UNIVERSITY OF VAASA

FACULTY OF TECHNOLOGY

TELECOMMUNICATION ENGINEERING

Lu Zhonglei

MOBILITY CONTROL IN WIRELESS SENSOR NETWORK —A SIMULATION CASE STUDY

Master´s thesis for the degree of Master of Science in Technology submitted for Inspection in Vaasa, 21st of May, 2008.

Supervisor	D.Sc. (Tech.) Mohemmed Salem Elmusrati
Instructor	M.Sc. Reino Virrankoski

TABLE OF CONTENTS

ABBREVIATIONS

2D	Two Dimension
AES	Advanced Encryption Standard
CDMA	Code-Division Multiple Access
CSMA	Carrier Sense Multiple Access
CSMA/CD	Carrier Sense Multiple Access with Collision Detect
CSMA/CA	Carrier Sense Multiple Access with Collision Avoid
CTR	Critical Transmission Range
CTS	Clear To Send
DSP	Digital Signal Processing
FFD	Full Function Device
FDMA	Frequency-Division Multiple Access
FIFO	First In First Out
GPS	Global Position System
IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial Scientific Medial
KF	Kalman Filter
LMSC	LAN/MAN Standards Committee
MAC	Media Access Control
MDS	Multidimensional Scaling
MIMO	Multiple-Input Multiple-Output
MOS	MANTIS Operating System
NESL	Networked and Embedded Systems Laboratory
NS2	Network Simulation version 2
NSE	Network Simulator Emulator
OS	Operation System
OSI	Open System Interconnection

PAMAS	Power Aware Multi-Access with Signaling
PID	Proportional Integral Derivative
QoS	Quality of Service
RF	Radio Frequency
RFD	Reduced Function Device
RSSI	Received Signal Strength Indication
RTS	Request To Send
SNR	Signal to Noise Ratio
SS	Spread Spectrum
STD	Standard Deviation
STEM	Sparse Topology and Energy Management
TDMA	Time-Division Multiple Access
TS	Time Synchronization
UDP	User Datagram Protocol
WBAN	Wireless Body Area Network
WEP	Wired Equivalent Privacy
WiNCS	Wireless Networked Control System
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
ZC	ZigBee Coordinator
ZED	ZigBee End Device
ZR	ZigBee Router

UNIVERSITY OF VAASA
Faculty of technology

Author:	Lu Zhonglei
Topic of the Thesis:	Mobility Control in Wireless Sensor Network
Supervisor:	Mohammed Elmusrati
Instructor:	Reino Virrankoski
Degree:	Master of Science in Technology
Department:	Department of Computer Science
Degree Programme:	Degree Programme in Information Technology
Major of Subject:	Telecommunication Engineering
Year of Entering the University:	2006
Year of Completing the Thesis:	2008 **Pages:** 88

ABSTRACT: Wireless sensor networks (WSNs) have become one of the most important topics in wireless communication during the last decade. WSNs integrates many different technologies such as in hardware, software, data fusion, and applications. Hence, WSNs has received recently special research activities. WSNs have so many applications in different areas such as health-care systems, monitoring and control systems, rescue systems, and military applications. Since WSNs are usually deployed with large numbers of nodes in wide areas, they should be reliable, inexpensive, with very low power consumption, and with high redundancy to preserve the life-time of the whole network.

In this M.Sc. thesis we consider one extremely important research topic in WSNs which is the mobility control. The mobility control is analyzed theoretically as well as with extensive simulations. In the simulation scenarios, static sensor nodes are first randomly deployed to the decided area. Then a reference trajectory for the mobile node is created based on the observed point phenomena, and the network guides the mobile node to move along the trajectory.

A simulation platform called PiccSIM is used to simulate the scenarios. It is developed by the Communication and Control Engineering Groups at Helsinki University of Technology (TKK). The obtained results from these simulations are discussed and analyzed. This work opens the doors for more real applications in this area in the nearby future.

KEYWORDS: Wireless sensor networks, PiccSIM, Mobility Control.

1 Introduction

Nowadays, with the development of the network technologies, the Wireless Communication Networks are becoming more and more common. Wireless Ad Hoc Network is such a network in which we do not have fixed structure, like base stations in cellular networks. Instead, the network can be self-organized. Since there is no centralized fixed structure in many cases, distributed algorithms are required. Typically the wireless devices in ad hoc network have just scarce resources such as transmission power and bandwidth. Thus, the network must support multi-hop communication to transmit packets flow source to destination.

This Master's Thesis focuses to localization in ad hoc sensor networks. The key idea is how a mobile node is guided to move in a Wireless Sensor Network (WSN). As the scenario, all the static wireless sensor nodes are randomly deployed in 2-dimensions (2D) environment, a reference path is also randomly created and the mobile node should work follow the computed path. Typically, static nodes report the distance measurements and transmit the control commands to the mobile node. Tracking and guidance is done in the centralized knowledge. The scenarios are simulated via the Wireless Networked Control System (WiNCS). A simulation platform called PiccSIM is used to simulate the scenario. It is developed by the Communication and control Engineering Groups at Helsinki University of Technology (TKK). This PiccSIM platform consists of two computers, one is running MATLAB and another is running Network Simulation version 2 (NS2).

The basic idea of the simulation part is shown in the Figure 1.1.

