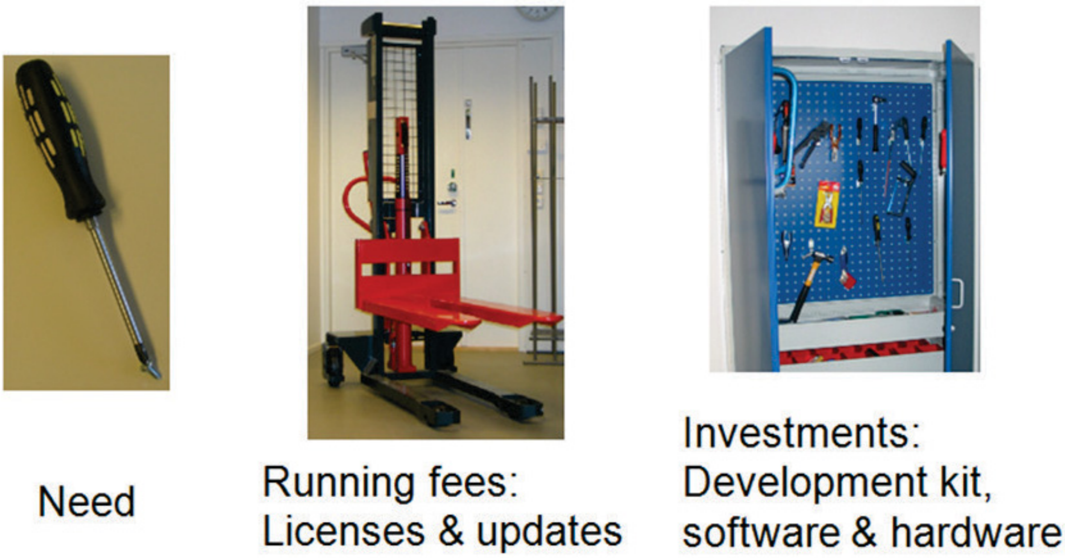


Figure 22. Solution with standard tools

5.2.2 Life cycle of technologies

When a standard is created, the features fulfil the demands of the day and the predicted future demands. The quality of the result depends on how good at predicting the creators are. It also depends on how big redirections (modifications) are possible based on the original standard. If the cap between the demands and standard grows large enough, the life cycle of the standard is at an end. This means that the life cycle of a standard is always limited (Figure 23 a).

New technology can be made in a protected customized technology development project inside a company. Wireless technology is made for specific demands or even for a single product. The development investment is quite high for every new product and for every new development cycle. The life cycle of this technology is very short, that is, it is as long as the lifecycle of the product. Different companies make the same development work. If that kind of protected customized wireless technology is the only alternative for standards, then the standards are, in practice, the only reasonable choice (Figure 23 b).

The third alternative development style lies between the two alternatives discussed above. This is open customized i.e. non-standard public wireless technology. When the base technology is first created with a larger investment, the next developers continue from the point the others have reached. Thus, the investments are much lower in each development period or product. The problem is that every developer creates work that is free for others. This is not usually the principle of companies, which must live through profit. The key operators in this kind of open development are universities and individuals who are interested in development. The life cycle of this kind of technology development is very long, because the next developers can continue the development and make as big a modification as possible to fit the demands of a new application (Figure 23 c).

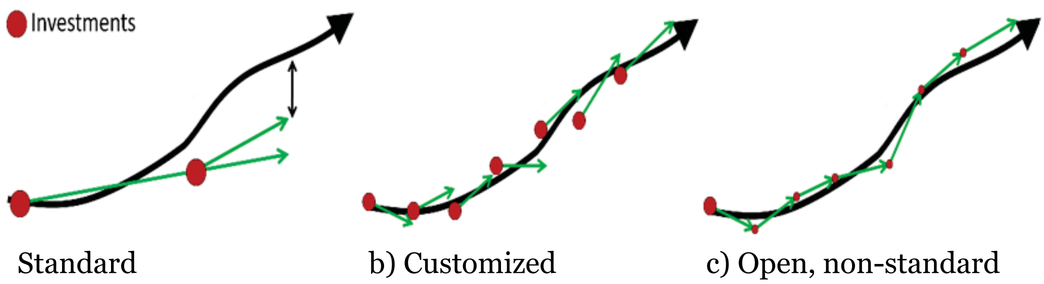


Figure 23. Life cycles and investments of technologies

6 IMPLEMENTATIONS

Seinäjoki University of Applied Sciences has invested in future technology. The electronics laboratories make is possible to research new technologies to be used in new applications. Near Seinäjoki there are many small companies which use mostly traditional technologies. The researchers have a vision that in these traditional enterprises new technologies can be realized, if the technology is simple enough to use, invisible and embedded in systems.

Our first project was called “Realizing New Technologies”. The principal aim of this project was to create innovative cooperation between the laboratories and companies. The second aim was to create possibilities for new projects to research and develop new future technologies to fulfil future demands. This project was evaluated by Tampere University when they researched the regional effectiveness of Universities of Applied Sciences (Marttila;Andolin;Kautonen; Lyytinen;& Suvinen, 2007, ss. 35-40).

From this base, the vision for this thesis research has arisen to develop technology for ambient intelligence applications which need low-power and low-cost networking wireless technology in the form of small buttons and modules. The applications are, for example, goods positioning and identification in factories and storages, access control and security monitoring of children, demented old people and animals. Exercise games are also a new fast growing market area for wireless technology.

6.1 Scheme of wireless project

The scheme and flow of the wireless technology research and the development project is described in Figure 24. The first alternative technology is to use wireless standards, which are already developed protocols with fixed features. The aim is to find new features and new possibilities and therefore non-standard technology is required. The non-standard wireless technology is normally fully customized and the features are developed for specific products in a company; this is not the target of this development, it needs wireless technology made for universal use to develop universal and open wireless tools for applications under development.

The research project produced the first realization in 2005: with the first simple wireless hardware including drivers for it. The wireless module had the form of an extension module that was compatible with the embedded systems teaching set. With this solution, it was possible to develop the software with universal and familiar tools (Palomäki, 2008, ss. 35-41).

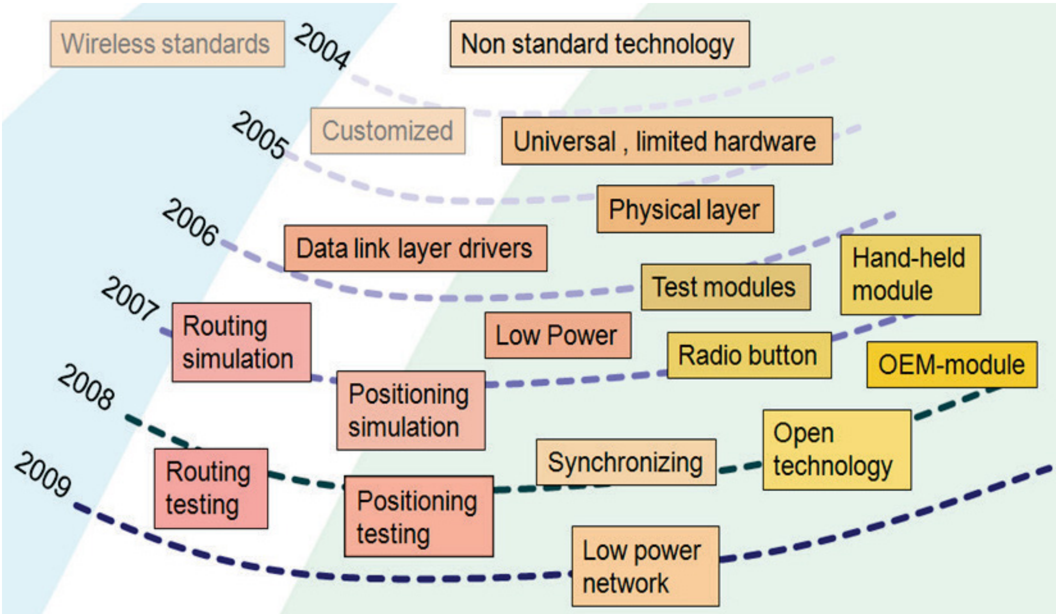


Figure 24. The scheme of the wireless development project

In the years 2006-2007, new extension modules were also made as stand-alone modules to test the technology for new ways of using it in ambient intelligence applications. These were a key chain module, a small RF button and a hand-held module. At the same time, simulating the routing and positioning features was started for this limited-hardware wireless network. In addition, a universal plug-in OEM (Own Electronic Manufacturer) module for custom projects was made.

In 2008, testing the technology and researching the synchronizing, positioning and routing methods began. Some of the results are available on the Internet. The aim was to have a working universal ultra-low-power wireless network using very limited hardware.

6.2 Development principles

The two main goals in hardware development are low-power and a small size. These features are essential to find technology for new applications in an ambient area, where the number of units is large and they can be mobile. Wireless units need e.g. a high-capacity battery and/or energy harvesting means such as a solar cell to allow for a long working life. However, radio communication needs power, and so the low-power feature means an ultra-low-power standby period with a very short high-power active period. A small size means small components, simple circuit connections and a few external passive components.

6.2.1 Selecting chips

To illustrate the hardware selections for our research team, the three RF transceiver chips that are most suitable for the chosen demands are introduced here. The layout of the electronics must be simple to fit into a small button or wristwatch. In addition, the power source, i.e. the battery must be of a small size. That means highly integrated and very low-power RF-transceiver and controller chips. These chips are compared in Section 3.2. The microcontroller chip has the same demands: small, simple to connect, include features to make synchronization and have an ultra-low-power standby period. The program memory capacity must also be big enough to include transceiver drivers, synchronization, routing and application software. In addition the controller should be compatible with bigger controllers that are used in the hand-held modules and data collection USB sticks. If so, the software exchange is simpler between different modules, which use the same RF transceiver and the protocol stack. For this purpose, the Atmel Corporation 8-bit RISC controller family was chosen, where the smallest used controller is ATtiny84 (Atmel Corporation, 2008).

6.2.2 nRF24L01-based electronics

All kinds of intelligent devices are possible if you have the required resources of time and money. To develop wide-to-use, low-cost, small-size and simple-to-use ambient intelligent devices with limited resources is challenging. The two most important key words are ‘simpler’ and ‘simpler’ when developing new small and distributed intelligent devices. Simpler commercial components result in lower-costs. A simpler structure means a smaller size and fewer failure risks. A simple user interface results in nearly invisible usage. However, this simplicity should not reduce the efficiency, performance, reliability, and all of the required functionalities.

One example for a minimized circuit connection is presented in Figure 25. This circuit includes an nRF24L01 radio transceiver, an ATtiny84 microcontroller and some passive components.

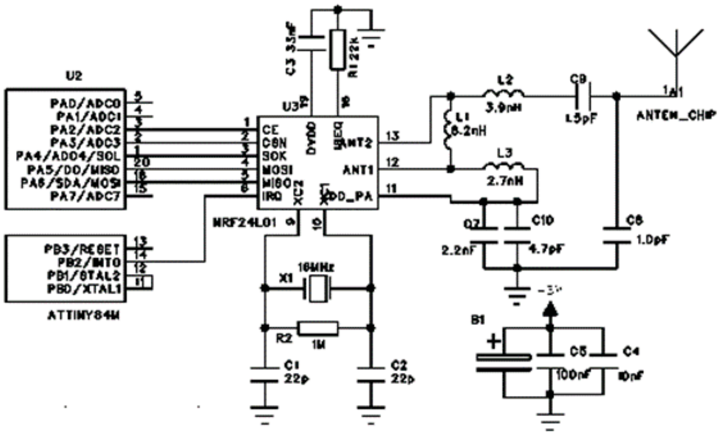


Figure 25. Minimized connection schema

One connection schema can have many layouts (Figure 26). Two of them have been realized in an RF button (23.5 x 17 mm) and a plug-in OEM module (22 x 10 mm).

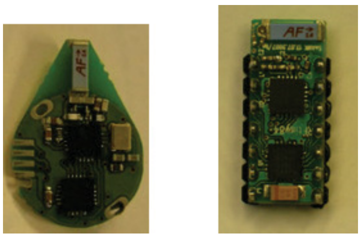
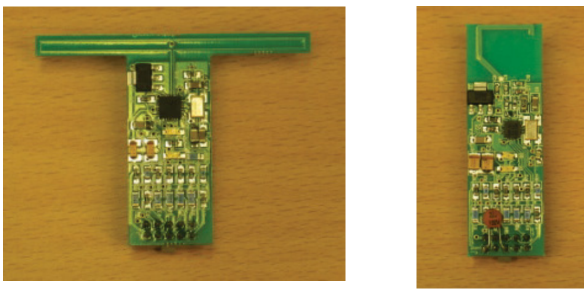


Figure 26. Minimized layout variants

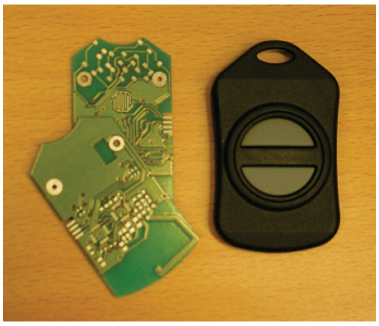
The first assembled test module uses the nRF2401 chip as shown in Figure 27a and Figure 27b shows the newer version using the nRF24L01 chip.



(a) (b)

Figure 27. The optional modules of the teaching kit

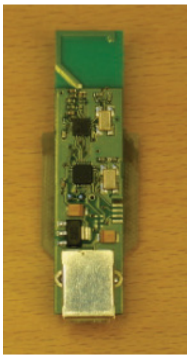
The test modules mentioned above were made for testing and software development purposes. For real applications a key chain module was developed with a simple 2-key user interface (Figure 28 a) as well as a hand held module with a small graphical display and rotary encoder input (Figure 28 b).

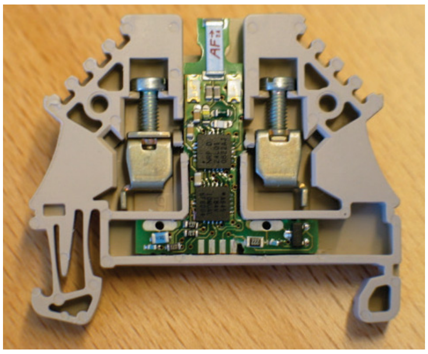


(a) (b)

Figure 28. Key chain module and hand-held modules

A USB stick was designed and made, which is connected via a USB port to a personal computer, to collect the events in the network into a database, such as the measured data and node positions. (Figure 29 a).





(a) (b)

Figure 29. USB stick and DIN rail connector

A smart DIN rail connector was also designed and made for distributed automation measuring and controlling purposes with standard 4-20 mA analog signals as shown in Figure 29b. They both have the same RF transceiver chip but a different controller with a specific interface.

6.2.3 nRF24LE1 based electronics

The use of the nRF24LE1 controller is more suitable for ambient intelligence applications. The chip includes an earlier nRF24L01 transceiver and a full microcontroller with seven I/O pins. Every pin can be used as an A/D channel with 12 bit accuracy. It also includes some serial interfaces. After taking the nRF24LE1 into use, the older transceiver above was no longer used. The connection schematic is now simpler, with only one chip as shown in Figure 30.

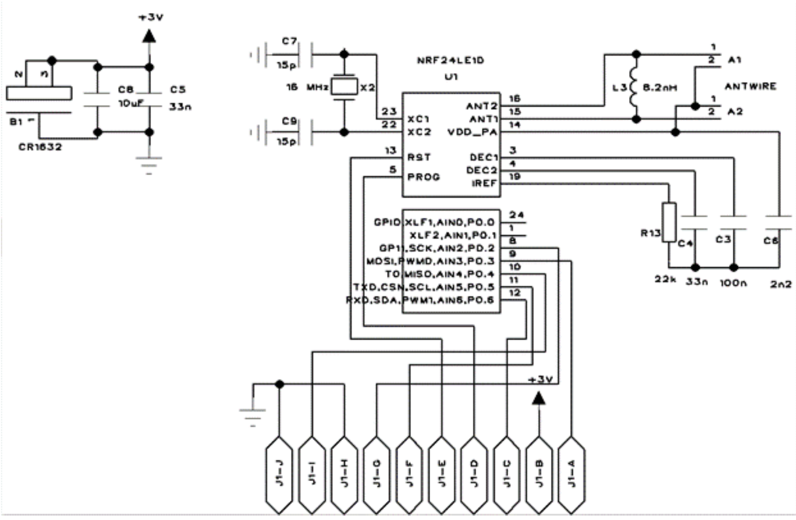


Figure 30. nRF24LE1 based node connection schema

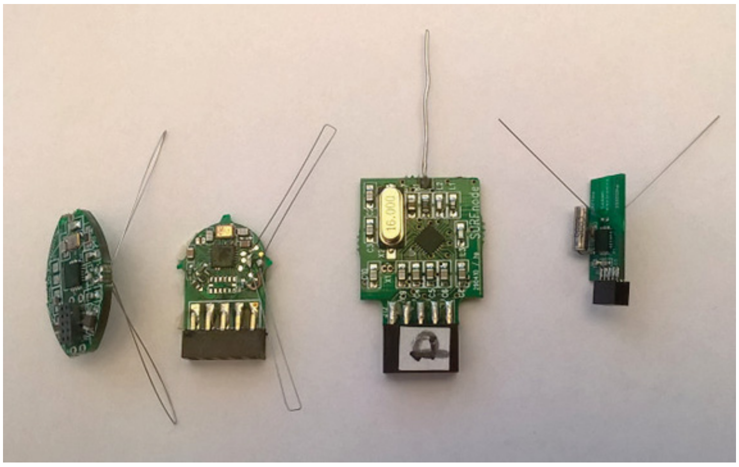


Figure 31. nRF24LE1 based SURFnet module platforms

Figure 31 shows several circuit boards based on the connection shown in Figure 30. The leftmost module can include also sensors to measure acceleration, temperature, light and voice. In addition, a modular version was developed as shown in Figure 32.

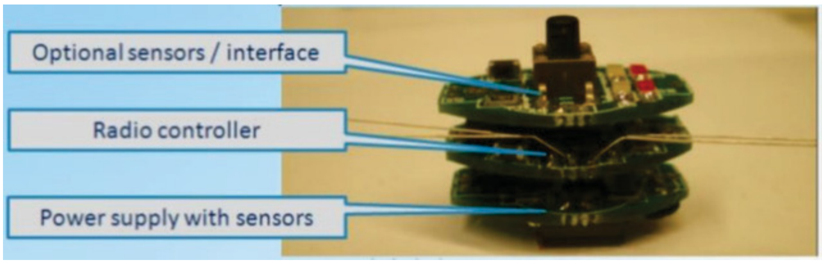


Figure 32. Modular SURFnet button (Palomäki, 2011a)

The dimensions of the smallest SURFnet button are 10 x 11 mm in Figure 33, excluding the 2.4GHz folded dipole antenna.

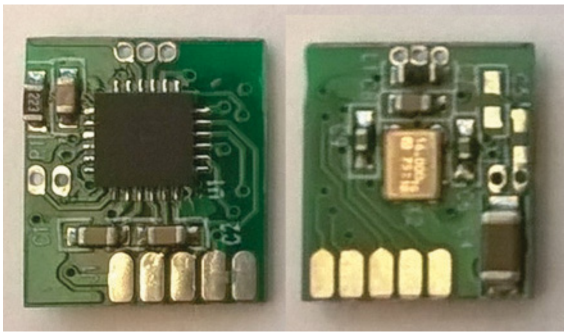


Figure 33. The smallest SURFnet button

For programming and data collection purpose the USB-SPI bridges were made in two versions as shown in Figure 34. These use Atmel AT90USB162 controllers to create a virtual COM-port for a PC. This has a direct connection to the nRF24LE1 SPI interface to program the flash memory of the controller or to open a data flow channel from the SURF network.

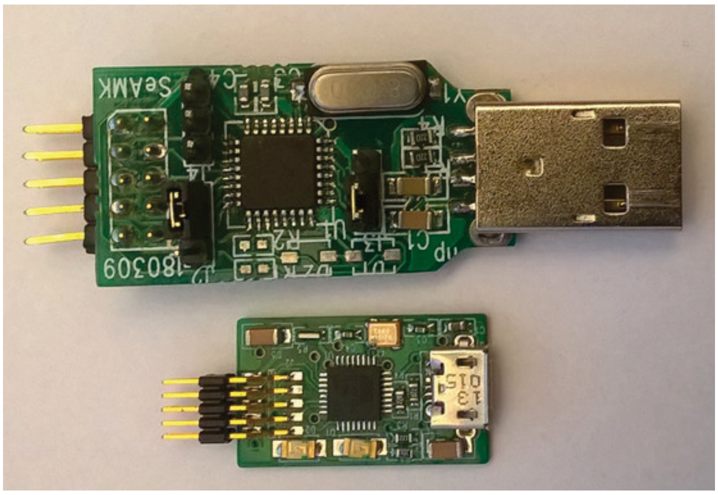


Figure 34. USB-SPI bridges

6.3 Software development

The developed software includes different versions of communication protocol stacks. The simplest version consists of a physical layer, a data link layer and a simple application layer. The development of other protocols is carried out and tested in application cases. The basic software development is made with a teaching kit using an optional radio module (Figure 27 c). The developed program modules have been moved to other layouts that have compatible controllers, so that only small modifications are needed.

6.3.1 Drivers

The physical layer is fully compatible with all layouts. The smart transceiver chip includes payload framing, address checking, error detecting and an automatic retransmit feature. Therefore, the simplest drivers for the link layer include only four functions: initialization, switch receiver on, receive message and send message:

- void initRF(unsigned char channel, unsigned int myaddr);
- unsigned char receiveON(void);
- signed char receiveRF(unsigned char *dat);
- unsigned char sendRF(unsigned int addr, unsigned char *data);

In the simplest applications, these functions can be valid as an application interface (API). One version of these drivers, a simple application and hardware layout has been made for a plug-in OEM module and they are free to download from the Internet. (Palomäki, 2008).

6.3.2 Low-power features

The transceiver and controller chips have low-power standby states, where the transceiver chip needs only 900 nA. In the active mode, when waiting for commands, it takes 22 µA. When the receiver is turning on and during receiving, it takes a maximum of 12.3 mA. When transmitting a signal, it takes a maximum of 11.3 mA (Nordic Semiconductor, 2007).

The controller chip needs in the sleep mode 100 nA if it wakes on an external signal. If waking by internal timing, it takes about 5 µA. In the active mode, the current of the controller chip depends on the supply voltage and clock speed. If a low-speed 32 kHz clock crystal drives the controller, the active current is as low as 35 µA at 3V supply voltage. These features are used while programming low-power nodes. They are usable when the application software is very simple and does not take significant time, even at a low clock frequency. At other supply voltages and clock frequencies, the controller chip takes from 0.55 mA to 5.5 mA in active mode. (Atmel Corporation, 2008).

We have tested three main usage types with the low-power modules:

- External triggered mode, which is suitable for e.g. lighting control applications. The radio module communicates only by manual operation.
- Internal timed mode, which is suitable for RFID type applications, where the RF button sends its identification code periodically e.g. every 2 seconds.
- Synchronized mode, which is suitable for a wireless network of mobile nodes e.g. in a day-care centre for children. The nodes position themselves and monitor the neighbourhood.

We have compared these three cases based on datasheets and measurements: assuming that the module is triggered 10 times per day, which battery capacity is needed if the lifetime is expected to be 3 years (Table 2). In the second case, the wake-up period was 8 seconds. In the third case, the synchronizing period was 2 seconds and high-power duty ratio 0.7 % of the time period. In all these cases, waking the RF transceiver would take 1 ms, the data transfer between the controller and transceiver would be 1 ms, and the high power sending period 200 µs.

Table 2. Low-power comparisons

	Power supply / Crystal	Average current	Battery capacity / 3 year
External triggered	2.0 V / 1 MHz	1.05 µA	28 mAh
External triggered	3.0 V / 4 MHz	1.4 µA	37 mAh
Internal timed	2.0 V / 1 MHz	6.4 µA	165 mAh
Internal timed	3.0 V / 4 MHz	6.8 µA	178 mAh
Synchronized	2.0 V / 1 MHz	164 µA	4320 mAh
Synchronized	3.0 V / 4 MHz	516 µA	13580 mAh

All current values in Table 1 are based on datasheets, but all time assumptions are based on different test cases. The practical measurements are nearly the same, for example, internally timed RF button with 3 V / 1 MHz takes on average 10 µA. However, the average µA-area current measurement of slow pulse-form current is not accurate with a traditional digital multimeter. The real battery capacities are highly dependent on battery technology, because the internal discharge varies a lot.

6.3.3 Synchronization in flat mesh topology

In a flat mesh wireless network, all nodes can function as routers and they must be able to receive message frames from a random node in the neighbourhood. However, it takes a lot of power to be in the receive state, and the battery powered node cannot be in the receive state all the time. The alternation of a long low-power sleep state and a short high-power receiving/transmitting state is carried out by common synchronization. The synchronization and optimization of the duty cycle are the most accurate functions in wireless communication and this determines the lifetime of the battery-powered wireless nodes. Synchronization must be hardware-based to function properly. At Seinäjoki University of Applied Sciences, the nRF24L01 transceiver chip was used, which is smart enough and simple to use. Time-critical optimization uses the device-dependent functions of the nRF24L01 transceiver and the ATtiny84 controller.

There are many ways to synchronize different size networks. In advanced wireless networks such as TUTWSN, the network is divided into clusters and the cluster head node controls the synchronization of the cluster. The cluster head must be synchronized with other cluster heads as well as communicate with them and this takes more power. Therefore, the role of the cluster head is taken in turns inside the cluster, and thus the power consumption is balanced (Hämäläinen & Hännikäinen, 2007).

In the ZigBee standard, the synchronization problem is solved by using higher power FFD (Full Function Device) nodes to control the synchronization of low-power RFD (Reduced Function Device) nodes. The RFD nodes cannot communicate with each other and cannot function as routers.

If you wanted to have global synchronization in a large wireless network, it is almost impossible to have the same synchronic time on different sides of the network. Although synchronization is needed between neighbours with sufficient accuracy, global synchronization can creep on different sides of the network. Another problem is the source of synchronization time. This can be solved by time priority based on the ID-code. For example, the time counter value of one single node is replaced by the time value of a node having a lower ID-number; the time values of nodes with higher ID-numbers are rejected.

To get results in synchronization, the possibilities of the transceiver and controller chips were tested first. A 32.768 kHz clock crystal was used to have an accurate and ultra-low-power time counter in the controller. The interface between the controller and transceiver is based on a SPI (Serial Peripheral Interface), so the data rate is also quite low. The active RF period must be very short, so it cannot include SPI data transfer of the message payload. The transceiver has a FIFO (First In First Out) buffer for three messages when receiving data, so the active period includes only three windows for communicating. However, the number of possible neighbours can be a lot more,. Therefore, five active periods in a round robin sequence was chosen to communicate with a maximum of 15 neighbours. The communication of one active period includes the following phases:

- Ultra low-power standby period when only the time counter functions.
- The time counter awakes the controller to initiate a time for the transmit phase, based on the tour number if it fits this active period. The software executes a low speed and low power routine to load the transmitted payload into the transceiver buffer.
- The high-power receiving state starts, including three windows to accept received messages.
- If transmit time matches, the transceiver is activated to send the message, and then it turns again into the receiving state.
- If messages are received, they stay in the FIFO buffer in the transceiver without software operations.
- After the time of three message windows, the high-power receiving state of the transceiver is turned into the power-down state.

- The controller loads possible received messages.
- The controller updates the time counter based on synchronization rules and checks its tour number.
- The controller executes the application software, for example positioning.
- The controller goes into the ultra-low-power state.

The power consumption of a single active period is presented in Figure 35. With this method, it is possible that two neighbours have the same tour number and they do not receive any messages from each other. The method is corrected by a randomly based silent period when it is time to transmit a message. During the silent period, the node with the same tour number has a possibility to transmit its message. So the tour number conflict is detected and the tour number can be redefined.

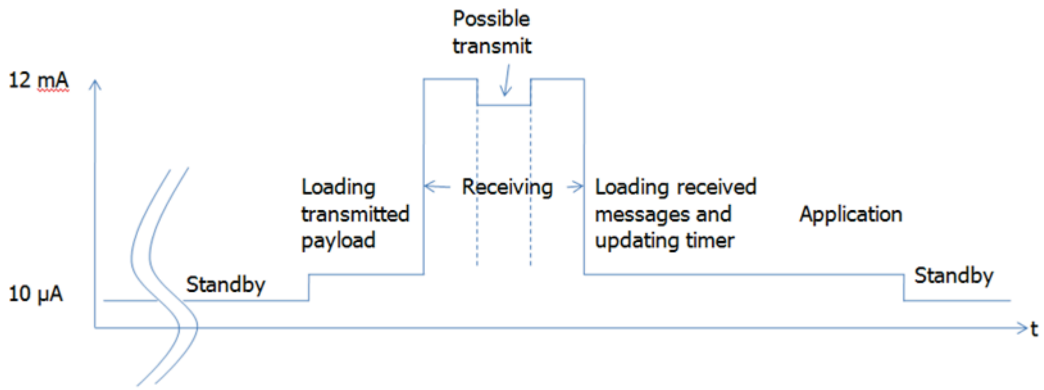


Figure 35. Power consumption of a single active period

In a high-density network, the number of neighbours can be more than 15, but this can be decreased by decreasing the transmitter power of the radio transceiver, and thus the range is reduced. This result complicates the positioning, because the ranges in the calculations cannot be fixed as suggested above. In distance measuring, the multi-hops must be replaced by cumulative range distance.

6.3.4 Synchronization in hierarchical topology

The purpose of this work was to find as simple a solution as possible to keep the hardware and software accessible and elastic. Dynamic modifications in the network structure require a fast connection without timing tuning. Therefore, the synchronization in hierarchical topology was made on the basis of collision-controlled self-synchronization.

In the simplest node, the sleep time is based on a counter controlled by an internal low-power RC oscillator. It is implemented inside the controller chip. Sleep time is therefore dependent on component dispersion and supply voltage. Time is not running at the same speed in separate nodes. The other alternative is to use an external low-speed and low-power crystal to drive the counter more accurately.

The smart transmit protocol of the nRF24LE1 controller uses an automatic retransmit feature if it receives no acknowledgement. This feature is used in the case of a collision and thus the transmission takes more time than it would normally. It also means that sleep time starts later, and so the sleep state timing shifts in collision cases. This results in that next time the transmit time point is no longer in the same place and a collision is less probable. The controller can use a maximum of 15 retransmits. The delay between transmissions can be defined individually for every node from 250us to 4000us. In addition, in the case of a collision, the retransmitted messages do not collide and both nodes succeed in transmitting.

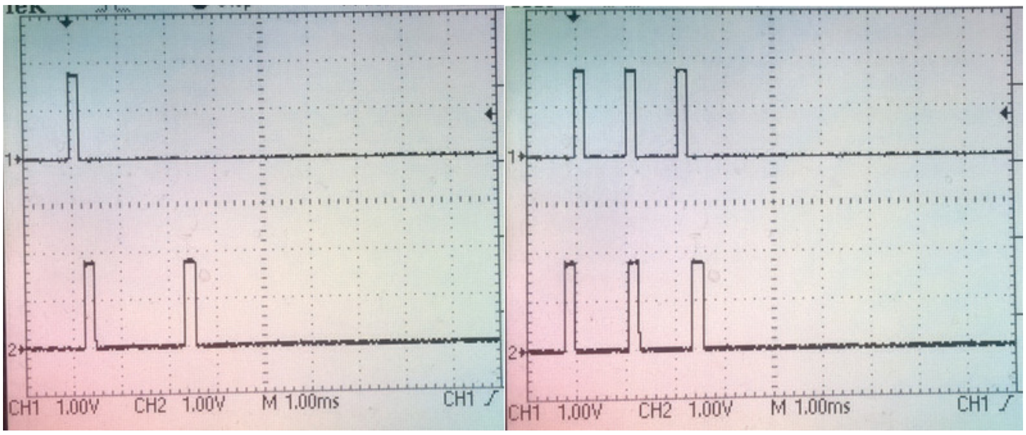


Figure 36. Normal case and retransmit case

In Figure 36, the oscilloscope screens show the typical cases, where 1=node, 2=receiving sink. On the left side, there is a normal transmitting pulse from a node and following it an acknowledge message pulse from the reader sink. The reader also accepts a message from another node not visible in this screen. On the right side, the receiver sends first acknowledge messages to the other nodes and therefore does not see the message from the node. As the node sends the same message the third time, the receiver accepts it with the following acknowledgement message.

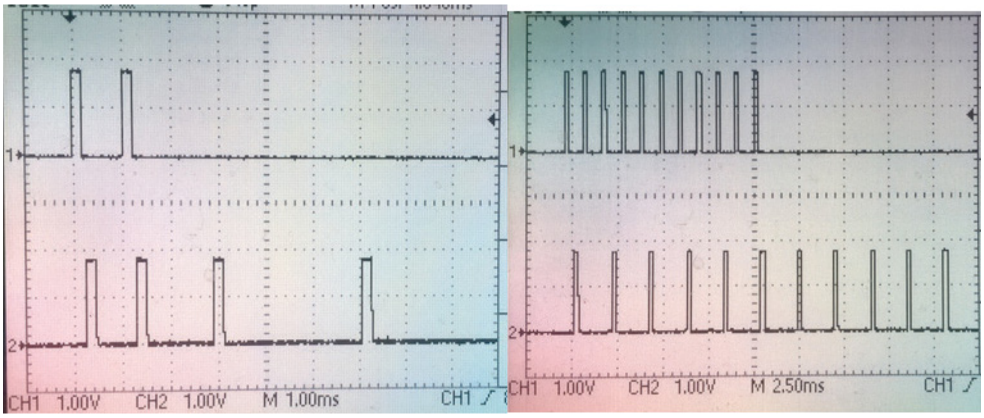


Figure 37. Communication jam and retransmit delays

In Figure 37 left, the receiver sends an acknowledge message, but because of the traffic jam the node does not see it. Therefore, it retransmits the message and gets an acknowledgement. The rightmost screen shows the separate retransmit delays of two different nodes: 750 us and 2000 us.

7 SIMULATIONS

Simulation is a simple way to view practical possibilities. Simulation shows the direction for practical development: what is worthwhile and what gives the best results. Especially when developing new methods and protocols, simulation is a fast way to find suitable results and directions for development work. Simulation is widely used to test hardware while planning and developing connections, especially when using discrete electronic components. However, using a SoC (System on Chip) radio controller, the connection needs only a few discrete components, which are not very critical. Therefore, the simulation in this thesis focuses on software simulation, which means the simulation of methods and protocols.

If the new features were to be tested using real hardware in a wireless network, the software modifications would be uploaded into every network node in every test case again and again. The test results and especially any problems in the software are difficult to see. The development process would include numerous modification phases; therefore testing with hardware in a wireless network limits the size of the network and takes too much time. Actually, it is nearly impossible.

The next simulations include routing, positioning and neural network tests. The routing simulations were carried out using Microsoft Visual Studio and C# language in calculations and to express the results. The first indoor positioning simulation uses EXCEL and macros to calculate and to express the results. The second indoor simulation and neural network simulations were carried out using C# language for the calculations and EXCEL to express the results.

7.1 Routing simulation

The routing simulation was made by Matti Ventä at Seinäjoki University of Applied Sciences as a bachelor degree thesis (Ventä, 2007). The main principles and results of the simulation are described in the next three chapters. The simulation is based on the features and limits of the planned hardware, which in this case was the older nRF24L01 radio modem and ATmega32 microcontroller. The platform is compatible with the newer nRF24LE1 radio controller and the results can be used in the newer platform. The results were tested partly with the older platform, which is not described in this thesis.

7.1.1 Direct diffusion

In simulation software, it is possible to set the dimensions of the network area, start the communication cases and position the nodes manually (Figure 38). The size of the network in this simulation was fixed as 60 nodes.

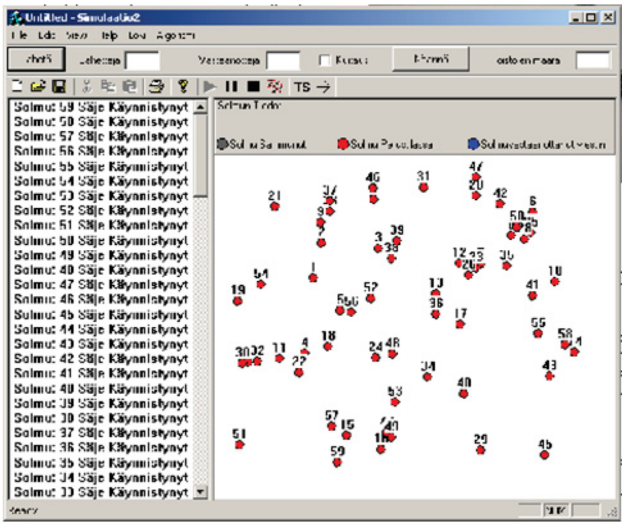


Figure 38. Simulation software (Ventä 2007)

The simulated routing protocol was developed based on the DD (Direct Diffusion) protocol being proactive, and it functions as follows: every node has some routing data records. One record consists of three numbers: the destination node ID (Identification number), the ID of the neighbour in the direction of the destination, and the distance, i.e. the number of multi-hops to the destination. The distance value is also called the altitude related to the destination. The destination, i.e. the sink node sends a question using the flooding method through the network. In Figure 39(a) sink node 6 sends a question using the flooding method to the nearest neighbours. At the end of the question process in Figure 39 (b) the last node has received the question message and updated its routing data. Mainly the routes downward that have the same altitude distance are saved. The red numbers are new data and the "Alt" number is the altitude related to node 6. For example, the numbers "6,2,1" in node 1 mean that the route to sink node 6 goes via node 2, which is 1 multihop away from sink node 6.

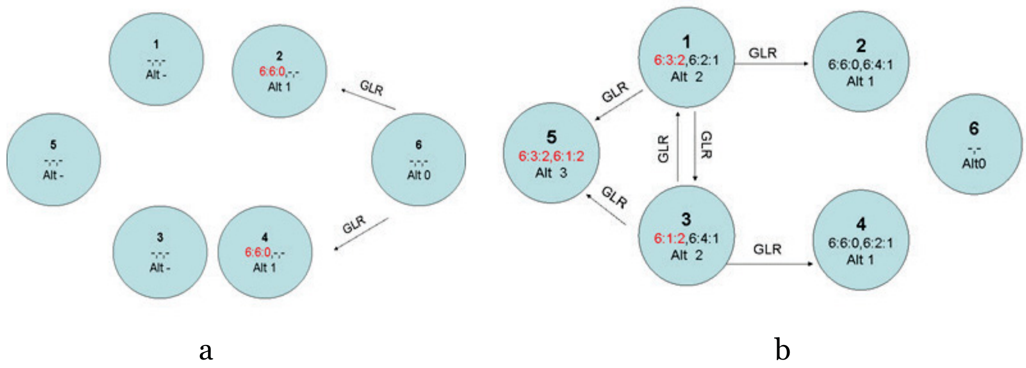


Figure 39. Flooding of the routing question (Ventä, 2007)

As can be seen, some nodes have two (or more, if defined) alternative route data. The route table can include routes to the other sink nodes as well. After the route data has been updated in the whole network, the message frames are always sent as a private message to one neighbour. When a node sends a message to the sink node, every node on the way sends the message frame downward, because the altitudes to the sink are known. If one route fails, the node tries the alternative routes and the route table is updated. If alternative routes on the same level are used it is possible that the message frame loops. This case needs an extra method: when a node receives the message frame from a node on the same level, it knows that the sink node is not in this reverse direction, so it removes this reverse routing data from the routing table. In addition, if a node receives the same message again, it removes the routing data to the node where the message was sent the first time and tries alternative routes. Therefore, it is not possible that a message stays in a loop in the network. If all alternative routes have failed as well, the node sends the message as a broadcast (not privately) to all the neighbours, and using return data it updates its own altitude and routing data.

In Figure 40 there is an example of using a simulator: node 54 sends a flooding query to ask for data from node 42. The broadcasting message flows through the network using the flooding method: The blue nodes have received the message and updated the route data. The green nodes are sending the message and the red nodes are receiving. In this case, the flooding question from the sink node finally reaches the last yellow node, which is the data source. After this flooding route query, node 42 can return data to node 54 using private addressing via the shortest route.

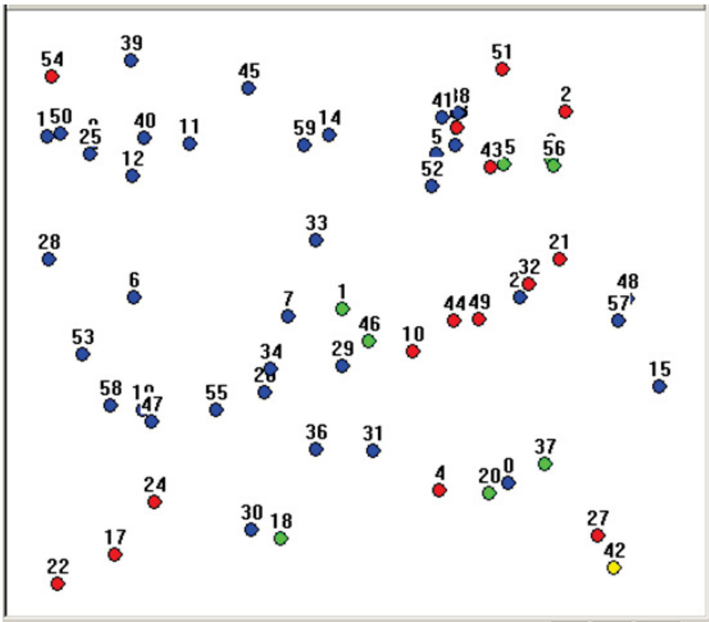


Figure 40. Broadcasting route query

7.1.2 Simulation results

The routing protocol simulation has been tested in typical problematic cases. The start state is described in Figure 41, where the route from the data source node 3 to the sink node 10 is defined. The range of node 10 is shown. There are 60 nodes in this network. The route is found from node 10 to node 3 with a flooding query. Next, nodes 1 and 58 are removed and the nodes try to find the route again (Figure 42) and the protocol finds the route. In addition, if the source node is moved, the route is found; the first time the message uses a longer way updating routing data, and the second time the message is routed directly (Figure 43).

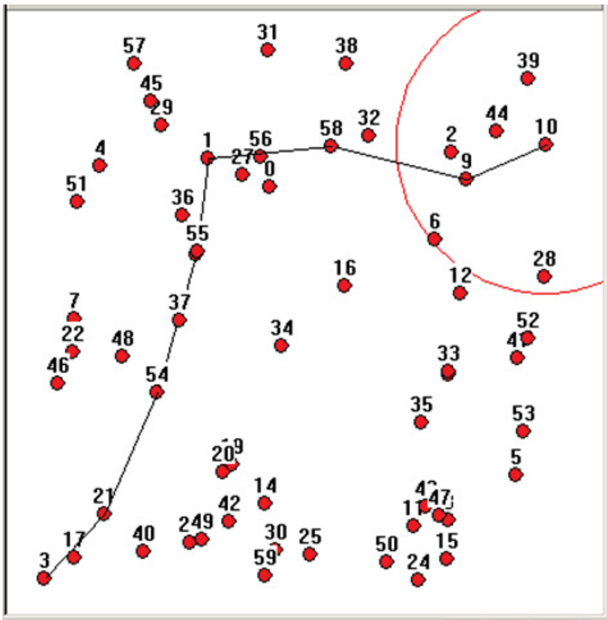


Figure 41. Start state of the routing simulation

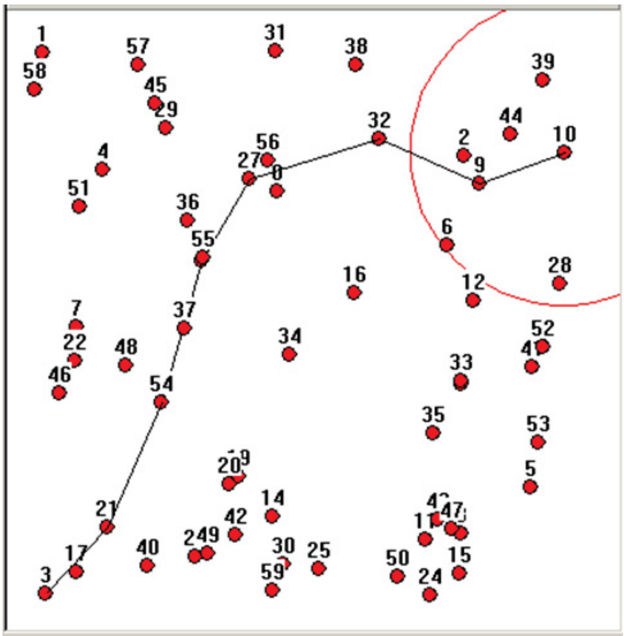


Figure 42. After removing nodes 1 and 58

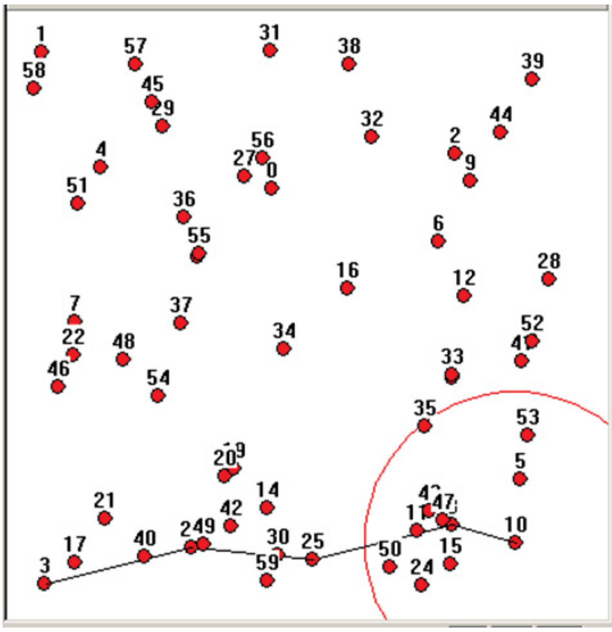


Figure 43. After moving the source and one try

Moving the destination (node 3) is critical, because the drain in the flow-down structure of the network and the routing tables of the nodes are no longer real. When this was tried, the route was lost (Figure 44).

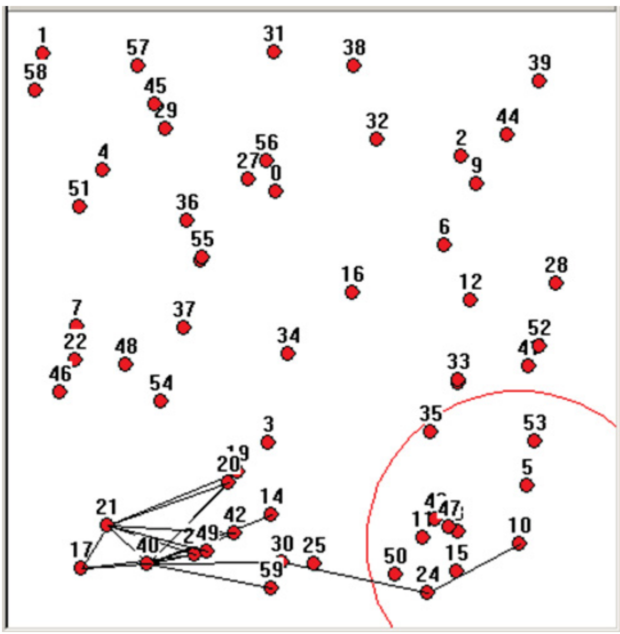


Figure 44. After moving the destination

In a mobile network, every node is moving short distances all the time. To simulate this, a new routing was started from node 34 to node 22 (Figure 45). Then all the nodes in the route were moved one by one and after each move, the route was found again (Figure 46). The destination node was then moved with short steps so that it did not lose all of its neighbours.

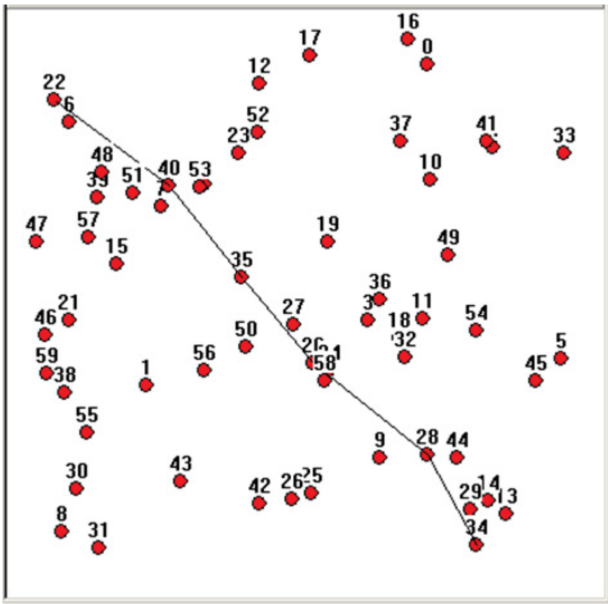


Figure 45. New start state

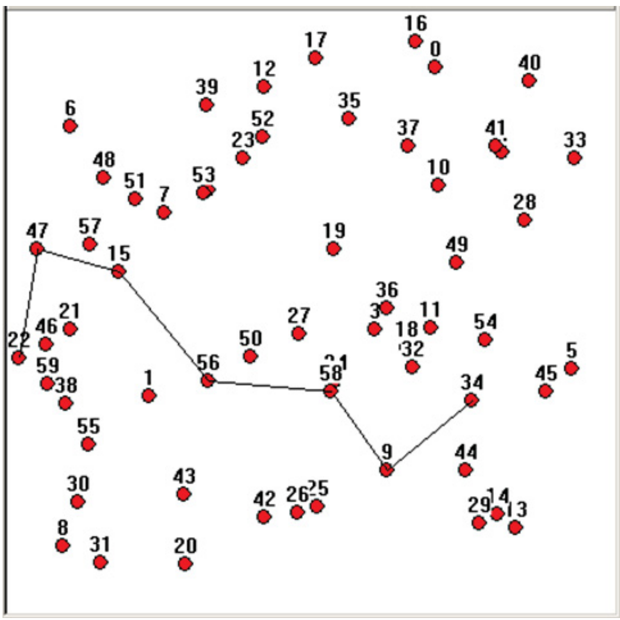


Figure 46. After moving every node in the route

As a result, it can be seen that the routing protocol finds routes very well in a normal case, when the nodes move slowly or are fixed, assuming that the destination node does not lose all of its neighbours. However, if the route must be secure, the application level in the software must check the routing, using for example an acknowledge response. In problematic cases, the message must be sent again.

7.1.3 Routing case: Gossiping routing

In chapter 3.5.4, it was shown that flooding routing loses communication resources. Flooding routing is usable if there already exist a lot of communication needs in the wireless network. Synchronization and some positioning methods need continuous message exchange between the nodes. Using radio transceiver nRF24L01 with 2 Mbps, the extra payload in the message frame does not increase significantly the load in communication. In this case, the message frame includes: synchronization data, positioning data and payload data. Every time when frames are transferred, the routed payload data spreads through the network like gossip. This gossiping routing does not cause a loss of communication resources in this case, but the propagation delay can be very large. Gossiping routing is suitable for a few applications, where the controller updates the sensor data or the positions of the non-moving nodes slowly.

The communication resources of the network are easily overloaded when using slow gossiping routing. To decrease overloading the gossiping routing can be directed, i.e. adding position-aware features or altitude data. That is possible if a relatively small network has a positioning method where every node has multihop distances to other nodes. The source node knows the distance to the destination and the original message frame includes this distance. Every time some node retransmits the message frame, it replaces the real distance to the destination. When a node receives this frame, it is not retransmitted if the real distance to the destination is more than the distance in the message frame. By this method, messages always flow in the direction of the destination and never in the opposite direction.

7.2 Neighbourhood positioning simulation

The most used positioning method today is satellite-based GPS. In indoor applications it does not work, because RF signals from satellites do not penetrate the building structure. In addition, the accuracy is not sufficient for indoor applications.

There are many ways to calculate the positions of the nodes using distances between the nodes. If a node has very little resources and the application does not need exact positioning, it can be enough to use only the neighbourhood data. In this simulation case, the focus is to use very simple electronics to measure distances between nodes using distance measuring by using range and multihops. The controller nRF24LE1 can use four different RF output power levels, which means four different ranges. A node measures the distance to other nodes inside its range by transmit power variations and the nodes outside of its range using the number of multihops. In both cases, the calculation does not include any trigonometry and square root functions to fit into the reduced resource of the node controller. The biggest advantage of this neighbourhood positioning is that the network size can be bigger that any node range. This positioning method was tested in a practical study case in chapter 8.1.2.

The planned practical hardware for this simulation was an nRF24LE1 radio controller and the corresponding platform. The simulation was carried out using EXCEL and macros in both calculation and in the visualization of the results (Palomäki, 2008, ss. 48-55).

7.2.1 Distance measuring

The RF-transceiver controller nRF24LE1 has some simple features to measure the distances between network nodes: it is possible to control the output power with the software at four levels: 0dBm, -6dBm, -12dBm and -18dBm. Thus, a node can form four circles around it, and it knows which neighbours are inside which circle, i.e. it can measure distances to its neighbours in four steps. Assuming that the range of the RF-signal is reduced as a square root of RF power, it results in the equation below (1):

$$r_r = \sqrt{2^{\frac{p}{3}}}\, r \qquad (1)$$

Where
p = output power in dB
r_r = reduced range
r = original range

While using RF output power reducing by 0, -6, -12 and -18 dB in the equation, the corresponding ranges are r, r/2, r/4 and r/8. In the study case, where the range is 20m at maximum power, a node in the network can measure which neighbours are inside the ranges 20m, 10m, 5m and 2.5m.

The compatible transceiver chip nRF24L01 includes no controller, but it has a feature to control the gain of the receiving amplifier (LNA, Low Noise Amplifier). The gain can be reduced by 1.5 dB by the software. If the equation (1) uses p=-1.5, the reduced range is 0.8 r. This feature was not used in this positioning simulation case.

Distance can be measured also by using the number of multi-hops in the routing method. As default, multi-hops use maximum RF power. If a multi-hop route is the shortest route between two nodes, the number of multi-hops is also the distance between these. Depending on the size of the routing table and memory capacity, distance measuring can be applied and easily used in simple wireless networks (Palomäki, 2007a).

7.2.2 Limited resources

Because of the limited resources in the nRF24LE1 controller, the positioning algorithm must be simple: the calculations should be done with integer values. Floating point arithmetic or trigonometric, which require a high amount of resources, are not recommended. If distance measuring needs two or more output power steps then two or more transmit sequences are also needed and it takes more power. If the node has a small capacity battery and a long lifetime is needed, this kind of measuring is therefore not recommended. Via synchronization or routing, the node usually knows its neighbours; that is the minimum data for the positioning method. Usually very rough positioning is enough, for example when used in animal positioning in a cattle house, so the method can be very simple.

Absolute positioning needs some anchor nodes, which have fixed coordinates to function as reference points for the positioning estimation. The anchor nodes can be equal to others that are communicating in the same way, except they are not modifying position coordinates. It is possible to find positions without the anchors, but the positioning is relative and the positioning area can be mirrored.

In this position simulation, the focus is on distributed positioning, where the calculation is executed in the nodes. It is also possible to calculate the node positions centrally in a more powerful central processor, and then the results should be more accurate. The main research concern however is about fully distributed systems (Palomäki, 2007b).

7.2.3 Simulation principles

There are some predefined and fixed values in the position simulation: the size of the area was 20 x 20 units and four anchor nodes with fixed coordinates were located in the four corners of the area. Figure 47 shows an example of the start state when simulating a network with 60 nodes.

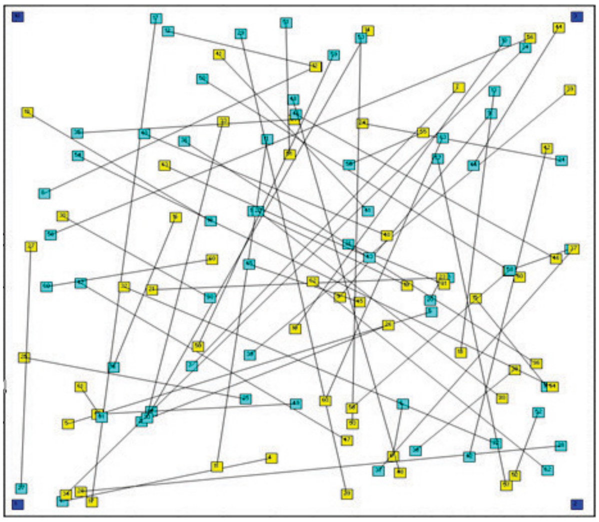


Figure 47. Start setup of simulation

The dark blue marks are anchor nodes with a fixed position. The light blue marks are the real positions of the nodes, which are located at random. The yellow marks are the estimated positions of the nodes, which are first located at random and move during the position estimation process.

7.2.4 Position estimation

The measured distances and the algorithm types are as variables in the positioning simulations. The position estimation of one node is made by scanning all neighbours and the distances to these and then calculating its own estimated position according to the position data of the neighbours. The principle of this simulation is that functions of all nodes are equal, but the fixed anchor nodes do not recalculate their estimated positions, because they have fixed positions. In the positioning calculations of one single node, the input data is as follows:

- Own estimated position data
- Measured distances to the neighbours inside its range and their measured distances to the next neighbours etc.

- Estimated positions of the neighbours inside its range and the estimated positions of their neighbours etc.

The positioning algorithms use different ways to calculate the estimated position. The simplest way is to calculate their own estimated position as a mean value of the estimated positions of neighbours inside its range and the nearest multihop-neighbours. This is a very fast way compared with other methods. Another way is the iterative position estimation, where their own position is corrected by and by so that the measured multihop distances to every neighbour and the estimated positions received from the neighbours are as equal as possible relative to their own position. Because of the inaccuracy of neighbour position estimations and distance measuring, the optimal value of their own position varies relative to every neighbour. That problem is solved by correcting their own position step by step until the result is stabilized in the iteration process. This method is relatively slow and the balance between accuracy and speed must be solved separately in every application.

In a network, where the low-power nodes are moving powered by small batteries and the routers are also battery powered, positioning is more problematic. The nodes must be in synchrony to keep the duty cycle as small as possible and it needs continuous data exchange between the nodes to keep the time counters equal in the nodes. In the radio controller that was used, the bit rate can be as high as 2 Mbps. Therefore, the length of the max. 32 byte message is not significant at the high baud rate. It means that continuous synchronization loads the communication resources significantly, while positioning information is 'riding' nearly free inside the message frame and does not need any significant extra power. In addition, in gossiping routing synchronization traffic can include a few measurements and control data. However, the iterative position method needs calculation resources and energy. The controller must be active for a short time while waiting for radio communication events, so this waiting time period is suitable for the execution of calculations. Actually, the power and time needed in positioning and routing methods are much less and almost insignificant relative to synchronization, but the memory resources needed for positioning are relatively high and must be taken into account when choosing the controller and the routing method. The slow routing method almost functions for free with the synchronization power and the positioning memory resources.

7.2.5 Positioning calculation

In the mean value positioning method, the calculation is made by using the different weights of neighbour positions. The neighbours inside its range have the highest weight. The positions of their neighbours have a lower weight etc.

In the passive iterative positioning method, the estimation calculation uses the measured distances and estimates its own coordinates in two ways: First, if the estimated position of a neighbour is within the measured area, no corrections are made. Second, if the estimated position of a neighbour is outside the measured area, the position of the node is tuned step by step in the direction of the correct position, depending on the measured distance to this neighbour as described below. Figure 48 shows a state where the neighbours have been measured to be between the ranges r1 and r2, i.e. within the measured area; the estimated positions are shown as red points.

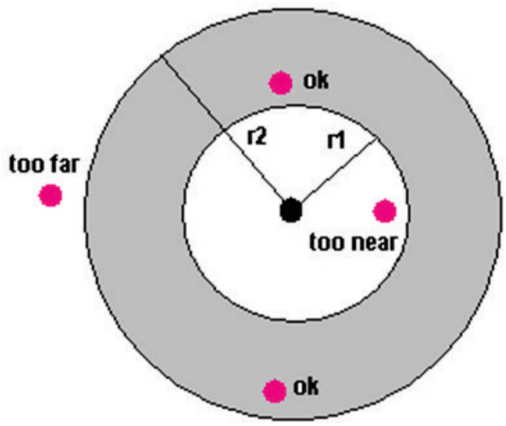


Figure 48. Passive positioning: neighbours in relation to the area

In the active iterative method, estimated position tuning is always made, except that in this case the neighbour is located on the centre line of the measured distances. The centre line ra between distances r1 and r2 is calculated so the area inside the centre line is the same as the area outside the centre line.

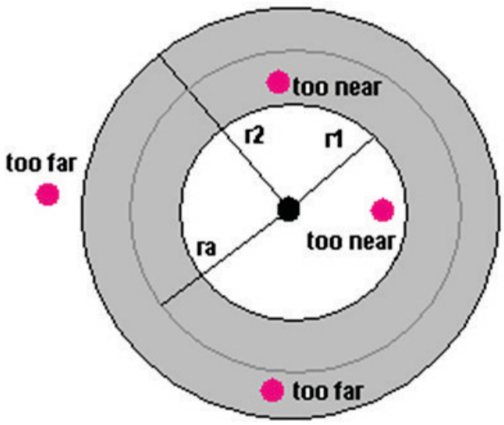


Figure 49. Active positioning: neighbours in relation to the centre range

The neighbours are nearer or further from the centre line ra. The areas on both sides of the centre line must satisfy the equation:

$$r_2^2\pi - r_a^2\pi = r_a^2\pi - r_1^2\pi$$
$$\Rightarrow r_2^2 - r_a^2 = r_a^2 - r_1^2 \qquad (2)$$

Solving ra, the result is:

$$r_a = \sqrt{\frac{r_2^2 + r_1^2}{2}} \qquad (3)$$

In most reduced nodes, the distances are not measured by multihop; only the neighbours inside the range are indicated, so the equation can be simplified:

$$r_1 = 0 \Rightarrow r_a = \frac{r_2}{\sqrt{2}} \qquad (4)$$

The resources of one battery powered node is limited, so it is not recommended to use the square root. If the ranges are known, the centre range can be fixed as a constant variable using equation (4).

The correction of an estimated position of a node is made iteratively step by step by pushing or pulling the position nearer or further according to each neighbour (Figure 50).

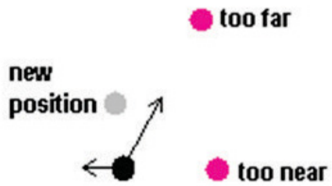


Figure 50. Push and pull corrections

The position simulations include four positioning algorithms with different features:

- Mean value calculation: the coordinates of a single node are the mean values of the coordinates of the neighbours. When measuring distances, the coordinates of near neighbours have more weight than neighbours located further away.
- Sectored mean value calculation: the same as above, but the mean values are calculated separately in four direction sectors before the main mean value is calculated.
- The passive iterative method: the coordinates of a node are pushed or pulled according to the measured distance of every neighbour, but only if they are outside the measured area.
- The active iterative method: the same as above, but the coordinates are continuously pushed or pulled if the neighbours are not located on the centre line.

7.2.6 Evaluating

To determine whether the simulation results are acceptable or not, they must be compared with known values and with each other. First of all, evaluations are made to find the values to compare, and then second, all the results are compared.

In the simulation test case used, the number of nodes is 60. The size of the area is 20 x 20 units. Four fixed nodes are located in the corners of the area. During testing, both the real and the estimated positions are first in a random state. The results are calculated using the mean value of ten single results. An example of a random state is given in Figure 51. The dark blue marks are the fixed nodes, the light blue marks are the real positions of the nodes, and the yellow marks are the estimated positions of the nodes. The Mean value of the Square of Distance Errors (MSDE) is 126 units in this example.

What is a good position result? It is fully dependent on the application. To get some reference it must be checked visually to be able to make an estimation. An example of positioning quality is shown in Figure 52, where the MSDE is 22. It is hardly sufficient in some applications. In the example in Figure 53, the MSDE is 4 and it is acceptable in most applications described in this paper. The example in Figure 54 is good, as the MSDE is under 3 units.

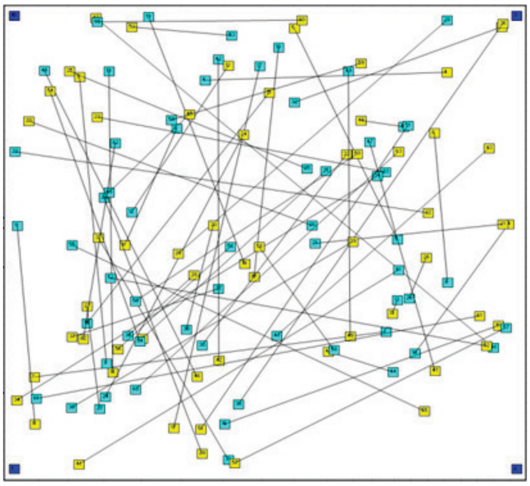


Figure 51. Result example when the MSDE = 126

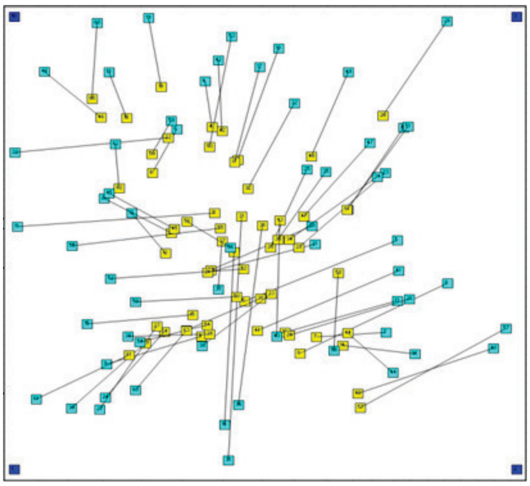


Figure 52. Result example when the MSDE = 22

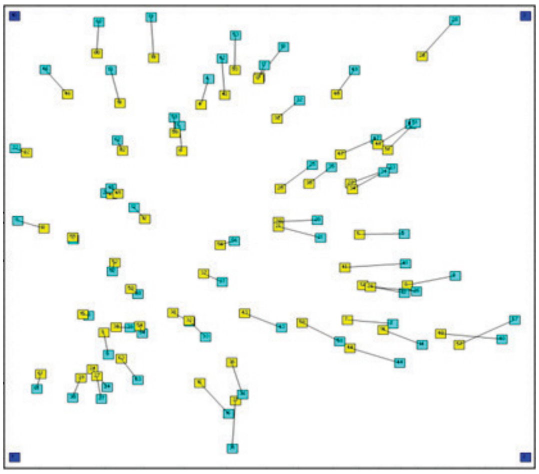


Figure 53. Result example when the MSDE = 4.0

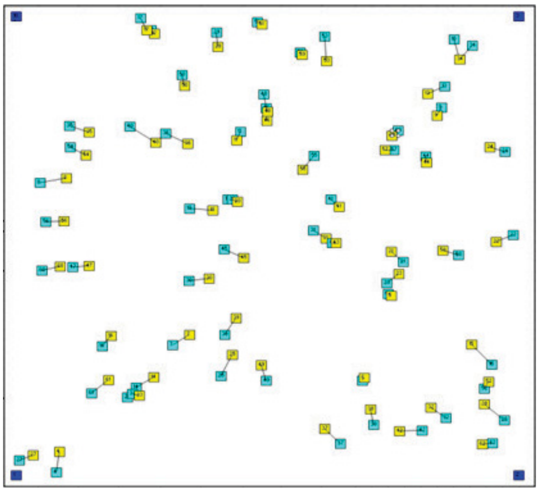


Figure 54. Result example when the MSDE = 2.6

7.2.7 Simulation results

The range distance is very critical for testing positioning methods. It is the main variable for comparing the methods; the other parameters are fixed.

While comparing the positioning methods at different ranges, it is possible to use 4 different ranges, which simulates 4 different RF output powers. In addition to the maximum range, the simulation uses ranges 1/2, 1/4 and 1/8 of the maximum. The measuring steps can be realized by controlling the output RF-power of the transmitter or by the multi-hop numbers on the routing table.

Only the tests where the MSDE reached a value under 20 units are significant here, all other results are unusable. Positioning with both fixed and mobile anchor nodes is shown in the following graphs. The data for the curves has been generated from the mean values of 20 position cases. In every case, the start positions and movements have been made randomly.

Using a maximum range of 5 units, every node has on average 6 neighbours. The results with most methods are not satisfactory; only the mean calculating methods provide sufficient results with 2 to 4 steps in distance measuring, as shown in Figure 55. The names of the lines on the graph are according to the method number and the number of distance measuring steps. For example, “Fix1.2” represents the fixed anchor node simulation of method 1 using 2 distance measuring steps.

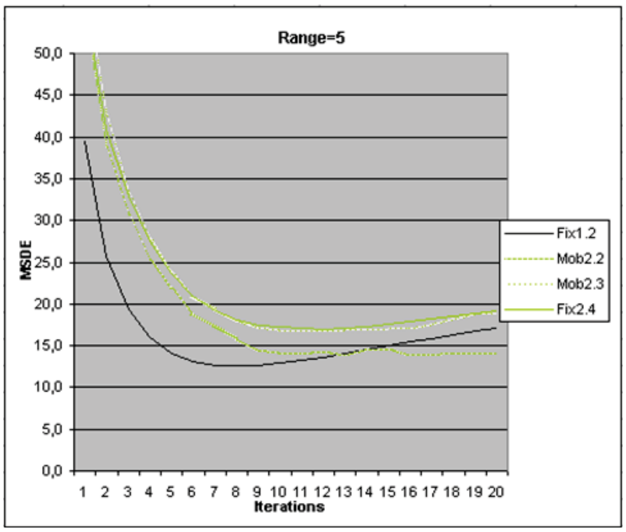


Figure 55. Positioning with range 5

With the maximum range of 8 units, the nodes have on average 14 neighbours and the results are better: Furthermore, the variations of method 4 functions are very acceptable (Figure 56). Here it can be seen that the mean calculation methods 1 and 2 are much faster than the others are, but the results of the other method are not better.

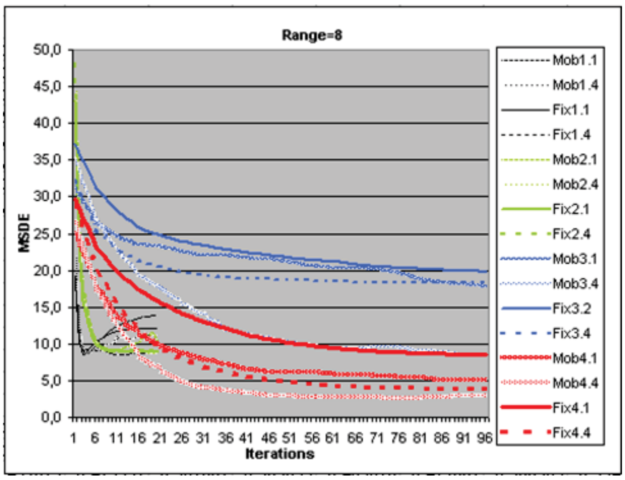


Figure 56. Positioning with range 8

With the maximum range of 11 units, every node has on average 22 neighbours, and method 3 also functions better (Figure 57).

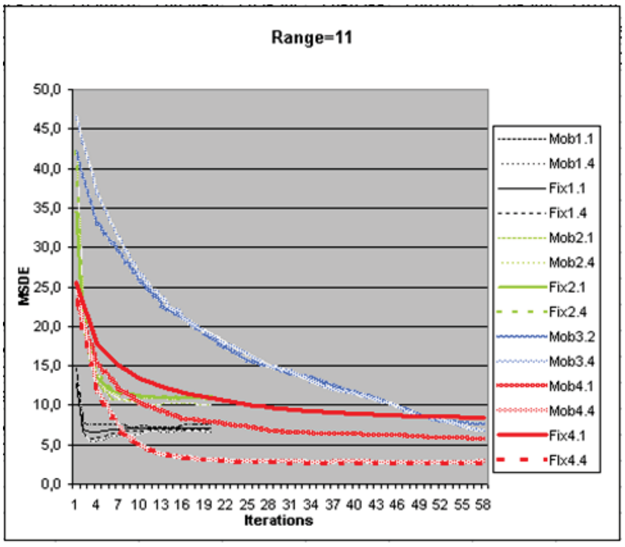


Figure 57. Positioning with range 11

With the maximum range of 14 units, every node has on average 29 neighbours. The results are good, but not as good as with range 11 (Figure 58). Here it can be seen that distance measuring in several steps is needed in dense networks.

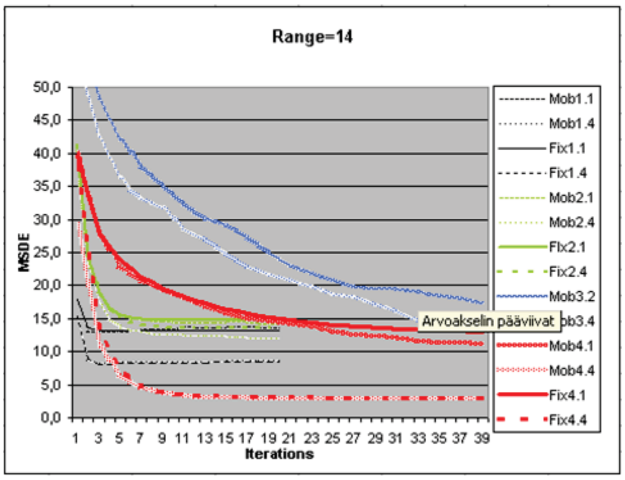


Figure 58. Positioning with range 14

Next, the methods in different node density (i.e. ranges) are compared. The variable is the number of neighbours compared to the MSDE value in every method (Figure 59). In this comparison, the last iterated MSDE value is used in every positioning method.

The results of the tests show that in coarse wireless networks calculating the mean value (Methods 1 and 2) is acceptable in some applications and this is a very fast way to position the nodes. It is a usable method also in denser networks but method 4, which is a more complicated and a slower method, provides results that are more accurate. In dense networks, more accurate positioning with some kind of distance measuring is needed to separate the nodes from each other.

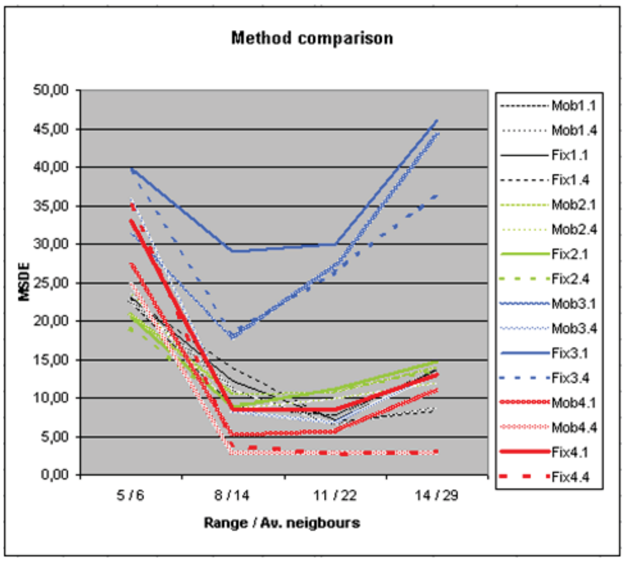


Figure 59. Comparison of the methods

Following these simulations, some practical test of the position methods began (Palomäki & Kivikoski, 2008). However, the results in chapter 8.1.3 are not yet as good as in the simulation.

7.3 Indoor positioning simulation by RSS

If the nodes in a wireless network include features to measure RSS (Receive Signal Strength), positioning can be much more accurate. One positioning method is to use a minimum of three fixed-positioning anchors that send a continuous RF signal. Nodes measure the RSS from every anchor and calculate its position. The accuracy of this positioning method depends on the environment and on the real transmit power of the anchors. The possibilities to change the environment are limited and are not taken into account in this case. The transmit power of the anchors depends on the value dispersion of the components and the mechanical structure and material variations of the antenna. The aim of the following method is to find the real transmit power (gains) of the anchors to get a more accurate positioning result. In addition, in this case, a neural network is used to find the anchor transmit gains using some fixed position nodes as example information. The principle of the neural network is written in chapter 4.1.1.

The planned nRF24LE1 radio controller and the corresponding platform define the limits and the base features of this simulation. When realizing this positioning method in practice, the area is limited to inside the range of the anchor nodes.

7.3.1 Indoor area setup and gain tuning method

The simulation case uses an indoor area of 100 x 50 units. Three corners of the area have anchor nodes to send continuous RF signals using their own unknown gains. The area includes 18 reference points in fixed positions. The Neural network teaching phase uses some of these points to estimate the anchor RF gains, others are for test purposes (Figure 60).

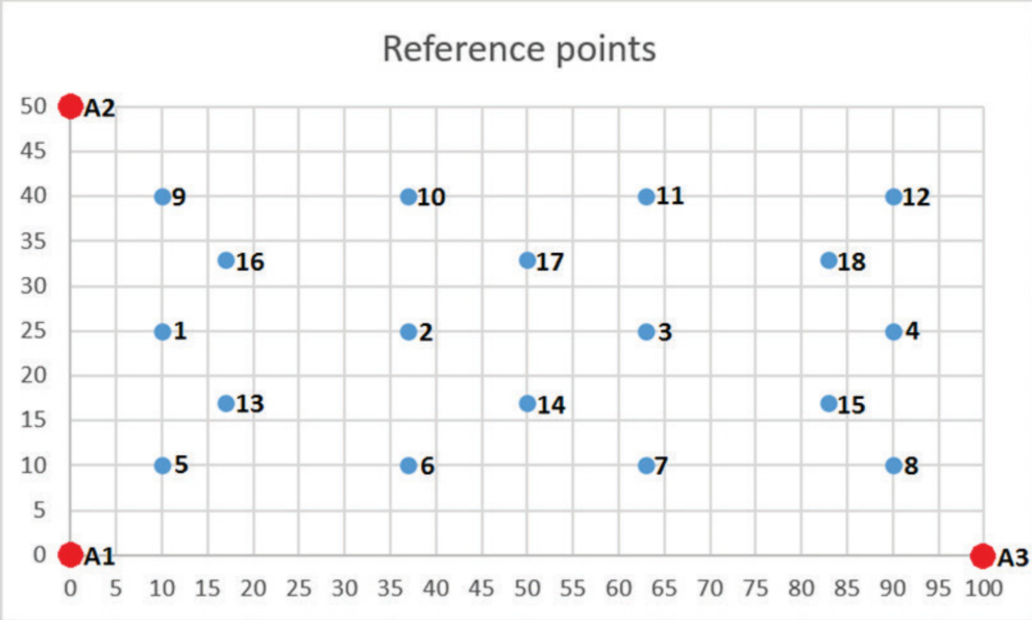


Figure 60. Indoor positioning area

In the simulation cases, the positions of the selected points are calculated based on the RSS from the anchors with default gains and using trigonometry. The neural network tunes the anchor gains comparing calculated coordinates with the fixed coordinates of the selected points. As default, the real gains for anchors A1, A2 and A3 were 0.5, 0.8 and 0.2, which the learning phase of the neural network tries to find. The estimated gains for all anchors were all 1.0. Using numerous iteration loops, the estimated anchor gains are more and more realistic and the positioning of all the points are more accurate. The result of the learning phase of the neural network depends on the number of selected points and on the positions of these points.

7.3.2 Anchor gain tuning results

In the first case, the neural network uses nodes 6, 7, 10 and 11 from the middle of the area to estimate the anchor gains. It takes 31 iteration loops to find the real anchor gains accurately enough. As a result of the right gains, the neural calculating found the coordinates of every other node accurately as shown in Figure 61. At the beginning, the gains drift before finding values that are more accurate. The coordinates of the nodes drift also, but remain inside the defined area until reaching the right coordinates.

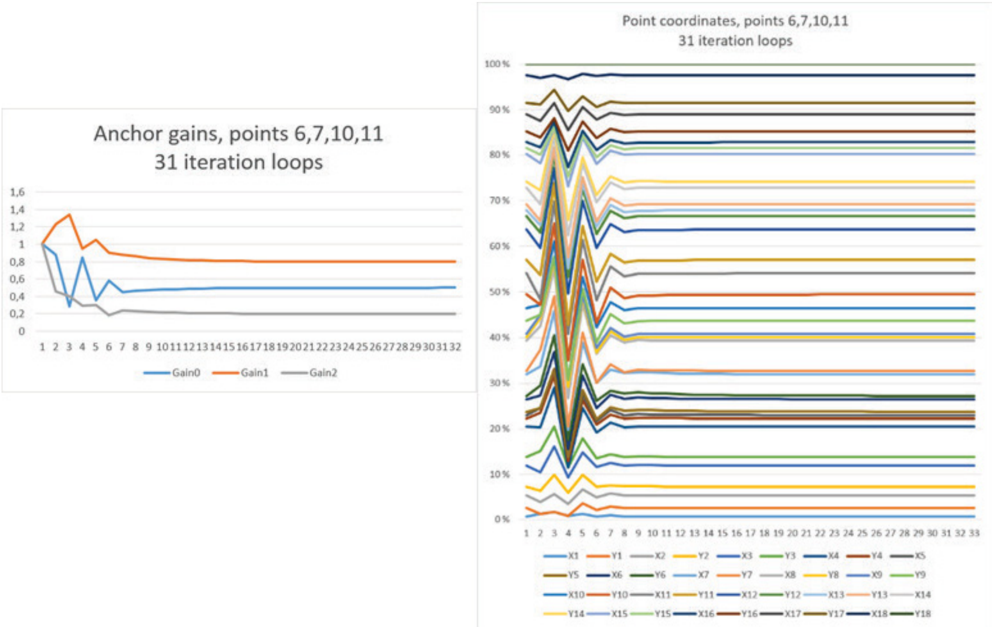


Figure 61. Anchor gain tuning, case 1

In the second case, the neural network uses nodes 13, 15, 16 and 18, which are located away from the middle area. The result was very different: after 6 iteration loops, the trigonometry results on position calculation was out of the defined area. Figure 62 shows how the gains drift in case 2 and the coordinates of the nodes drifts outside the defined area 100 x 50. The position calculation with triangular trigonometry gives imaginary values and has no results. If the selected nodes are too near to the area boundaries, the coordinates can drift outside the area and this result in problems.

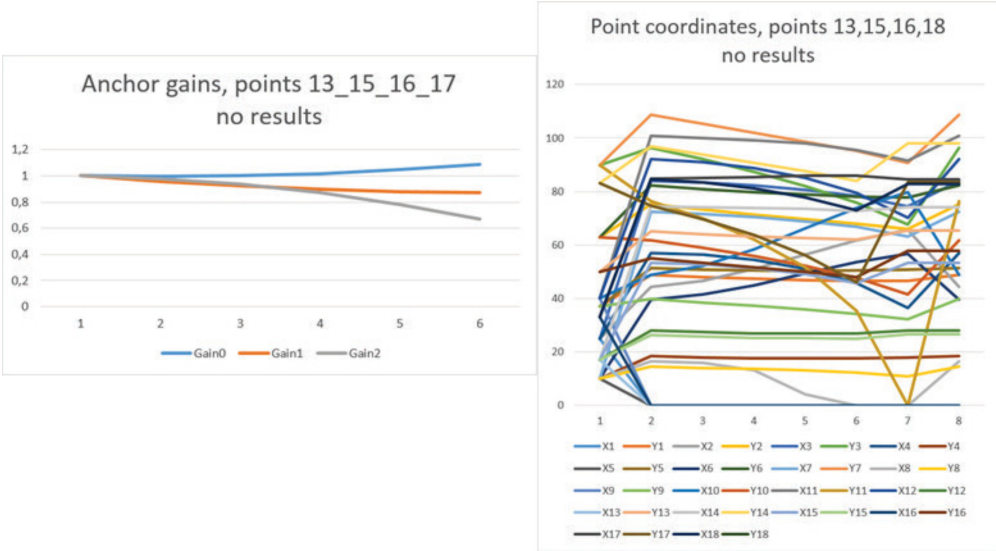


Figure 62. Anchor gain tuning, case

In the next cases, the aim is to test the anchor gain tuning method with only two selected reference nodes. As seen before the nodes should be in the middle of the area. In the third case, the selected nodes are 2 and 3.

In the third case, in Figure 63, the neural network needs more iteration loops to have stabilized gain values: 182 loops. The found gains were 0.4425, 0.9845 and 0.2272, while the real gains were 0.5, 0.8 and 0.2. The maximum error was 4.25%, so the result is much worse than with four reference nodes.

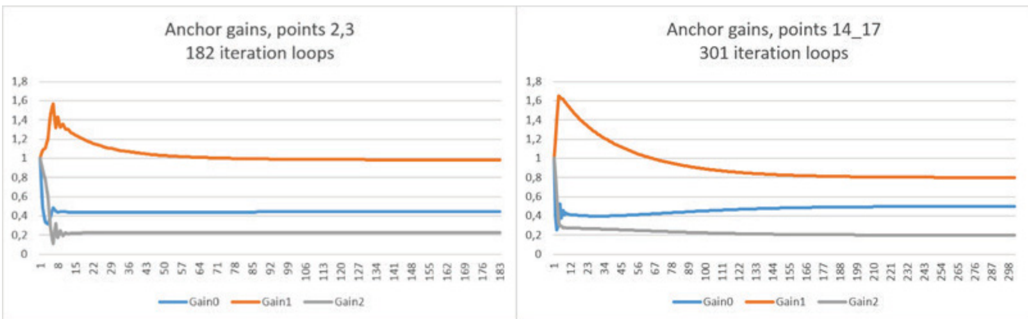


Figure 63. Anchor gain tuning, cases 3 and 4

In the fourth case, the selected nodes are 14 and 17. The fourth case in Figure 63 was very similar to the third, except it needs 301 iteration loops to have stabilized values. The found gains were 0.4996, 0.8006 and 0.2002, and the maximum error was 0.06%. It seems to have good enough results with two selected reference points, if the nodes are in optimal positions.

7.4 Automatic excavator depth measuring tuning

The exact use of the excavator requires some kind of depth measuring system to know the position of the bucket tip and to ensure a good result when carrying out excavating work. The excavator needs a depth measuring system, for example when preparing the base of building foundations or digging a diversion ditch. The accuracy of the results depends on the measured dimensions of the excavator bars and the accuracy of the tilt sensors for every bar.

Wireless sensors simplify the installation, but there are a lot of installation stages, which requires accurate manual work. The installation of the measuring system requires care when assembling tilt sensors and measuring bar lengths. The different excavating work phases use different bucket shapes and the tilt sensor is normally located in the bucket connector, therefore because the buckets have different angles and lengths it changes the depth calculating parameters. When moving the measuring system to another excavator, all the installation phases must be carried out again. Accurately setting up the installation takes a lot of time and there are many potential errors.

The idea of the automatic tuning method is to let the excavator measure and tune itself. It already has sensors and with neural computing this kind of automatic tuning method is possible.

7.4.1 The depth measuring method

The measuring system requires the length and the tilt of every bar: main bar, arm bar and bucket. The most used tilt sensor measures the acceleration between the sensor direction and the direction of gravity. Therefore, the sensor gives the sine or cosine of the angle in relation to the direction of gravity. The horizontal or the vertical dimension of the bar is the length of the bar multiplied by the sine or cosine of this angle. The principle of bucket tip position calculation is shown in Figure 64.

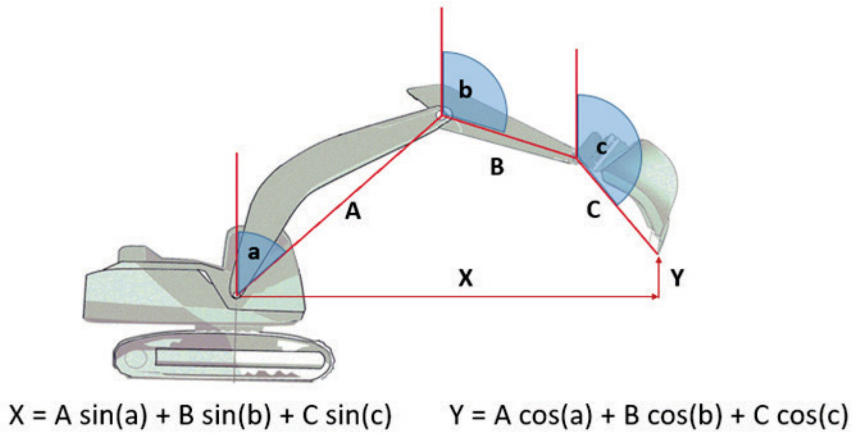


Figure 64. Excavator bucket tip position calculation

In a normal case the main bar length (A) and the arm bar length (B) are measured and are constant. The bucket length and the angle vary during the work stage. All angles can vary at installation. Because when installing a measuring system in a new excavator, all of these values are variable.

7.4.2 Tuning method with neural network

While executing automatic tuning with a neural network, it needs learning data as an example of the right measured data, as mentioned in chapter 4.1.1. In this case, the learning data consist of some fixed, predefined locations aligned by the bucket tip in the learning phase. The learning phase tunes the weights or features of a neural network to fit with the learning material in the iteration loop. The method controls the activity of weight adjustments using coefficients. Later in real measurement cases, the system uses the found weights when calculating the real position data.

In this excavator case, the weights are the unknown data of the depth calculation: bar sensor angles and the bar lengths. The teaching material is a set of fixed reference test points. Using the teaching iteration loop, the optimal method finds the actual sensor angles and bar lengths without manual tuning and measurement work.

In a practical teaching phase, the excavator driver points to every test point with the bucket tip as accurately as possible. The first trunnion of the main bar in the excavator body should be the origin. The accurate positioning of the excavator itself is not always possible, so the position of the trunnion is also unknown. Now the neural network has a maximum of eight weights or unknown features to find in the learning iteration loop:

1. Main bar angle error
2. Arm bar angle error
3. Bucket angle error
4. Main bar length
5. Arm bar length
6. Bucket length
7. Origin X-error
8. Origin Y-error

The block diagram of the learning phase of the tuning method using a neural network is in Figure 65. Test reference point locations are the fixed predefined point coordinates, which the excavator driver points to with the bucket tip. Sensor values are the real measured sensor values from every reference point scaled to be sines or cosines of the bar angles. Sensor data includes angle errors and origin position error.

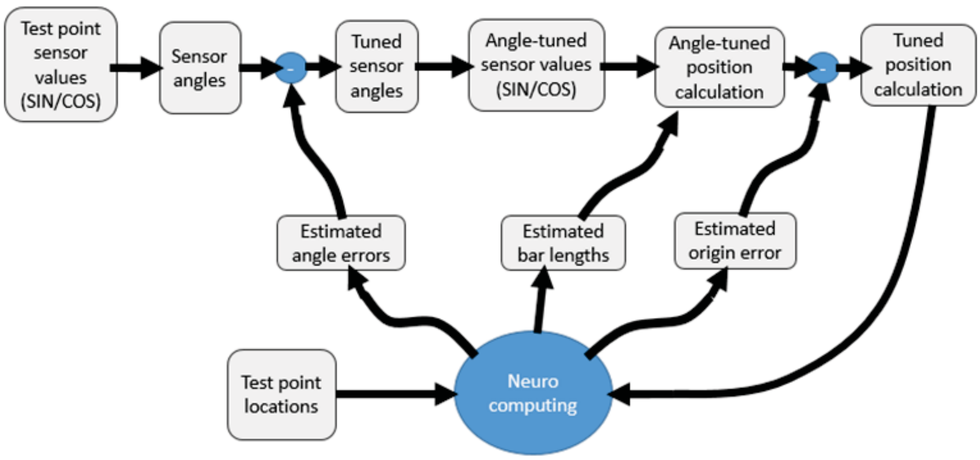


Figure 65. Neurocomputing iteration loop for excavator tuning

Calculation removes the estimated angle errors (calculated in the neural network) from calculated angles and again generates sine/cosine values of angles. Using the estimated bar length values, the tuning program calculates the estimated tuned bucket tip position. It removes the estimated origin error from the result to give the estimated test point coordinates. The neural teaching phase iteration compares these coordinates with the predefined fixed-point coordinates. This iteration gives more and more accurate values for the estimated data and hopefully finally finds just the right weights: angle errors, position error and lengths.

The calculation of the unknown estimated feature values uses three separate tuning coefficients to stabilize the iteration loop results: for the bars, for the angles and for the origin. Bigger coefficients give the result faster or the calculation oscillates without obtaining a results. If the coefficients are smaller, the calculation needs more iteration loops to get the result and it takes more time. Three separate tuning calculations with three coefficients influences each other and it results in new problems. The next simulation shows that the coefficients must be in balance: too small a coefficient effects unbalanced calculations in the other calculations and gives no tuning results. Therefore, the coefficients must be in the secure range found by the trial run. This kind of tuning method needs a lot more research to be usable in the real world.

7.4.3 Simulation of tuning method

Simulation uses the tuning method with virtual data. As an input, simulation defines the initial reference test points including the origin error and the initial bar angles for every point. Using input data, the simulation first generates initial bar angles for every bar at every test point and the corresponding initial sensor values, which are the sine and cosine of the angles. The simulation adds the real angle errors to bar angles to generate simulated sensor data and corresponding accelerations. After that, the simulation calculates the default positions for the test points from the default sensor values. The points include origin error, which the program removes using default origin error to get the actual default test point coordinates. Then the simulation follows the teaching block flow described above. The block diagram of the simulation is in Figure 66. When the simulation is working properly, after simulation the real angle errors, bar lengths and origin error are just the same as the corresponding estimated values.

The simulations use three, four or five test points. As mentioned above, every simulation case requires different teaching coefficients, which should be in balance. While testing the simulation, angle errors over 45 degrees do not give the correct results. The real data and default data for the neural network are in Table 3. The coefficient values for every simulation case are in the results Table 4.

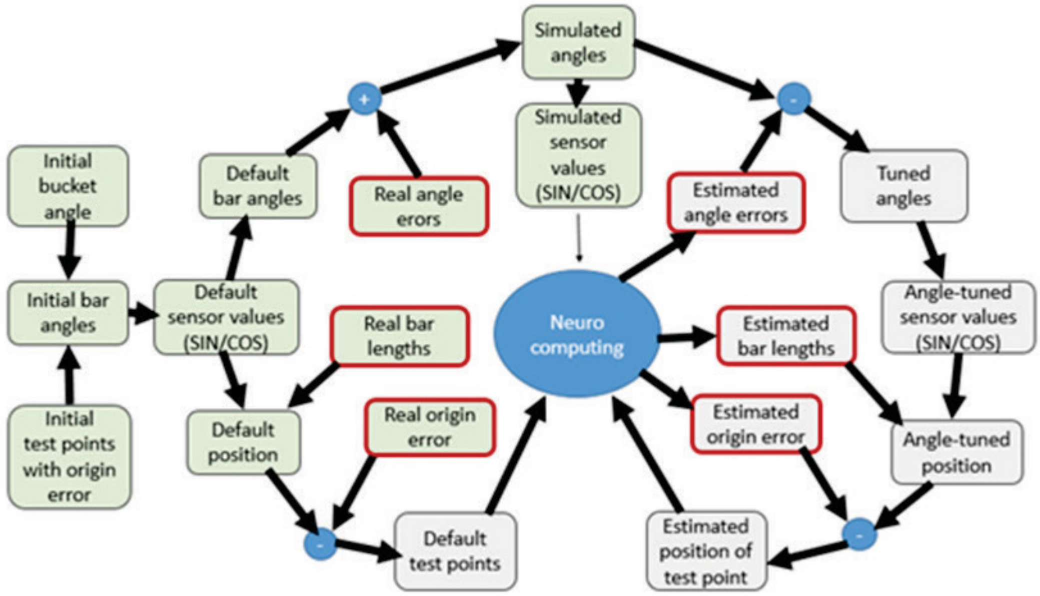


Figure 66. Simulation loop for tuning method

Table 3. Real and estimated data for simulation input

	Default real data	Estimated data
Angle errors	44, -40, 45 degrees	0, 0, 0 degrees
Bar lengths: main, arm, bucket	442, 222, 207	490, 180, 270
Origin error: x,y	-50, 30	0, 0
Test points x,y	150, 30 350, 30 450, 30 550, 30 750, 30	200, 0 400, 0 500, 0 600, 0 800, 0

7.4.4 Simulation results of tuning method

All simulations continue until the difference between the real and the estimated value is under 0.1 or as in the first 3-point test, the values are stable. The simulation includes three cases: In the first case, only the angles and origin error are unknown. In the second case, the length of the bucket is also unknown. In the third case, the lengths of all of the other bars are also unknown. In every case, the reference test point coordinates are fixed and known. Each case includes three parts with 3, 4 and 5 test points, each with different coefficients found in trial runs.

The output of the neural network computing is the condition: did it find the right features and how many iteration loops it needed.

As an example, in Figure 67 there are curves of the estimated angles and the origin position of the first case with 4 reference test points. It takes 179 iteration loops to get an accuracy of 0.1.

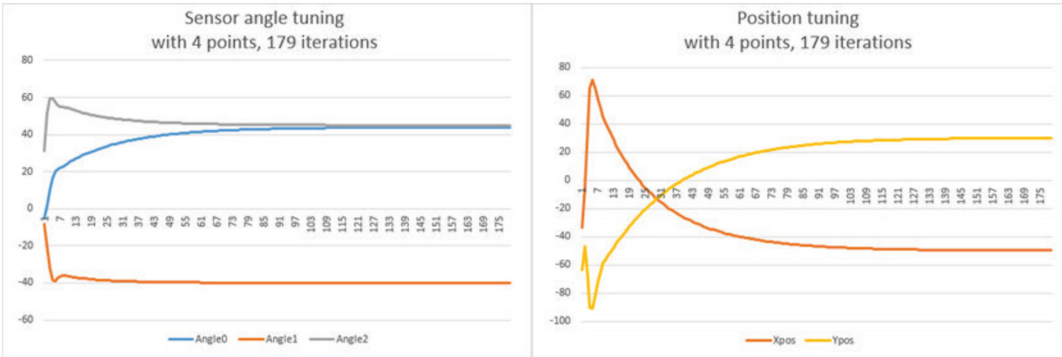


Figure 67. Case 1 with 4 test points

In the example of the second case in Figure 68, with only 3 points and an unknown bucket length, it takes 25673 iteration loops until the results are accurate enough.

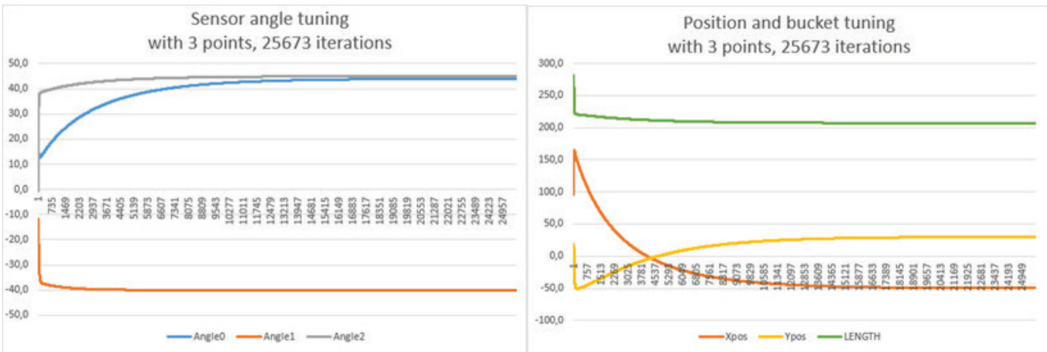


Figure 68. Case 2 with 3 test points

In the most demanding case, where all bar lengths are unknown, the neural network learning phase gives the wrong results with just 3 reference test points. While using 5 test points, it also found, in addition to the angles and the right origin, the lengths of all unknown bars in 5688 iteration loops, as shown in Figure 69.

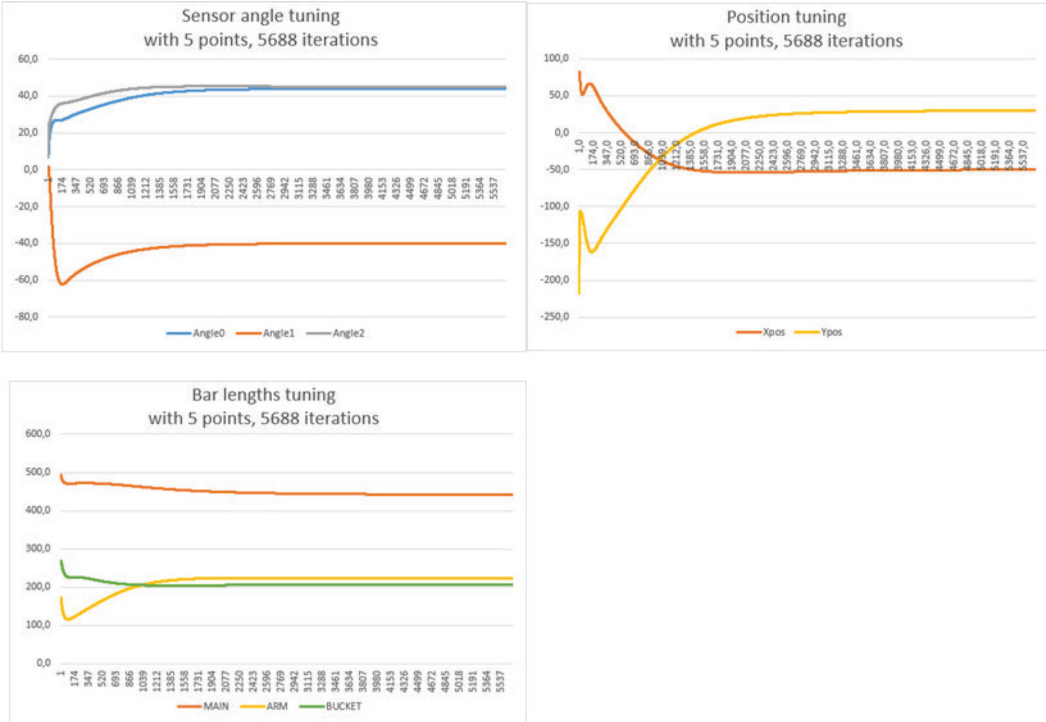


Figure 69. Case 3 with 5 test points

All the results of the three different cases, with 3, 4 and 5 test points, are in Table 4. As seen in the results, it takes a lot of calculation power to also find the lengths of all the bars in case 3. Therefore, the tuning method in case 3 is suitable for the first installation of the depth measuring system of the excavator. It perhaps requires a more powerful computing device than a simple depth display. While the depth measuring system is ready to use a simpler processor can calculate features with 4 or 5 test points, as in case 2, if the driver for example replaces the wireless acceleration sensors and changes to a different bucket for a new task.

Table 4. Simulation cases and results

		Case 1	Case 2	Case 3
Fixed features (F) and Unknown features (U)	Test points	F	F	F
	Main bar	F	F	U
	Arm Bar	F	F	U
	Bucket	F	U	U
	Angles	U	U	U
	Origin	U	U	U
3 test points	Iterations	**3599**	**25673**	No results
	Coefficients	0.05/0,35	0.05/0.35/0.1	0.03/0.5/0.01
4 test points	Iterations	**179**	**399**	**11878**
	Coefficients	0.1/0.7	0.1/0.7/0.1	0.03/0.5/0.01
5 test points	Iterations	**117**	**86**	**5688**
	Coefficients	0.1/0.7	0.1/0.7/0.1	0.03/0.5/0.01

The neural network found these results using selected default values for bar lengths, angle errors and origin error in Table 3. When tested with some other default real data, case 3 does not give any results. It seems that the excavator tuning method requires a lot more research work to function in a practical universal excavator application.

8 NEW PROPOSED APPLICATIONS

The wireless network topology defines for which applications the technology is suitable. There are many ambient-based application possibilities with ad-hoc, mesh and flat topology. If the nodes are logically identical without a hierarchy and communication between two random nodes is the normal way of discussion, it is a flexible tool for plenty of new applications. The most significant limit is low speed. This limit is a result of the low-power feature of mobile modules and buttons. As mentioned in chapter 3.4.4, the low-power feature means a long stand-by period, during which communication is not possible. Therefore communication is at a very low-speed. Naturally, in many applications only a few features are needed. If continuous power supply in some nodes is possible, the communication rate rises drastically, but the network becomes hierarchical: communication depends on the powered nodes, which limits the application possibilities.

8.1 GENSEN project study cases

Seinäjoki University of Applied Sciences took part in the GENSEN project to test developed wireless technology. Other partners were the Aalto University and the University of Vaasa. The project took place during 2009-2011. The aim of this project was to create generic hardware and software for wireless automation. Five different pilot cases were realized: radio environment in industry, vibration monitoring in wind turbines, distributed energy production, greenhouse sensors and cow monitoring in a cattle house. The wireless technology developed in Seinäjoki UAS was called the SurfNet-platform. (Virrankoski, 2012), (Palomäki, 2011a).

8.1.1 Wind turbine case

In this case a radio button (SurfNet node) monitors wind turbine blade vibrations with a 3D acceleration sensor. A sink node was connected to a laptop via a USB-SPI bridge module. The distance between the nodes was about 20m. In this case, the aim was to test the range of the wireless link and how suitable the SurfNet node is when monitoring wind turbine blades. The setup for this case is in Figure 70 (Palomäki, 2011a) (Virrankoski, 2012) (Palomäki, 2011b).

Figure 70. Blade vibration monitoring test setup (Virrankoski, 2012)

When the blades were not rotating, no transmission errors occur and the sensor monitors only the wind effect on the blades, as shown in Figure 71. Measures accelerations, while no rotation (Palomäki, 2011a)

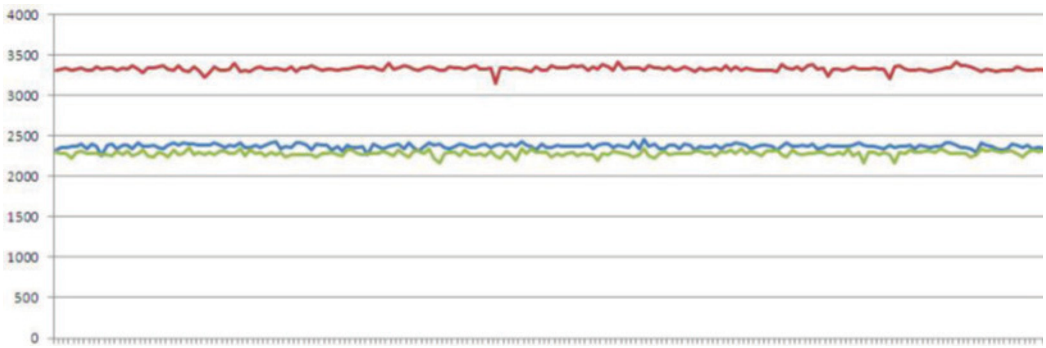


Figure 71. Measures accelerations, while no rotation (Palomäki, 2011a)

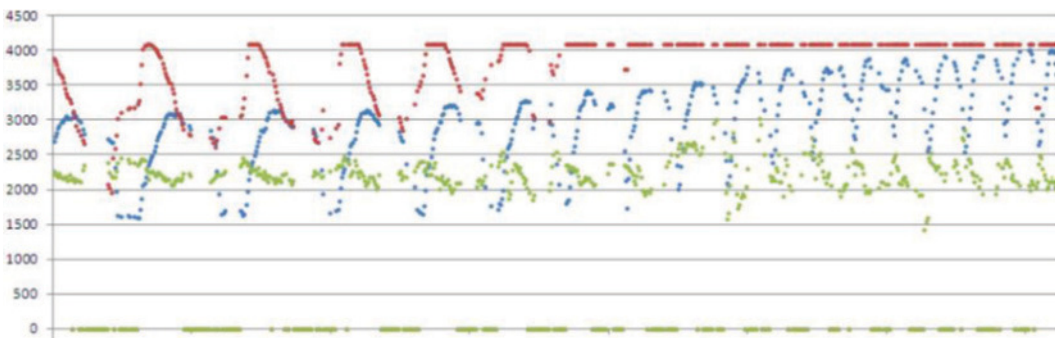


Figure 72. Measured accelerations, while rotating (Palomäki, 2011a)

The acceleration sensor has a 1.6 g limit in the vertical direction, and so this signal saturates as the blade starts to rotate (Figure 72). The controller has a 12 bit A/D conversion, so the maximum value at 1.6 g acceleration was 4095. A transmitted packet loss of 33-36% existed, because the wireless link does not work when the node in the blade was behind the turbine. The loss can be reduced with an extra routing node at the top of the turbine or by buffering the measured data, which is then transmitted when the wireless link is working again. (Virrankoski, 2012).

8.1.2 Greenhouse case

At the beginning of the GENSEN project, the SurfNet nodes were tested in a greenhouse environment. The aim was to test the energy harvesting feature and mesh-topology functions. The node was encapsulated in a class tube to protect it from dropping water. It includes temperature and humidity sensors and a solar panel (Figure 73). The soil water content was measured indirectly by air humidity using a closed glass tube partly in the soil.

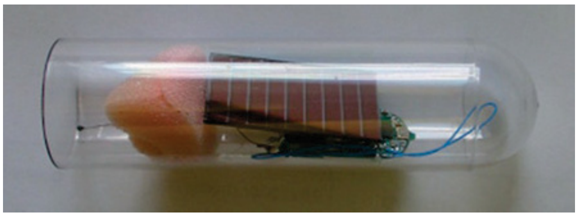


Figure 73. Greenhouse sensor (Palomäki, 2013, ss. 292-293)

The sensors were usable in a greenhouse. The transmission range was shorter than expected: only 1-5m compared with an open-air range of 10m. Perhaps the tomatoes absorbed a significant amount of the radio signal; the diameter of a tomato is near the size of 2.4GHz antenna. It means that the tomatoes absorb the radio signal. The solar panel needs quite high light intensity to work properly. The sensors were uncalibrated and the temperature sensors functioned quite well but the humidity sensors needed calibration (Figure 74) (Palomäki, 2011a).

Node	Temperature	Humidity
3	18.9 °C	45.1 %
4	19.3 °C	45.1 %
6	19.6 °C	
7	19.3 °C	46.8 %
8	19.3 °C	41.7 %
10	19.3 °C	44.0 %

Node	Humidity
2	71.0 %
5	65.4 %
9	71.6 %

Figure 74. Sensor results from air (left) and soil (right) (Palomäki, 2011a)

8.1.3 Cattle house case

The aim in the cattle house case was to test the mesh network topology, gossip routing, the relative positioning method and usage of acceleration sensors. The wireless network consists of 12 fixed-point nodes and some moving nodes fastened onto cow collars, the moving nodes also collected 3D acceleration data. The measured data and node positions flow through the network using gossip protocol, jumping from node to node, until reaching the sink node. A PC collects data from the sink node via a USB-SPI bridge. All data is visible on the PC screen in graphical and numerical format, as shown in Figure 75. (Palomäki, 2011a), (Palomäki, 2014a), (Virrankoski, 2012).

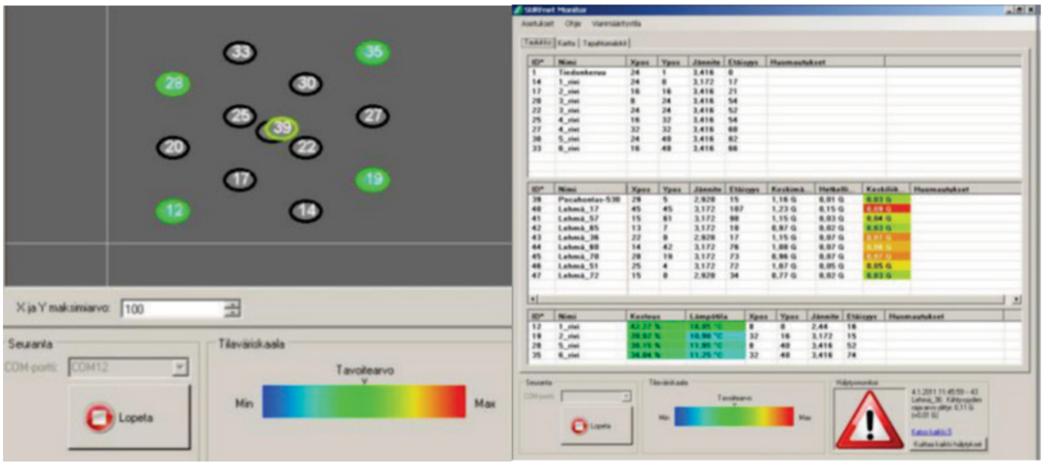


Figure 75. Graphical and numerical view (Palomäki, 2011a)

In Figure 75 node 39 is a moving node and it nearly covers sink node 1. Other nodes are in fixed positions. The numeric display shows the cow's names, estimated positions, battery voltage and activity. Because of the low-power slow gossip-routing method and synchronization, the measured data was visible on the screen after 15-60 seconds. The walls worked almost like plate antennas, so neighbour based relative positioning does not function very well near walls. In the middle of the house, the positioning functions quite well, as shown in Figure 76. Nodes 37, 39, 40, 41 and 42 are moving nodes (cows). The real position of node 41 is a small red point.

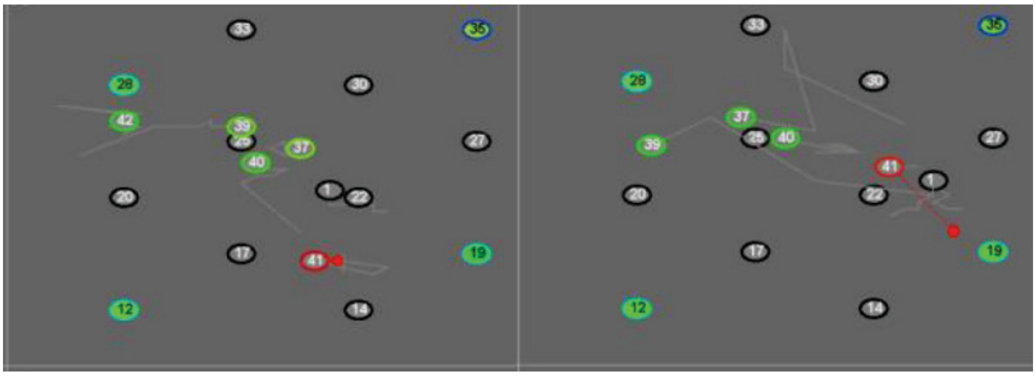


Figure 76. Positioning in the cattle house (Palomäki, 2011a)

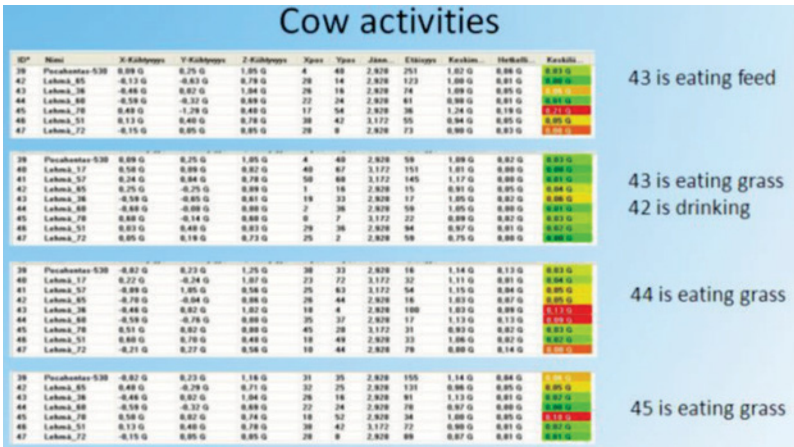


Figure 77. An example situation in cattle house

Figure 77 shows one example situation in the cattle house. One of the measuring problems was that the cows are moving very slowly, so that the acceleration sensor did not recognize the activity very well.

8.2 Hierarchical topology test cases

In this case, there were tests of the hierarchical topology, as mentioned in chapter 6.3.4. The radio controller nRF24LE1 can be in 11 different modes consuming different amounts of power. The most usable modes are described in Table 5.

Table 5. Different modes (Nordic Semiconductor, 2010b, s. 183)

MODE	Power consumption
Deep sleep	0.5 uA
Memory retention using external 32 kHz crystal	1.6 uA
Memory retention, using internal RC oscillator	1.8 uA
Register retention, using synthesized 32 kHz oscillator	87 uA
Active mode	2.5 mA

Deep sleep mode is not usable in this test, because the controller is woken only by an external signal. The next two modes are usable in this test and in real applications. In this case, an external crystal replaced the internal synthesized oscillator to avoid assembling the external low speed crystal and because it functions in just the same way, except for the power consumption. The frequency of the external 32 kHz crystal is very accurate at 32.768 kHz, but the internal RC oscillator has a frequency accuracy of ±10% (Nordic Semiconductor, 2010b, s. 123). It is important to keep the layout as simple as possible, so the focus is on the use of the internal RC oscillator.

In a real ambient intelligence application, the node can move outside of the range and messages are missing. Actually, single random errors are not critical. It is only important that contact does not break continuously when the node is inside the range. In addition, the propagation delays of the messages were tested in the alternative test cases with a different router structure.

8.2.1 Test setup

The test case was made with 20 battery-powered nodes, maximum 3 router nodes and one sink node, which transmits the received messages via SPI-USB bridge to the PC's virtual port interface. Software in the PC collects data from the messages and generates the result reports (Figure 78). The nodes have incremental counters added to the message, so the software can recognise missing messages. In the propagation delay test, a wire was connected between the tested node and the sink node to give a pulse when the message is transmitted. The sink node compares the received pulse time and the received message time and adds a delay time to the message transmitted via the SPI-USB bridge.

In the node, the delay includes the propagation time of the message frame in the air. If a router is used the delay includes the load time from the receiver FIFO buffer, the load time for the transmitter FIFO buffer and the propagation time of the message frame. In the sink node the delay includes only the load time from the receive FIFO buffer. In a real application, A/D conversion of the sensor signals, the building of the message frame and the load time to the transmitter FIFO buffer are also parts of the information propagation delay.

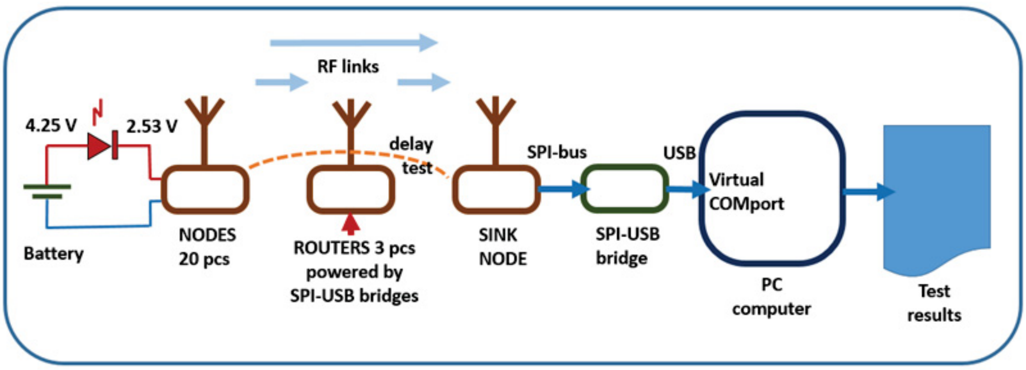


Figure 78. Test case structure

The tested network includes 20 battery-powered nodes shown in Figure 79. The real voltage in one example in the battery is 4.25 V. The power supply range for the radio controller is 1.9-3.6 V. The LED lamp drops the voltage by 1.72 V, so the power supply for the controller is 2.53 V. The LED has one additional function: in sleep mode the LED does not light up because of the very low current. The transmitter current is 11.1 mA and the LED lights up clearly. So the nodes blink for every transmit time, thus also giving the function a visual signal.

Figure 79. Node layout

The router is a node powered by a microUSB via a small SPI-USB bridge (Figure 80). The routers receive and transmit message frames without changing the contents. When the router transmits, it cannot receive. Therefore, it is expected that the router loses and/or delays message frames significantly. The radio controller has three message FIFO buffers separated for transmit and receive. Because of a non-documented error, the controller program cannot read the messages from the FIFO buffers if it overflows. Only by emptying the FIFO buffer after each read routine can the message frame routing continue. This feature can lose also message frames.

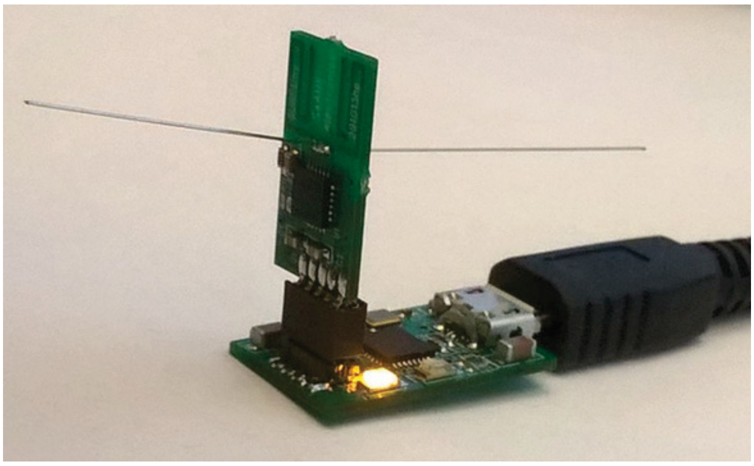


Figure 80. Router with SPI-USB bridge powering

The sink node is connected to a PC via a SPI-USB bridge (Figure 81). The radio controller software does not have a multitasking feature, therefore, while transferring data via the bridge interface to the PC virtual COM-port, the sink node cannot read the receive FIFO buffer at the same time. Because of the FIFO error mentioned above, it is expected that the sink node loses or delays some received messages.

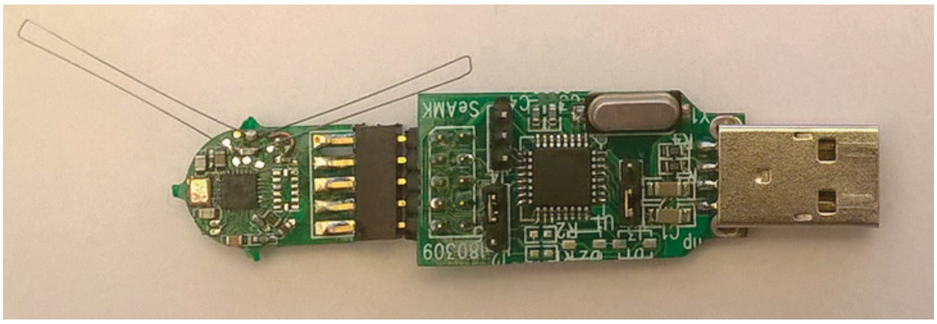


Figure 81. Sink node with a SPI-USB bridge

8.2.2 Star topology

The first test was a simple star topology network without routers (Figure 82) which was synchronized by the internal RC oscillator of the radio controllers. The nodes sent altogether over 220 000 messages during this test. The percentage results are in Table 6. Software calculates lost messages using the incremental counter in the messages in every node. The node can retransmit one message a maximum of 15 times. If it does not receive an acknowledgement, the retransmit counter overflows and the node loses the message.

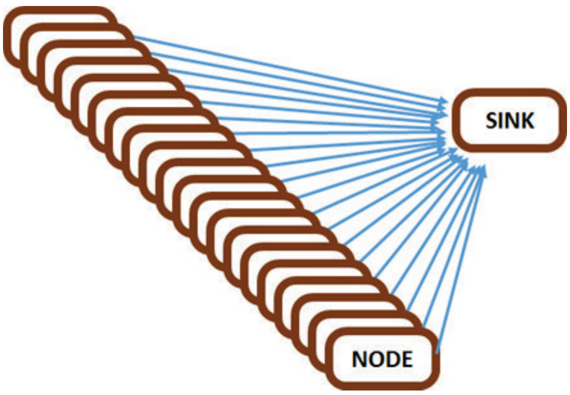


Figure 82. Star topology

Table 6. Star topology test results using RC oscillator

Lost messages total	0.487 %
Lost by nodes because of retransmit overflow	0.312 %
Lost messages by sink node or software	0.174 %
Messages sent without retransmit	86.8 %
Messages needed retransmitting	13.2 %

The test results of the propagation delay are in Figure 83. Every node was tested separately connecting a wire between the tested node and the sink node (Figure 78). The node signals to the sink node via the wire as it starts to transmit the message. The sink node calculates the delay between this signal and the time it actually receives the message. The sink node adds this delay into the message before transmitting it to the PC. The delays of 500 messages were calculated for every single node for this test. The first high peak line describes the messages transmitted without retransmit. The first and second retransmit delays are visible and marked with red lines. The delays between the retransmits can vary. The nodes use 15 different retransmit delays: node 15 has the smallest, node 1 the second smallest and node 14 has the longest. Node 16 has the same delay as node 1.

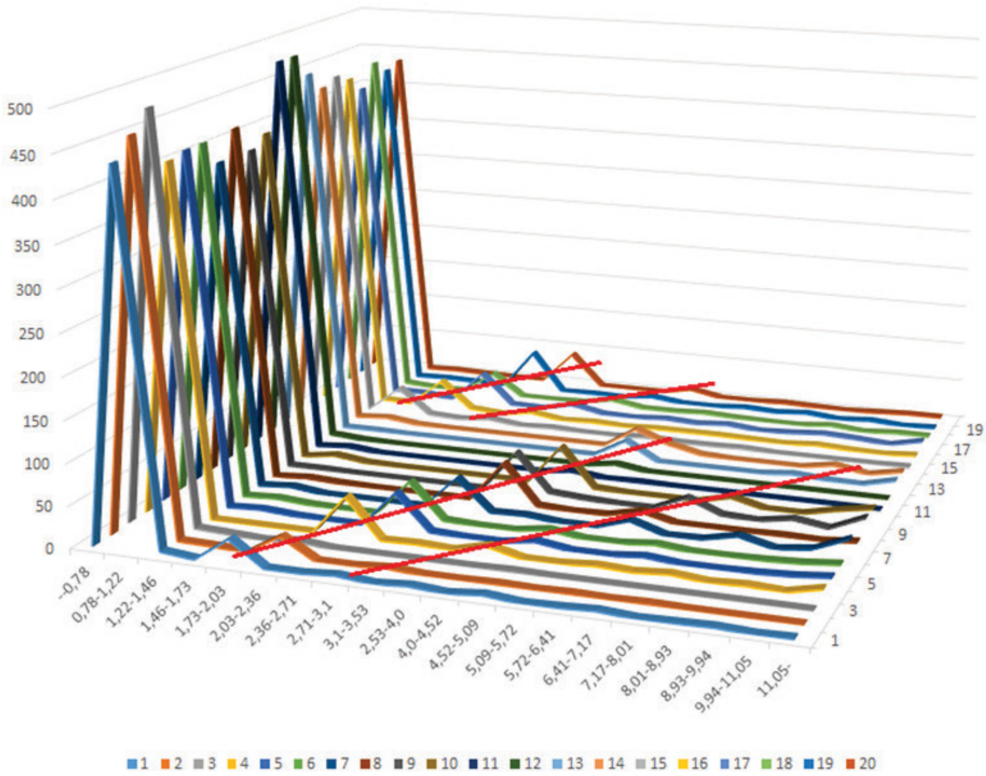


Figure 83. Star topology propagation delays using RC oscillator

8.2.3 Three parallel routers

The next test used three routers connected logically in parallel. The first five nodes sent messages directly to the sink node and the others were in five node groups via 3 routers (Figure 84). A total of over 100 000 messages was sent during this test. As expected, the routers lost messages such that the total lost increments were over 1.5 % higher at the routed nodes when compared with the star topology. The nodes that were connected directly to the sink node lost a lot fewer messages (Table 7).

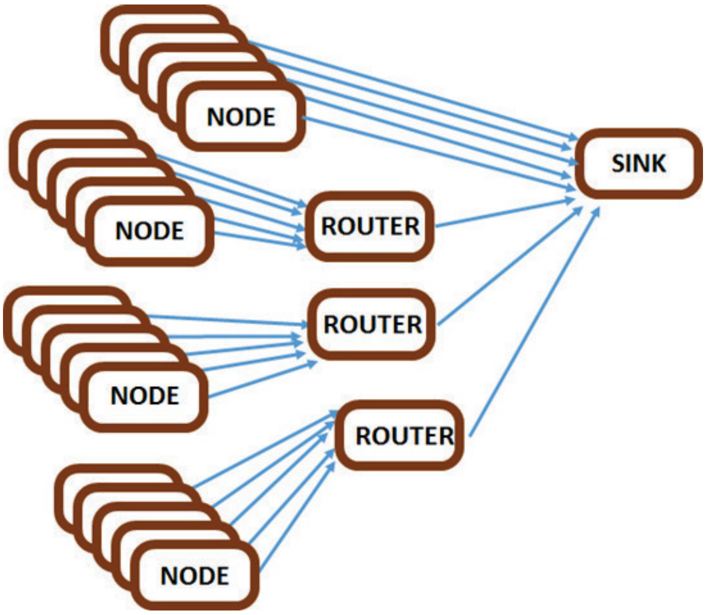


Figure 84. Topology with 3 parallel routers using RC oscillator

Table 7. Topology with 3 parallel routers using RC oscillator

Lost messages nodes 1-5	0.14 %
Lost messages nodes 6-20	2.00 %
Lost messages in total	1.528 %
Lost by nodes because of retransmit overflow	0.138 %
Lost messages by the sink node, routers or software	1.390 %
Messages sent without retransmit	97.5 %
Messages needed retransmit	2.5 %

In the propagation delay test the delay of 200 messages were calculated. The delays generated by the routers are clearly visible in nodes 6-20 in Figure 85. Also the first retransmit delays are visible.

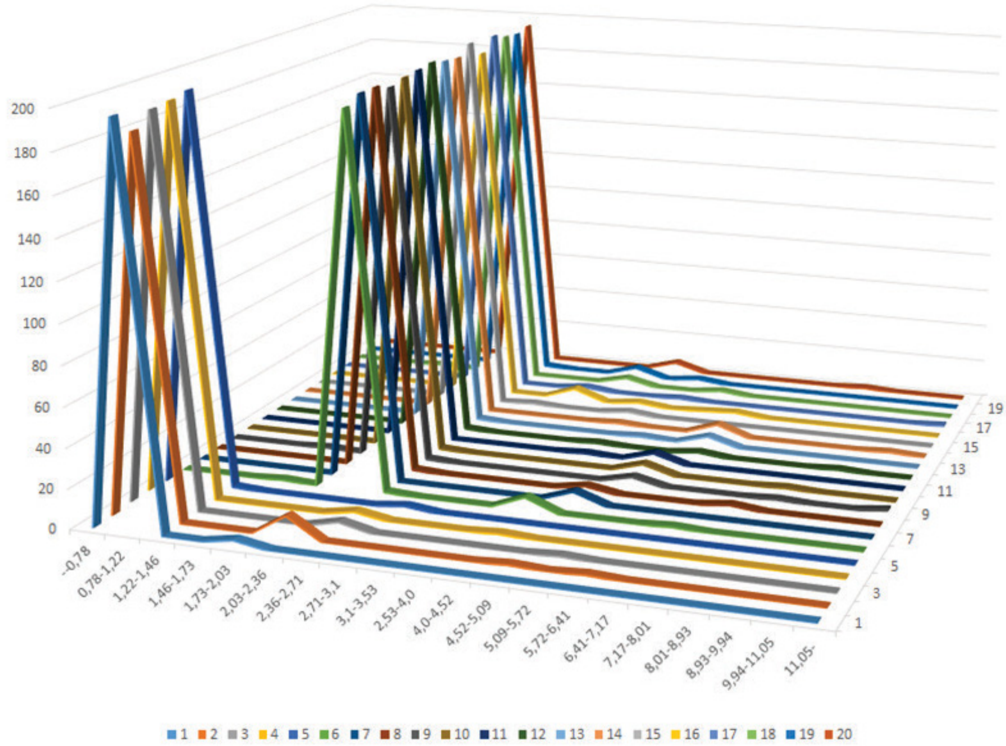


Figure 85. Propagation delays with 3 parallel routers using RC oscillator

8.2.4 Two parallel, one serial routers

In this test case, five first nodes are connected directly to the sink node. The next five nodes are connected to the sink node via one router. The last nodes are connected to the first router via two other routers in groups of five nodes (Figure 86).

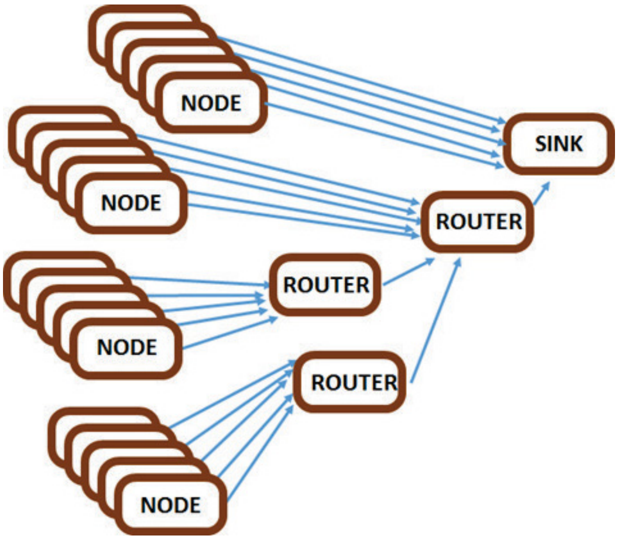


Figure 86. Topology with 2 parallel and 1 serial router using the RC oscillator

The error rate increases significantly via two routers (Table 8), but the messages seem to be lost during the test very randomly. It means that the connection breaks are usually no more than 2-3 seconds. However, the error rate via the directly connected nodes 1-5 keeps quite low.

Table 8. Topology with 2 parallel and 1 serial router using RC oscillator

Lost messages nodes 1-5	0.19 %
Lost messages nodes 6-10	1.37 %
Lost messages nodes 11-20	4.18 %
Lost messages total	2.447 %
Lost by nodes because of retransmit overflow	0.152 %
Lost messages by the sink node, routers or software	2.295 %
Messages sent without retransmit	96.0 %
Messages needed retransmit	4.0 %

The delays generated by routers are clearly visible in the propagation delay diagram in Figure 87, as is also the individual retransmit delays. Node 20 had some problems in this test case. All the nodes are inside the sink node range, but in collision cases, the distance of the node or the direction of the antenna can significantly cover a weaker signal when compared with other nodes.

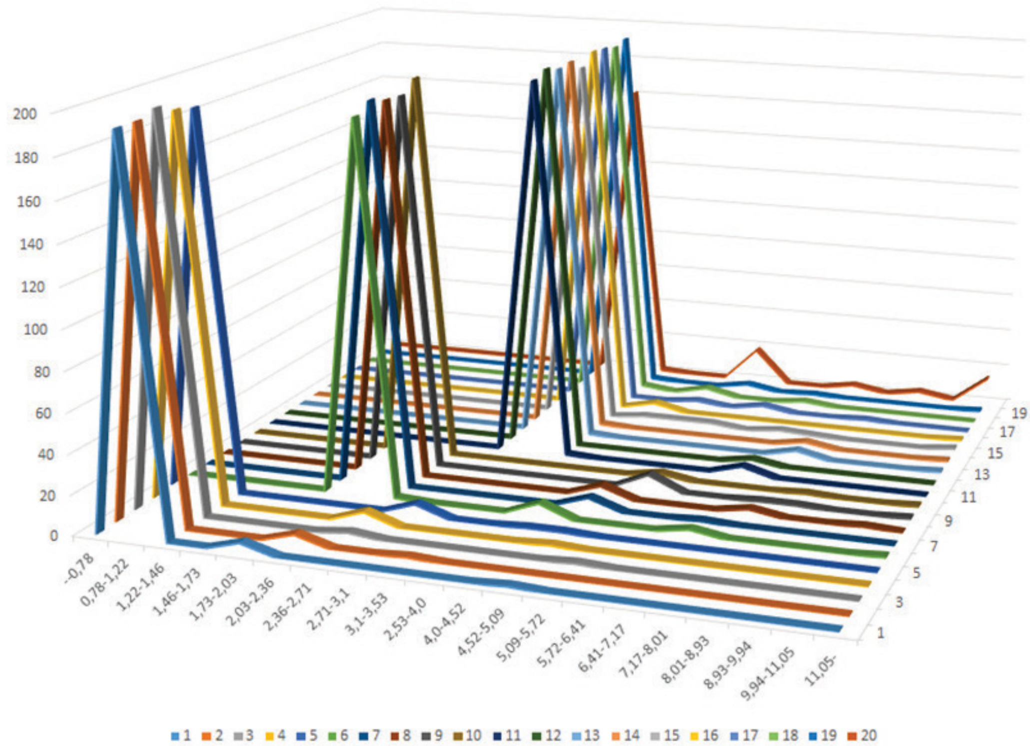


Figure 87. Propagation delays with 2 parallel and 1 serial router

8.2.5 Serial routers

In the worst case, the routers must all work in serial so that some messages were transferred via all the routers. This kind of case was also tested as in Figure 88.

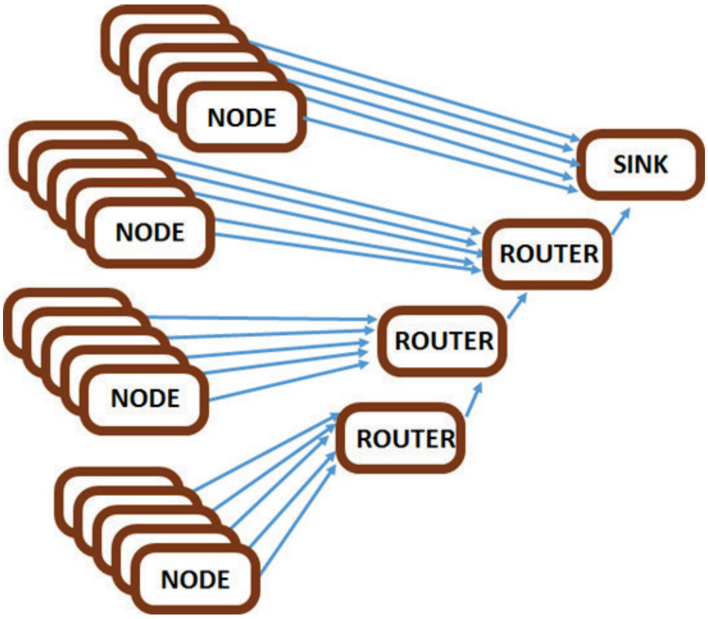


Figure 88. Topology with 3 serial routers

In this case, the propagation via all three routers generates a lot more message losses (Table 9). Again, in this case the messages seem to be lost very randomly, generating only 2-3 second connection breaks. The most noticeable thing is that the portion of retransmits is significant higher than in the other tests. The same feature is visible in the propagation delay diagram Figure 89. The self-synchronization described in chapter 6.3.4, does not work properly, because the nodes behind the routers do not recognize collisions between routers. Sleep time delaying and shifting is not possible between nodes behind separate routers.

Table 9. Topology with 3 serial routers using RC oscillator

Lost messages, nodes	1-5	0.331 %
Lost messages, nodes	6-10	2.736 %
Lost messages, nodes	11-15	6.373 %
Lost messages, nodes	16-20	8.646 %
Lost messages total		4.400 %
Lost by nodes because of retransmit overflow		0.591 %
Lost messages by the sink node, routers or software		3.809 %
Messages sent without retransmit		83.9 %
Messages needed retransmit		16.2 %

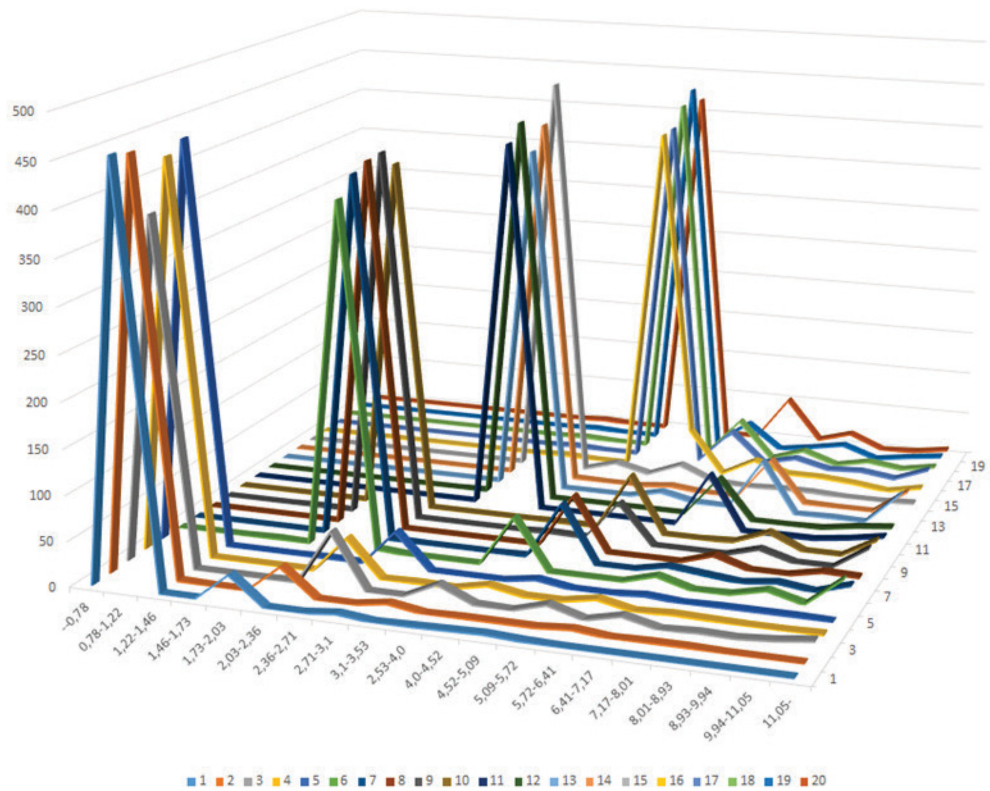


Figure 89. Topology with 3 parallel routers using a RC oscillator

8.2.6 Star topology with accurate synchronization

Sleep time shifting in the case of a collision is more stable if the node timing is more accurate. In this case, the nodes use a frequency synthetized from a 16MHz external crystal. In star topology (Figure 82) all the nodes recognize collisions with others, so it is excepted to produce much better results. As Table 10 shows, the total lost messages drop to about half when compared with the results in chapter 8.2.2. In the test data, it is also possible to see that seven nodes sent all the 500 test messages without retransmitting and without loss, but the sink node or SPI-USB bridge lost some messages. Figure 90 shows that the retransmit concentrated on six nodes, but their message losses are not significantly higher than the others are.

Table 10. Star topology with crystal timing

Lost messages total	0.258 %
Lost by nodes because of retransmit overflow	0.195 %
Lost messages by the sink node or software	0.062 %
Messages sent without retransmit	97.3 %
Messages needed retransmit	2.7 %

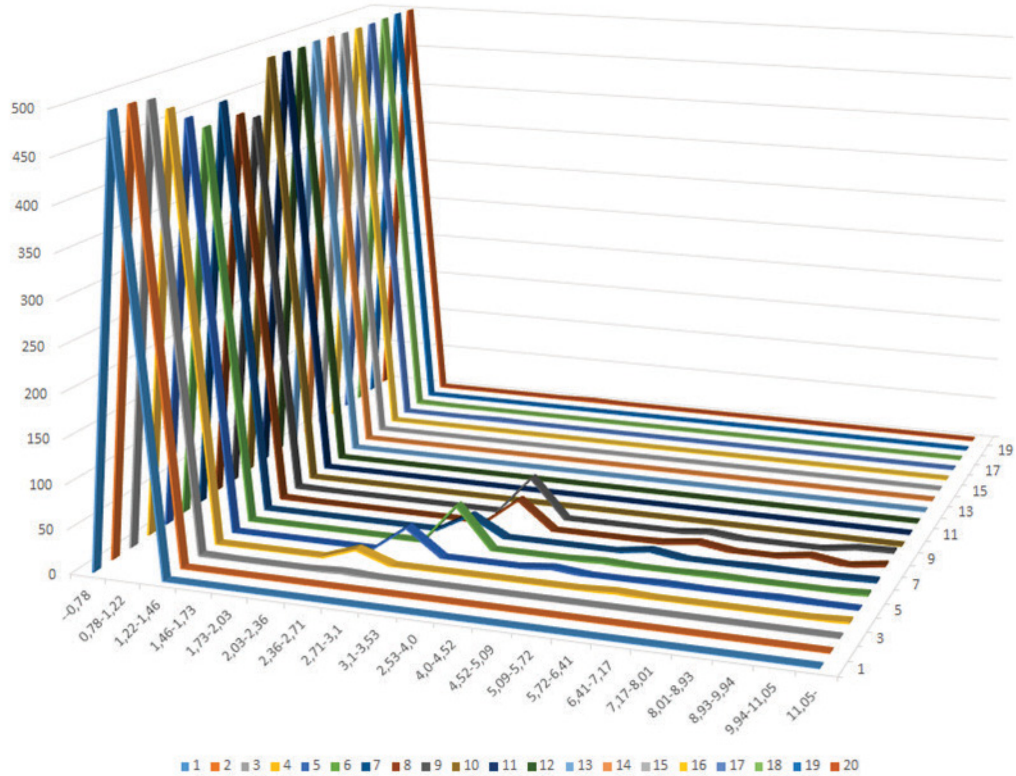


Figure 90. Propagation delays in star topology with crystal timing

8.2.7 Using routers with accurate synchronization

In the next test case, the nodes use crystal synchronization transferred via the routers. In this case, there occurs one critical problem: the connection to some single nodes breaks for a longer time; because the nodes do not recognize collisions behind routers, they do not shift their sleep time timing. The result is that some nodes collided using accurate timing for a longer time period. The stronger signal covers the weaker one without recognizing it. In real applications, it is not allowed to lose the connection to some nodes for a long time. Therefore, this test case was cancelled without any results.

8.3 Excavator tuning study case

Automatic excavator tuning was tested with a simulation in chapter 7.4. A student group at Seinäjoki University of Applied Sciences tested a part of the simulation in practice. The excavator for the test case was the small light model in Figure 91. The python-language program realized the neural network learning in this case. Three wireless acceleration sensors measured the sine and cosine of the angle of every bar. A USB-bridge functions as a virtual COM-port and collects the measured data from the sensors in Figure 92.

Figure 91. Excavator model with wireless acceleration sensors

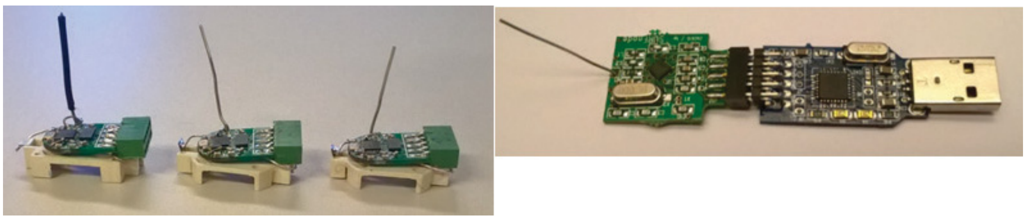


Figure 92. Wireless acceleration sensors and USB-bridge

The user interface of the python language program is in Figure 93. The measured and the tuned angles are visible for all three sensors. Four red points are the reference test points. The model of the excavator shows the real positions of all the bars.

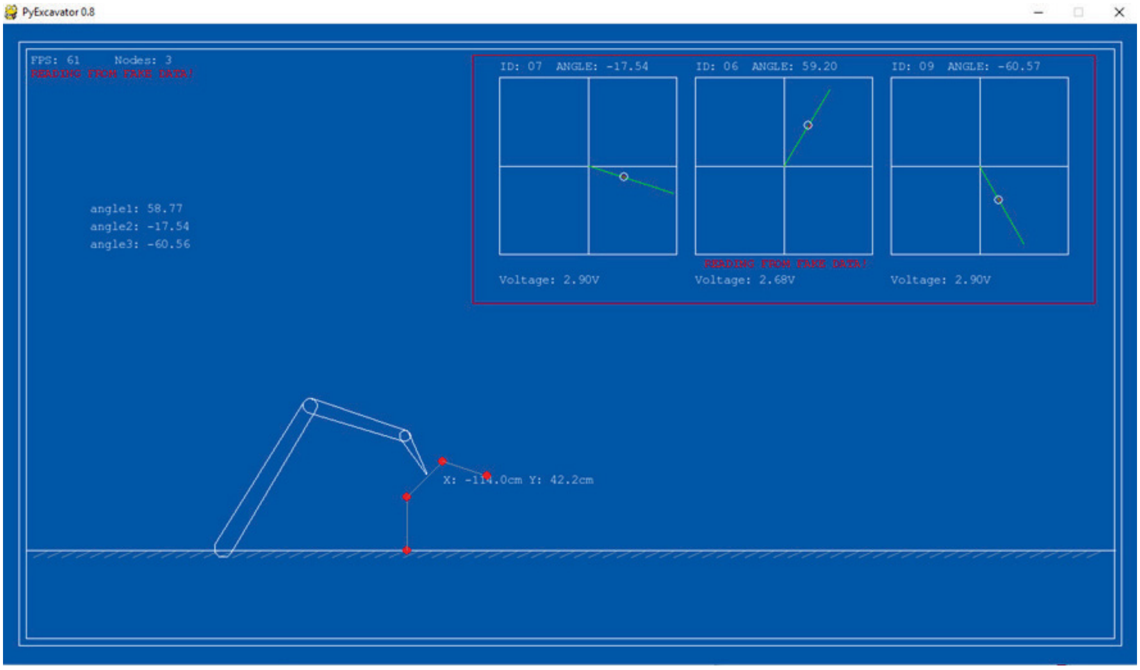


Figure 93. User interface of the python program

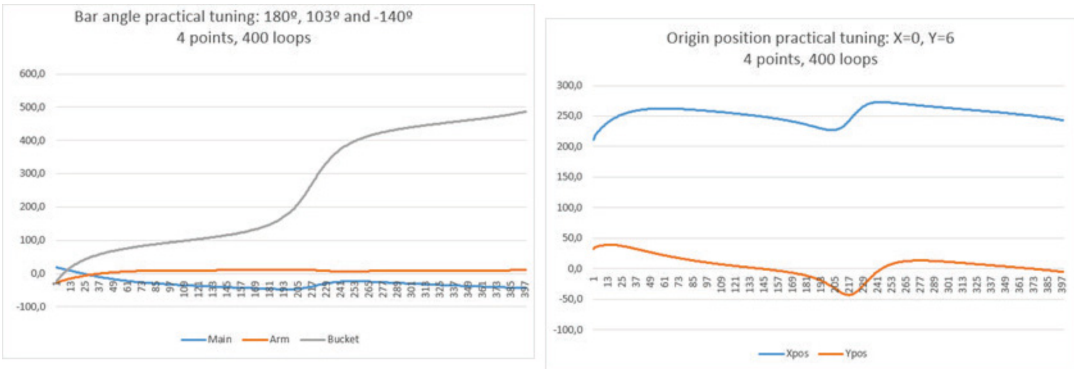


Figure 94. Practical test with 4 reference points and max. ±180°

The setup situation of the test case was that the lengths of every bar were known and measured. The sensor angles and the origin coordinates were unknown and should be defined by the neural network program. The practical test used 3 or 4 reference points for the bucket tip. Three different test cases showed how the practical tuning functions. In the first test, there were four reference points and angle errors: 180°, 103° and -140° and the origin error X=0, Y=6. The neural network did not find the result. The origin position and the Bucket angel did not stabilize (Figure 94).

In the second practical test case, the angle errors were limited to a max ±30°: -5°, 15° and -8°, while the origin errors were X=3.2, Y=1.9. In this case, the neural network found the right results in 573 iteration loops (Figure 95).

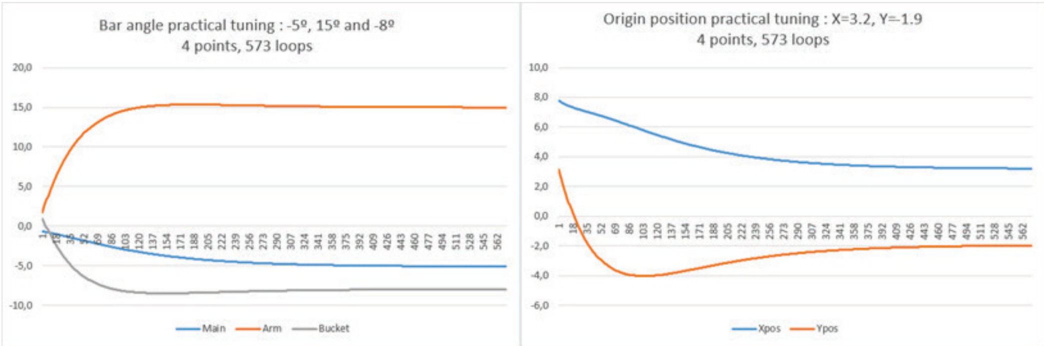


Figure 95. Practical test with 4 reference points and max. ±15° error

The third practical test case includes only 3 reference points, but the angle errors were a max. ±30°: 13°, 15° and 29°. The origin error was: X=8.5, Y=20. In this case, the neural network also found the right results, but needed more iteration loops: 763 (Figure 96).

The practical test cases follows the features of the simulation in chapter 7.4 with maximum angle error limits of ±45°.

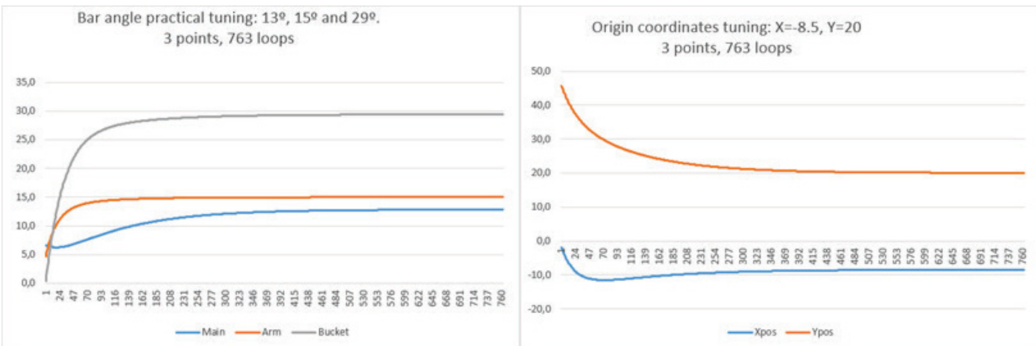


Figure 96. Practical test with 3 reference points and max. ±30° error

8.4 Positioning application possibilities

The functionally of the most demanding network solution is self-positioning mobile buttons. These buttons are low-power battery-driven and they must be synchronized. That means a low communication speed. The layout of the buttons could be small enough to be embedded, for example, in clothes. The use of these buttons could be extremely invisible; the user does not need to know that he/she/it carries the wireless technology (Palomäki, 2007a).

8.4.1 Day-care centre

The concern for children's safety increases day by day. Ambient wireless technology is one solution to this problem. If a child carries a self-positioning wireless button in a wristband, then positioning and monitoring is possible.

The positioning can be realized by installing a line of fixed-position anchor buttons into the fence poles of the day-care centre, around a play field and possibly into a forbidden area as well. All these anchor buttons can be invisible and driven by solar cells or a continuous power supply. The childminder has a hand-held module to monitor the children. The buttons in the wristbands are self-positioning using the fixed anchors and with the neighbours. They transfer their position to the childminder via the network. If the limits of the security area are defined both in the buttons and in the hand-held module, the button can alarm first with an acoustic signal of a non-allowed position. The second alarm is signalling by the hand-held module of the childminder as it receives the non-allowed position information. The third alarm criterion is executed if the hand-held module does not regularly receive the position data of some monitored child.

The self-monitoring buttons give other possibilities to take care of children: The buttons know their neighbourhood, who is close and who is not. The nodes transmit this neighbourhood data via the network to the hand-held module or a monitoring computer, which can calculate the relationship of children. If some child is left alone too often the childminder can take care of this situation before it becomes a visible problem. In addition, other social relationships can be monitored to check the welfare of the children.

It is possible that children do not care about or do not understand the wristband signals. The form of the button can also be a virtual pet, which visually and acoustically displays its welfare. By taking care of this virtual pet, a child actually takes care of him/herself.

8.4.2 Demented old people

The number of old people is increasing quickly in western countries. The institutional care of people with poor health is very costly, so living at home as long as possible is very important in the future. Wireless technology has many solutions to arrange for the home-care of old people. With the new features of the described ambient intelligence, monitoring and control possibilities become more useful.

An acceleration sensor installed into a wireless button gives some new information about behaviour. The sensor can measure movement and inclination, i.e. whether the person has fallen down, is standing or sleeping. Equipped with a system connected to the Internet or to a cellular network the system can alarm instantly if problems or abnormal behaviour arise.

The homes of old people can be equipped with fixed position anchor nodes in different rooms and outdoor areas. The wireless button in a wristband transmits position data to a monitoring device. This system replaces many mechanical switches and smart carpets and is less visible. Because the person does not see the devices and it is not possible to switch them off or pass them by the monitoring of demented people is more secure.

8.4.3 Animal monitoring

In agricultural production, the behaviour of cows in cowsheds is one of the most critical monitoring issues. The cows must be positioned and identified, but the behaviour of cows is a strong indicator of their welfare (Huhtala;Suhonen;Mäkelä; Hakojärvi;& Ahokas, 2007).

Using self-positioning buttons equipped with a 3-dimensional acceleration sensor, it is possible to monitor the routes of a cow, its moving activity and position. This is the base information to estimate the behaviour of cows and to indicate their welfare.

The cowshed can be equipped with fixed position anchor buttons in the corners and on the long walls. The computer collects the position and acceleration data of all the cows; it calculates welfare information and informs about possible problems or special events. This kind of cow monitoring was one subject of a practical study case in chapter 8.1.3.

8.4.4 Watchdog

A watchdog barks at unknown people, not at friends. Security monitoring can take the form of a watchdog, with an infrared movement detector and wireless module. If the RF button in a key chain sends an acceptable code the movement detector does not alarm. If the code is outdated or there is no RF button, the watchdog signals an alarm. This kind of a watchdog is useful, for example, on a golf course, where only paying players can move, or in a garden or in a car, where the existence of an unwanted person is not acceptable.

8.5 Exergame possibilities

Computer games are very popular with young people. With wireless technology for ambient intelligence, the games can be modified to be carried out in real life. These new games are called exercise games or exergames, which means all kinds of games, which motivate people to exercise, and are realized using information technology. *"Exergame and mobile exergame markets are expected to grow significantly within 5 years. At the same time due to unhealthy lifestyles, new demands are directed to new processes of wellbeing"* (Kangas, 2007).

Many computer games increase strategy planning, motivate thinking and give great experiences, but they do not motivate the players to move. Many party games motivate some moving, but it is difficult to control the game rules and events. With wireless technology, hand-held modules and RF buttons can control the strategy of the game. The players have wireless contact with each other and with objects in the terrain. The contacts between the wireless modules are positive, negative, resource handling, building etc. that all help in reaching the goal of the game. The final result of the game is unchallenged and misrepresentation is impossible (Palomäki, 2007a).

8.5.1 Party games

There are thousands of different party games. In the best games, there are some common features: reciprocal effects (sometimes secret) between players, secret roles and the seeking of the guilty. Normally, this kind of game needs a controller or a role selector who is outside the game. The group of players are sitting near each other because they must always be near the game controller. In addition, many times some events are disputed or misrepresented.

A wireless solution for games of this kind can be made using wireless hand-held modules. The modules select the secret roles randomly; everyone knows only his or her own role. The game needs no controller because the software defines the game rules. The players can move freely and the game events themselves are neighbour contacts described in the previous chapter and in chapter 4.2.

8.5.2 Strategy games

Strategy computer games motivate youngsters to play for hours to reach a high experience level. Some of these games are based on area and object conquest by creating a growing population, power and technology. Several players can play against each other or against the computer. A single player controls a virtual person who has different roles and features. If the player takes care of his population in a more versatile and balanced way than the others, his or her population, power and technology increase faster and he or she wins the game.

A wireless strategy exergame means that a player does not control a virtual person, but he or she is this person him/herself. Real people, who run in forests and fields to execute the aim of the game, form the population of the game. Wireless exergames increase significantly social contacts and provide good physical exercise. They also offer a more realistic experience.

Tags in the ambience and hand-held modules realize the wireless solution of an exergame. The tags are virtual material sources, castles or buildings. The hand-held module can have the form of a sword, gun or tool. The game is executed by carrying materials and features between the players and tags and having battles with the enemy.

8.5.3 Orienteering routes

The exergames described above are mostly aimed at children, the youth or people wanting to have experiences. Wireless route tasks are another type of exergames, i.e. exertasks. To execute a traditional orienteering route you need visible control point panels with stamps in the forest and a competition card to carry with you. The results are checked visually and modifications to the route and tasks are difficult to implement.

RF buttons with control tags and with hand-held modules can realize the wireless exertask solution. The hand-held module can make a sound when passing a control point button preventing problems with machines and carelessness. Because the devices are equipped with software, all modifications are simple to implement. For example, every player can have his/her own random route sequence via tags or different route lengths. In addition, the results can be displayed automatically or even in real time.

A special orienteering application is an indoor guidance system for customers in large buildings. A student thesis was made at Seinäjoki University of Applied Sciences to guide patients in the new hospital building in Seinäjoki. The planned guidance system includes a guide sign at every crossing to give personal guide information with direction arrows when the patient is close enough to the guide sign device (Raittinen, 2011).

8.5.4 Learning routes

Learning routes is another type of exertask aimed at student groups or tourists. Traditional learning routes use big information panels with possible questions. If the learning results need to be checked then every person for example needs a paper and a pen for the answers. It is very troublesome to make text or question modifications and the panels can be destroyed over time.

The wireless solution is similar to the orienteering routes, but the hand-held modules must have a better user interface; meaning larger text display and control keys. The wireless modules in objects can have text in non-volatile memory and real time data measured from the object. The text and questions can be updated centrally in an office and the questions can be selected randomly from a large question set. The learning route exertasks are suitable for historical and nature routes and for zoo routes.

8.6 Distributed automation possibilities

Actually, automation is traditionally not ambient intelligence but can use very distributed computing. Automation is becoming more embedded and dynamic in the direction of ambient applications. As described in chapter 2.1.2, distributed automation has many advantages over centralized automation. Very distributed intelligence behaves amorphously, which is described in chapter 4.1.6. Thus very distributed automation can be called amorphous automation.

8.6.1 Traditional automation

Control systems in automation that are implemented in a traditional way can be built by means of programmable logic controllers (PLC). They have been mainly developed for the needs of industry or quite limited applications. If more demanding reasoning and data processing is expected of the system, the PLC:s may not have sufficient intelligence or the system needs a centralized, high power computer unit.

Another way to implement demanding control systems is to use embedded systems developed by local electronics manufacturers. Because the system is built for one application the properties are sufficient, but expensive development charges and the continuity of the product will be the biggest problems in this case: The information about the embedded system is processed by one electronics company. If the activity of the company ends, with the owner's retirement or in bankruptcy, the control system will soon be unusable.

In both situations one more problem remains, one intelligence or processor controls all the functions of a system. If one component breaks down all system activities stop. In applications realized using very distributed solutions, the need in automation is for small, intelligent parts, which can be modified dynamically and where the intelligence of the whole system can be increased fluently (Palomäki 2003).

8.6.2 Investments in amorphous automation

When traditional automation is used, the investments are always considerable. For these investments, a certain repayment period must be calculated and financing must be arranged. In addition, the possibility to expand the system must be taken into account in the investment strategy. There is no point in investing in non-productive capacity, although that is often inevitable.

Figure 97 shows an estimated typical process of automation needs and investment. First, there is an automation need, which it is not worth investing in immediately. When the need to invest is sufficiently great a control system will be acquired where extra capacity is reserved for the future. A non-productive investment must be made in the control system. When the automation needs to grow they will not be immediately satisfied. The biggest weakness is the fact that a compromise must be found between the unsatisfied automation need and futile investment.

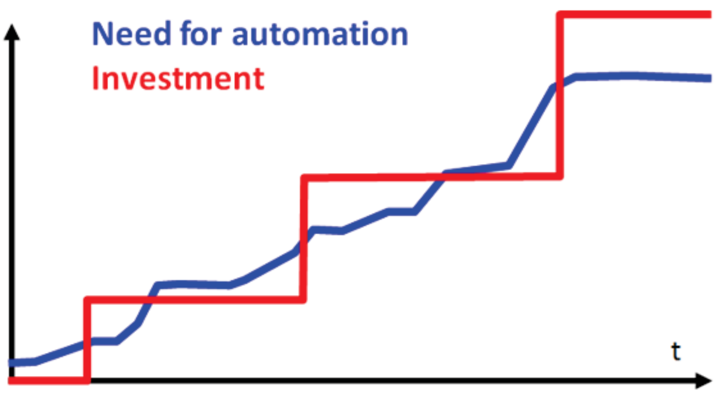


Figure 97. Investments in traditional automation

In amorphous automation, small units can be adopted at a small cost and they are productive immediately. When the automation needs to grow gradually investments can be made little by little. There is no need to buy unused capacity; therefore all investment is immediately productive. Figure 98 shows that there is no need to make non-productive investments, and moreover, there are no significant unsatisfied automation needs. The threshold for adopting the first automated system is commonly high in the private sector and in small businesses. For this reason, another kind of compromise is often made in practice: when one does not automate, one does not have to invest in vain (Palomäki, 2003).

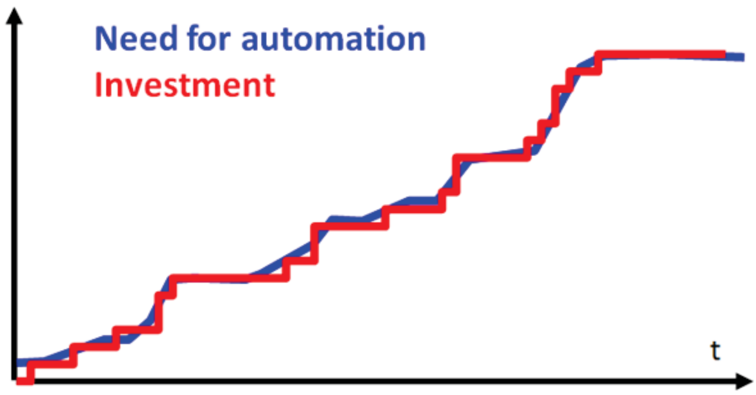


Figure 98. Investments in amorphous automation

8.6.3 Job management in amorphous automation

Amorphous automation has some advantages over traditional automation in maintenance and modifications. When a control system is taken into use, it often happens that the customer has his own comprehension of how the system should operate and how modifications should be made. Similarly, during use new demands and wishes for the system will arise. In a traditional system, the modifications are successful if the control logic has enough extra capacity. If not, extra properties must be built by installing either additional control logic or some extra modules. If the extra properties demand additional control logic parallel to the existing one it is awkward to implement new activities seamlessly along with the old structure. In amorphous automation, the existing and increased units form one entirety, and thus the modifications and new properties are simple to implement.

It is very challenging to create application programming methods in amorphous automation. In simple systems, in which adaptive features and task exchange cannot be realized, the application software is built in the conventional way but is divided into small task parts, which will then be shared between the units. Each unit can also be programmed separately in accordance with the growth of the system and tasks.

If a task exchange feature is required, each unit should have an extensive memory, where it gathers all the task parts of the whole system, which it can complete by itself. According to the information it gathers, it takes the free task that is best executed by the unit. Adaptive features are implemented when the unit supervises feedback and creates a task in a conflict situation that compensates for the conflict and thus carries out the tasks of the system. This kind of system needs extra memory capacity than usual. This kind of a system resembles living cellular tissue, where the tasks are saved as genetic codes and successful mutations create increased properties. A lot of work is still required to achieve these targets and a partial achievement is already an excellent result.

When a task (or role) exchange is implemented, it is possible to reach a very good fault tolerance. The model of the activity in amorphous automation has been taken from a living organism, where a large group of almost identical, independent and intelligent units (cells or insects, for example) carry out all of the system tasks together. When some parts break down, the whole system still functions, because the rearrangements of the work assignment occur in the immediate surroundings of the failed unit and other units take on the task of the missing unit. In the same way, amorphous automation should have reserve resources (units, nodes), which seek failed jobs to execute, and thus the failures in automation are compensated for with reserve units.

8.6.4 Installation of amorphous automation

One principle in ambient intelligence electronics is to be small and embedded. In amorphous automation, this means for example intelligent DIN-rail connectors (Figure 29), which have signal conditioning functions and wireless communication. In a traditional control box, there is a rail of DIN-rail connectors with cables to and from a PLC. In amorphous automation, the control system can be in the form of a rail of intelligent wireless connectors without a separate PLC. Assembly modifications are simple to do by adding a new intelligent connector for a new sensor or actuator without extra wiring to the PLC.

8.7 Object monitoring possibilities

A wireless RF button can replace a RFID (Radio Frequency Identification) tag and offer many more possibilities to positioning and access control. Traditional RFID technology needs a special high-cost reading device. The RFID tags communicate only if they are close enough to a reading device. With wireless RF buttons, which have a normal range of 10 m, the position of the object, including the button, can be monitored and it can start various events from different distances. The reading device can be a wireless USB stick or even another RF button, which is small and cheap as well.

8.7.1 Object positioning in storage

It takes a lot of time to keep a component storage database updated. Mostly components or component boxes are identified by a printed bar-code or store code. The storeman uses manual operations to read the codes and write the store code into a terminal or on a card. The storeman must also add the shelf number to the component data to find it later. The data input and validity depends on a human being and therefore the data is not always up-to-date.

If the objects or the component boxes include self-positioning wireless technology, the data input is done automatically. First, the identification code is transferred into the database via a wireless network when the component is in storage. Second, position data is added into this transferred code, so the database is always up-to-date without a human user interface.

8.7.2 Container positioning in harbours

A similar case is used in large harbours, where the database includes the identification codes and position data of containers. The harbour supervisor must drive around the harbour to keep the database updated. Video cameras and image processing can carry out automatic positioning but it has many error possibilities. With self-positioning wireless buttons, the positions of the containers are automatically updated into a database.

8.7.3 Automatic tool rent storage

Rental activity is carried out traditionally under human control. If the control does not function, the customers can carry objects out without registration or return them without registration. When these little failures accumulate over a long period of time the rental database is completely out of date. The rental activity can be automated using bar-code or RFID tags. However, human control is essential and cumulative errors are possible.

Automatic, unmanned rental activity is possible with low-power RF buttons, for example in a game equipment rental storage. The RF buttons can be implemented into the equpment nearly invisibly with a small solar cell to charge the battery. The storage can have lights to aid the charging process. The buttons transfer their identification codes, e.g. every 5 seconds, to a reading device and so the rental storage database is updated in real time. The rental permission is received, e.g. with a mobile phone, from a centralized database computer. Permitted and unpermitted renting is registered. Return on-time can also be controlled. A device equipped with a strictly directed high-gain antenna can find lost equpment far away from the storage.

A study case of a wireless rental system was made at Seinäjoki University of Applied Sciences as a student thesis. The aim of this case was to control the rental activity in the game equipment storage (Heikkilä, 2009).

9 CONCLUSION

Wireless technology develops continuously. The research described in this thesis is soon old but hopefully it shows a way to future technology, especially in very distributed intelligence suitable for everyday use. Many different operators show the way and drive development in different directions. The target of some operators is to do business and make money. Although the work of some operators creates technology and principles to be the basis of future technology.

9.1 Main results

The ambition of this thesis and all the research and test cases was to open new visions to future technology and possibilities. In the comparison of current standards with new ambient intelligence demands, the standards are not always suitable for simple, low-power and universal network structures. The research shows that it is worthwhile to use open platforms where the developers can define the new features freely without the limitations of standards. Especially, when looking at the possible uses of very distributed artificial intelligence features, flexible features are required.

The applications, which are using artificial intelligence, are linked mostly with high-power supercomputers and highly demanding applications. This thesis shows the feasibility of soft computing. For example, a neural network can be realized with low-end controllers and in simpler commonplace applications.

9.1.1 Practical results

One practical result was the new small-size, low-power platform for test cases. Despite the small size, it was possible to use many different sensors. Simple neighbourhood connections were enough for smart positioning methods. Significant results were seen in the simulations, which show new possibilities to have smart features and applications using the distributed structure of short-range wireless networks. Especially, the simulations of the neural network were promising. The other outcome was the practical study cases, in which the new platform and many simulation results were tested. The cases show that the development, research and simulations were not only suitable in theory, but also in practice. A significant outcome was also the novel visions of future applications with distributed intelligence and the suitability to realize these using developed platforms.

The challenges in the area of health and security will be very important in the future. Position-aware wireless buttons are suitable for connecting objects and people with each other in a secure way. The applications for motivation, monitoring, warning and alarms are the main outcomes of distributed wireless technology when applied to human life style in the future. These kinds of application areas are perhaps the fastest growing information technology area in the future and they have a lot of new challenges to increase the welfare, health and security of people.

9.1.2 Evaluation

The purpose of the research described in this thesis is to build basic knowledge for the research and development process to be executed in the future.

There are some limits in this thesis when choosing the focus of the research work: Specific controller and transceiver chips have been used, and the results of routing and positioning simulations have been made with very specific attributes. Are these limits reasonable? The future will show when the technology is realized in applications. With networks and inside positioning in areas with other sizes, the results may not be the same because of the single simulation and study case environments. To get a better conception, it is necessary to research more solution based structures and protocols. However, these results indicate the possibilities of very light but intelligent wireless network structures. Especially when simulating the positioning methods the parameters are roughly estimated (i.e. correction gain, push/pull ratio, random moving method). A usable method could be found as a combination of the mean value method and the active iterative method.

9.2 Discussion

The research and development of the new technology needs manufactures and resources. The development of semiconductor technology offers new possibilities continuously. Often development seems to be a race between operators, time and abusers. It means the race between new possibilities and security.

9.2.1 Development drivers

The most powerful driver for all development work is money. The developers want to have a profit for their investment. This skews the development direction. In wireless standards, only these members can accredit new products, those who have paid the membership fees. Therefore, the standards do not take into account the requirements of small companies and private operators as producers.

Monetary control also has an effect on the network structure. For example, the organization of the hierarchical cellular network can better collect fees than in the flat structure in a D2D network, where devices discuss with each other directly without control.

Globalization is another important development driver. The control of different systems is more and more independent of the place where the system and operator is. All systems should be connected to the global Internet to be accessible all over the world. IoT is the keyword for this kind of network usage. In addition to the advantages, it offers some problems also, which are discussed later.

As an opponent for the above-mentioned manufactures, there is a fast growing independent manufacturing group for all high-technology development: the private amateurs. People today have more and more free time. High-technology development is one growing hobby for free time. Because of lots of time, they deeply research some narrow areas and without control (of money), they can work independently and find totally new ideas and products. One example of the 'private' product is the Linux operating system. This kind of open source product appears at a prolific frequency. This research and development resource eats the profit of money-based development but drives the technology forward faster.

Military and space technologies are making demands in a different way. Because of the uncertainly of connections, devices must act wirelessly and as independently as possible. In history, these development drivers were also the most effective in wireless communication, as in other scientific areas. In these kinds of demanded applications, the network structure is different. Hierarchical connections have a greater role as an initializer, to prepare devices to work independently in problematic situations. The flat mesh network structure is suitable when some part of the network is destroyed. The network structure should be very dynamic so that every node can route messages forward and can replace the missing nodes.

9.2.2 Security

A big challenge in wireless networks is the non-authorized use of network data. In the case of payment cards, the harm can be very large. The use of technology moves faster than the corresponding security of the used technology. Sometimes the security problems are recognized only when the technology is widely in use (Hakamäki & Palomäki, 2015).

In short-range wireless sensor networks, the security problem is not significant. The misuser must be very close and the difference in wired or wireless network is not very big. In the low-power battery powered node, the use of encryption takes a lot of calculation power; therefore, this kind of network can often be without protection. In networks with longer ranges and especially when controlling critical actuators, communication requires some kind of encryption.

In applications that use the Internet, private data is no longer fully protected. For example, while offering IoT applications for domestic appliances, a big motive for the operators seems to be to monitor the behaviour of the customer so as to have more individual advertising.

9.2.3 Possibilities in the near future

Very light wireless networks are suitable for many new applications. The smaller size of the wireless button offers new possibilities to embed them into objects. With a single chip SOC (System On Chip) circuit, as with the nRF24LE1 Nordic Semiconductors, the size of the RF button can be as small as 10 mm in diameter with a current consumption of about 2 µA (Nordic Semiconductor, 2010b).

If the network consists of small RF buttons, the application possibilities are very large. It is possible to build smart clothing, access control of people and objects, security monitoring and build automation using very limited resource nodes. Outside games can also be controlled using RF buttons. With added sensors, RF buttons can monitor the bar positions and movements of different power tools or the behaviour of animals. The condition of a cornfield can be monitored by scattering RF buttons in the field. Widely distributed intelligence is the future layout in new control systems.

Limited resource wireless technology is suitable for the described game solutions. It seems to be possible to create numerous new games and other non-industrial solutions in a short time with developed electronics. The routing protocol is suitable for sensor networks, which can also be mobile. Adding positioning to the monitoring network, it is possible to monitor mobile groups or swarms in a new way. For example, by monitoring a herd of cows it is possible to save lots of work when the temperatures, accelerations (i.e. moving) and positions are collected automatically and unusual cases are filtered for alarm messaging.

Very distributed ambient intelligence is the focus of this thesis. The target was to research and develop features that are as universal as possible for many different application areas. The personal use: security monitoring and exergames, seem to have the biggest demand in the future. However, there are many applications that can use very distributed intelligence, but do not perhaps have the same demands. If the focus of the application area changes significantly, all hardware solutions, routing and positioning methods need to be rethought.

9.2.4 Uncertain future

The demand for distributed intelligence technology and exercise games seems to be high in the near future: automated functions are more and more commonly used in the daily life of people. At the same time, the health of people is deteriorating and they are looking for new more healthy and challenging activities to fill their leisure time. However, the world lives in a new crisis: ecological energy technology and personal security solutions can overcome ambient intelligence research. The centralized security control of all functions can be more important; therefore very distributed intelligence can be too uncontrollable and function too independently to fulfil new security demands.

9.2.5 New technology

The research of this thesis is based on a specific hardware solution. The semiconductor technology is developing fast, so new chips, including a controller, an energy source control and wireless communication, can replace the hardware chosen for this research. New calculation and memory resources can fully generate new principles to create distributed intelligence networks. If limited resource wireless technology is applied to low-power solutions, it can be the next wireless technology for low-cost applications.

If new technology gives the possibility to use nearly unlimited memory resources, the nature of routing and positioning methods will be different: every node knows every route and the distance to every node in the whole network. Every node is able to take any role in the job description of the whole network. This kind of network gives new possibilities for intelligent and fault-tolerant functions.

The atmosphere contains more and more electromagnetic background emissions, so a wireless transmitter can interfere with the existing emissions with zero-power to allow the receiver to detect data bits. In limited areas and short ranges, this can be the new wireless technology as well.

9.2.6 Development of standards

In this thesis it was concluded that the wireless standards are not always suitable for new ambient intelligence applications, because of their limited features and unnecessary resources. If standardization institutes are interested in creating open and free standards for simple distributed wireless communication, the development of ambient intelligence applications can take a big step forward. However, in fast developing technology, such as light wireless electronics, the standards do not encourage the research of technology, but rather its applications and use.

References

Abelson, H., Allen, D., Coore, D., Hanson, C., Rauch, E., Sussman, G. J., & Weiss, R. (2000). Amorphous computing. *Communications of the ACM*, 74-82.

Aiello, R., & Batra, A. (2006). *Ultra Wideband Systems, Technologies and Applications*. Oxford, UK: Elsevier.

ANT wireless. (2017, 1 30). *FIT2 Module*. Retrieved from This is ANT: Available at: https://www.thisisant.com/developer/components/fit2

Atmel. (2016). *Ultra Low Power BLE 4.1 SoC*. Atmel corp.

Atmel Corporation. (2008). *ATtiny24/44/84 datasheet*. Retrieved 7 15, 2008, from Available at: http://www.atmel.com/dyn/resources/prod_documents /doc8006.pdf

Bitcraze.io. (2016, 10 4). *Crazyradio, general discussions*. Retrieved from bitcraze.io: Available at: https://forum.bitcraze.io/

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York, USA: Oxford University Press.

Callaway, E. H. (2003). *Wireless Sensor Networks, Architectures and Protocols*. Boca Raton, FL, USA: CRC Press, Inc.

Chang, H.-T., Lee, C.-Y., & Chih-Yung, W. (2007). Design and modeling of electromagnetic actuator in mems-based valveless impedance pump. *Microsystem Technologies* , 1615-1622.

Cisco. (2017, 2 7). *Cisco Visual Networking Index, 2016–2021 White Paper*. Retrieved from Cisco Visual Networking Index, 2016–2021 White Paper: Available at: http://www.cisco.com/c/en/us/solutions/collateral /service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

Costa, J. A., Patwari, N., & Hero, A. O. (2006). Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 39-64.

Dursch, A., Yen, D. C., & Shih, D.-H. (2004). Bluetooth technology: an exploratory study of the analysis and implementation frameworks. *Computer standards & interfaces 26*, 263-277.

Dynastream Innovation Inc. (2014). *ANT Message Protocol and Usage*. Dynastream Innovation Inc.

Elmusrati, M. (2017). Symbol-multicast mutual coding for massive MIMO broadcasting. *IET Communication, Vol. 11, No. 3, 11.*

European Communications Office. (2016). *The European Table of Frequency Allocations and Applications in the Frequency Range 8.3 kHz to 3000 GHz*. European Communications Office.

Gawlowicz, S. (2014, 12 1). *Smart dust technology could reshape space telescopes*. Retrieved from RIT Rochester Institute of Technology: Available at: https://www.rit.edu/news/story.php?id=51127

Gomez, C., & Paradells, J. (2010, June). Wireless Home Automation Networks: A Survey of Architectures and Technologies. *Consumer Communications and Networking*, 92-101.

Gomez, C., Oller, J., & Paradells, J. (2012, August 29). Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. *Sensors 2012*, pp. 11734-11753.

Graupe, D. (2007). *Principles of Artifical Neural Networks 2nd Edition*. New Jersey: Word Scientific Publishing Co.

Hac, A. (2003). *Wireless Sensor Network Designs*. New York: John Wiley & Sons Ltd.

Hakamäki, T., & Palomäki, H. (2015). Security of RFID-based technology. *14th International Symposium on Ambient Intelligence and Embedded Systems (AMIES-2015)*. Oostende, Belgium.

Hedetniemi, S., & Liestman, A. (1988). A Survey of Gossiping and brocadcasting in Communication Networks. *IEEE Network, vol. 18, no. 4*, 319-349.

Heikkilä, J. (2009). *Langaton vuokrausjärjestelmä*. Seinäjoki: Seinäjoki University of Applied Sciences.

Hitachi ltd. (2003, 9 2). *Hitachi Develops a New RFID with Embedded Antenna μ-Chip*. Retrieved from Hitachi: Available at: http://www.hitachi.com/New/cnews/030902.html

Hoffman, T. (2003, March 24). Smart Dust. *Computeworld*.

Huhtala, A., Suhonen, K., Mäkelä, P., Hakojärvi, M., & Ahokas, J. (2007). Evaluation of Instrumentation for Cow Positioning and Tracking Indoors. *Biosystems Engineering (2007) 96 (3)*, 399-405.

Hämäläinen, T. D., & Hännikäinen, M. (2007, 9 3). *TUTWSN - Wireless Sensor Network Technology, Ultra Low Energy - Full Mesh WSN Features*. Retrieved 9 23, 2008, from Available at: http://www.tkt.cs.tut.fi/research/daci/wia_open /TUTWSN%20public%20description.pdf

Intelligence, G. (2014, 12). Retrieved from Understanding 5G: Perspectives on future technological advancements in mobile: Available at: https://www.gsmaintelligence.com/research/2014/

Kalyani, V. L., & Sharma, D. (2015). IoT: Machine to Machine (M2M), Device to Device (D2D) Internet of Everything (IoE) and Human to Human (H2H): Future of Communication. *Journal of Management Engineering and Information Technology*, 17-23.

Kamal, A. E., & Al-Karaki, J. N. (2004). Routing techniques in wireless sensor networks: a survey. *IEEE Xplore, release 2.5*, 6.28.

Kangas, S. (2007). *Exergame*. Retrieved 7 8, 2008, from Available at: http://www.vtt.fi/exergame

Kindt, P., Yunge, D., Diemer, R., & Chakraborty, S. (2014). *Precise Energy Modeling for the Bluetooth Low*. München: Technische Universität München.

Kohonen, T. (2006). Self-organizing neural projections. *Neural Networks, Volume 19, Issues 6-7*, 723-733.

Kohvakka, M., Suhonen, J., Kuorilehto, M., Hännikäinen, M., & Hämäläinen, T. D. (2007). Energy-Efficient Medium Access Control Protocol for Dynamic Wireless Sensor Networks. *Ad Hoc Networks Journal*.

Liu, D., Ning, P., & Du, W. K. (2005). Attack-resistant location estimation in sensor networks. *Information Processing in Sensor Networks, 2005. IPSN 2005* (pp. 99-106). Piscataway, NJ, USA: IEEE Press.

Marttila, L., Andolin, M., Kautonen, M., Lyytinen, A., & Suvinen, A. (2007). Älytekniikan jalkauttaminen -hanke. In L. Marttila, M. Andolin, M. Kautonen, A. Lyytinen, & A. Suvinen, *Uutta luomassa* (pp. 35-40). Tampere: Tampereen yliopisto, Tampere University.

MaxStream. (2007). *Demystifying 802.15.4 and ZigBee*. MaxStream inc.

Mulligan, G. (2007). The 6LoWPAN Architecture. *EmNets 07 Proceedings of the 4th workshop on Embedded networked sensors* (pp. 78-82). New York, USA: ACM.

Mäkelä, P. (2008). *Local Positioning Systems and Indoor Navigation*. Tampere, Finland: Tampere University of Technology.

Nasipuri, A., & Li, K. (2002). A Directionality based Location Discovery Scheme for. *WSNA'02* (pp. 105-111). Atlanta, Georgia, USA: ACM.

NFC Forum. (2016, 10 4). *NFC Forum*. Retrieved 9 23, 2008, from NFC Forum: Available at: http://www.nfc-forum.org

Nilsson, R. (2013, 10 3). *Shaping the Wireless Future with Low Energy Applications and Systems*. Retrieved from connectBlue: Available at: http://www.connectblue.com/press/articles/

Niskanen, V. A. (2003). *Sumea logiikka, kirkasta äly ja mallinnusta*. Vantaa: Werner Söderström Oy.

Nordic Semiconductor. (2007). *nRF Single Chip 2.4GHz Transceiver Product Specification*. Nordic Semiconductor ASA.

Nordic Semiconductor. (2010a). *nRF24AP2 Single-chip ANT(tm) ultra-low-power wireless network solution*. Nordic Semiconductor ASA.

Nordic Semiconductor. (2010b). *nRF24LE1 Ultra-low Power Wireless System On-Chip Solution*. Retrieved 6 19, 2008, from Available at: http://www.nordicsemi.no/files/Product/data_sheet /nRF24L01_Product_Specification_v2_0.pdf

Palomäki, H. (2003). Swarm intelligence in automation. In S. Riukulehto, *New technologies and Regional Development* (pp. 155-167). Seinäjoki: University of Helsinki, Seinäjoki Institute for Rural Research and Training.

Palomäki, H. (2007a). Light Short Range Wireless Network and its new Possibilities. *Smart Systems 2007*. Seinäjoki: Seinäjoki University of Applied Sciences.

Palomäki, H. (2007b). Node Positioning in a Limited Resource Wireless Network. *IWES2007 6th International Workshop on Ambient Intelligence & Embedded Systems*. Vaasa, Finland: Vaasa University of Technology.

Palomäki, H. (2008). *SURFnet - Researching and developing project of wireless technology*. Retrieved 7 16, 2008, from Available at: http://lompsa.seamk.fi/sulautetut/surf/surf_e.htm

Palomäki, H. (2008). *Wireless Network in Ambient Intelligence - Licentiate of Science Thesis*. Tampere: Tampere University of Technology.

Palomäki, H. (2010). Low Power Synchronization in Wireless Network. *2nd Workshop of Wireless Communication and Applications (WoWCA2010)*. Vaasa.

Palomäki, H. (2011a). GENSEN project, New Platforms and Applications in Wireless Automation. *10th International Symposium on Ambient Intelligence and Embedded Systems*. Chania, Crete, Greece.

Palomäki, H. (2011b). Langatonta tekniikkaa automaation uusiin käyttötarkoituksiin. *SePRO-Seinäjoen ammattikorkeakoulun verkkolehti*.

Palomäki, H. (2013). Connecting Objects: langatonta tekniikkaa älykäässä ympäristössä. In E. Varamäki, & S. Päällysaho, *Tapio Varmola - suomalaisen ammattikorkeakoulun rakentaja ja kehittäjä* (pp. pp 285-299). Seinäjoki: Seinäjoki University of Applied Sciences.

Palomäki, H. (2014a). Kevyt langaton lähiverkkotekniikka - Connecting Objects. In M. Leino, *Teknologiatiedolla tuottavuutta* (pp. pp 69-76). Pori: Satakunnan ammattikorkeakoulu.

Palomäki, H. (2014b). Wirtual World in real Environment. *13th International COnference and Workshop of Ambient Intelligence and Embedded Systems (AMIES-2014*. Aveiro, Portugal: Aveiro University.

Palomäki, H., & Kivikoski, M. (2008). Wireless positioning realized by Limited Hardware. In *Smart Systems 2008* (pp. pp. 40-47). Seinäjoki: Seinäjoen Teknologiakeskus.

Patwari, N., & Hero, A. O. (2003). Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks. *Proc. of 2nd International ACM Workshop on*

Wireless Sensor Networks and Applications (WSNA) (pp. 20-29). San Diego: ACM.

Raittinen, M. (2011). *HospiCaseY: Langaton opastusjärjestelmä*. Seinäjoki: Seinäjoki University of Applied Systems.

Ray, S., Lai, W., & Paschalidis, I. C. (2006). Statistical location detection with sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 2670-2683.

Reghelin, R., & Fröhlich, A. A. (2006). A Decentralized Location System for Sensor Networks using Cooperative Calibration and Heuristics. *MSWiM '06* (pp. 139-146). Torremolinos, Malaga, Spain: ACM Press.

Rincon, P. (2007, 4 18). *BBC news*. Retrieved from Smart dust to explore planets: Available at: http://news.bbc.co.uk/2/hi/science/nature/6566317.stm

Savvides, A., Han, C.-C., & Strivastava, M. B. (2001). Dynamic fine-grained localization in Ad-Hoc networks of sensors. *MobiCom 2001* (pp. 166-179). Rome, Italy: ACM.

Solomon, K. (1965). Proverbs, chapter 6, vers 6. In *The Amplified Bible* (p. 705). Michigan: Zondervan Publishing House.

Somani, N., & Patel, Y. (2012, May). ZigBee: A Low Power WIreless Technology for Industrial Applications. *International Journal of Control Theory and Computer Modelling (IJCTCM), Vol.2*(No 3), pp. 27-33.

STMicroelectronics. (2016, 4). *MEMS and Sensors Smart Motion tracking, IoT and enhanced user experience*. Retrieved from Available at: http://www.st.com/content/ccc/resource /sales_and_marketing/promotional_material/brochure/5c/3f/63/0b/5f/7e/49/68/brmems.pdf/files/brmems.pdf/jcr:content/translations/en.brmems.pdf

Stoleru, R., He, T., Stankovic, J. A., & Luebke, D. (2005). A high-accuracy, low-cost localization system for wireless sensor networks. *Proceedings of the 3rd international conference on Embedded networked sensor systems* (pp. 13-26). San Diego, USA: ACM.

Stoleru, R., Vicaire, P., He, T., & Stankovic, J. A. (2006). StarDust: a flexible architecture for passive localization in wireless sensor networks. *Proceedings of the 4th international conference on Embedded networked sensor systems* (pp. 57-70). Boulder, USA: ACM.

Tampere University of Technology. (2009). *Technology and Performance of TUTWSN - WIreless Sensor Network*. Tampere: Tampere University of Technology.

Texas Instruments. (2016). *CC2630 SimpleLink 6LoWPAN ZigBee Wireless MCU*. Texas Instruments.

Warneke, B., Last, M., Liebowitz, B., & Pister, K. S. (2001). Smart dust: communicating with a cubic- millimeter computer. *Computer, vol 34*, 44-51.

Weiser, M. (1996). *Nomadic Issues in Ubiquitous Computing*. Retrieved 9 16, 2008, from Available at: http://www.ubiq.com/hypertext/weiser /NomadicInteractive/

Ventä, M. (2007). *Langattomien anturiverkkojen reititysprotokollat*. Seinäjoki, Finland: Seinäjoki University of Applied Sciences.

Vertegaal, R. (2003). Attentive User Interfaces. *Communication of the ACM, Vol. 46, No. 3*, 31-33.

Virrankoski, R. (2012). *Generic Sensor Network Architecture for Wireless Automation (GENSEN)*. Vaasa: University of Vaasa.

Wu, H., Wang, C., & Tzeng, N.-F. (2005). Novel Self-Configurable Positioning Technique for Multihop Wireelss Networks. *IEEE/ACM Transactions on Networking*, 609-621.

Zensys A/S. (2006, 4 24). *Softwre Design Specification, Z-Wave Protocol Overview*. Retrieved from Available at: https://wiki.ase.tut.fi/courseWiki/images/9/94/SDS10243_2_Z_Wave_Protocol_Overview.pdf

ZigBee Alliance. (2016, 10 3). *Zigbee Alliance*. Retrieved 9 22, 2008, from Join the ZigBee Alliance: Available at: http://www.zigbee.org/zigbeealliance/join/

Z-Wave alliance. (2017, 2 1). *Z-Wave alliance*. Retrieved from Available at: http://z-wavealliance.org/