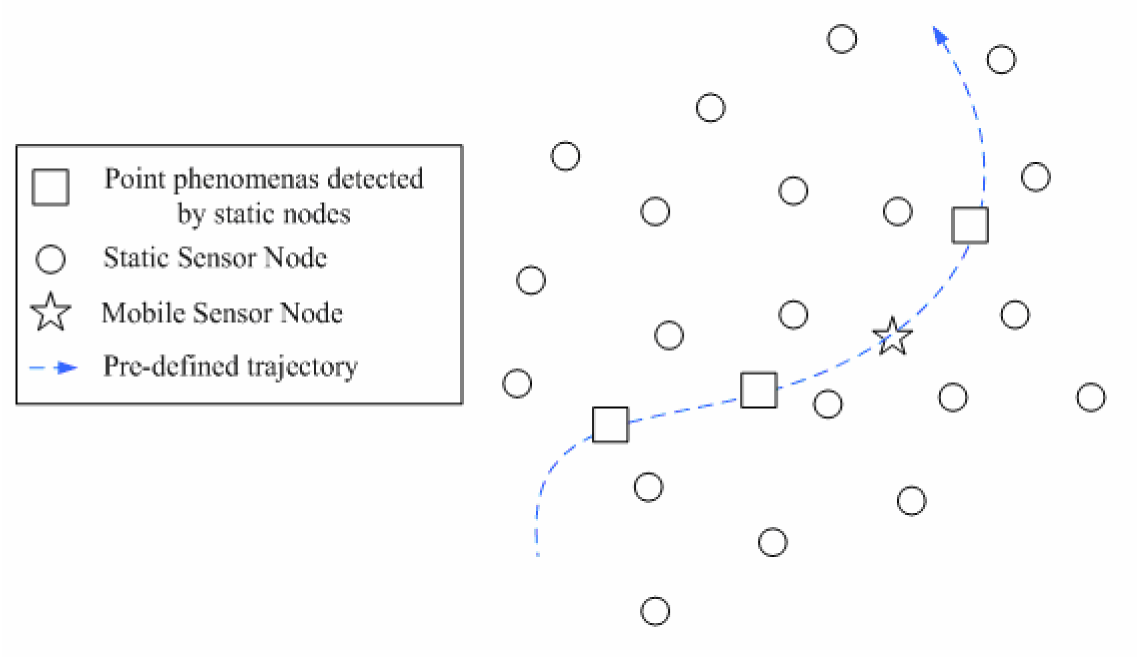

Figure 1.1. The basic idea of the Simulation part

The rest of this Master's Thesis is structured as follows. Chapter 2 presents an overview of the wireless sensor networks. Chapter 3 introduces the PiccSIM simulation Platform which is developed by TKK. Simulations are given in Chapter 4. Finally, conclusions and directions for the future work are presented in Chapter 5

2 Wireless Sensor Networks

2.1 Basics

A wireless sensor network (WSN) is formed by spatially distributed wireless sensor nodes, which by using their sensors to cooperatively monitor environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations (Sohraby 2007: 2).Connections between sensor nodes can be formed by multi-hop paths in which the intermediate nodes are acting as routers. The basic architecture of the typical WSN is shown in Figure 2.1**.**

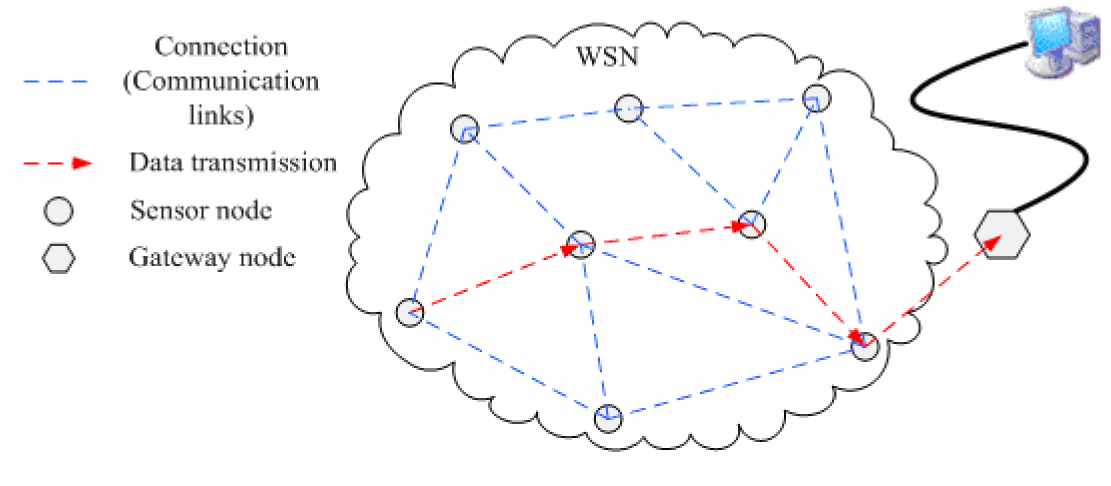

Figure 2.1. The architecture of a typical WSN

Normally, the sensor node builds by 5 basic components:

1. Microprocessor or microcontroller. This part is the heart of the sensor node.
2. A radio for wireless communication. A node can transmit and receive data by using its radio.

3. An external memory, which must be capable to store the simple programming and data
4. The power source, which is usually some kind of battery.
5. One or several types of sensors depending on the requirements.

A typical WSN is built by a group of wireless sensor nodes and they connected to each other over single or multi-hop paths. Each node senses its environment, transmits its own messages and act as a relay to route messages between other nodes. Each node can also perform computation to provide the automatic network initialization, network control and data processing. Even though the localization is not applied in all WSN applications, the node location information is needed in certain applications. Global Position System (GPS) could be one way to solve this problem, but it is too expensive to equip it in every sensor node. On the other hand, GPS or any kind of satellite positioning can not be used indoors at all. Typically, for solving this localization problem, the distances between each node can be estimated by using Received Signal Strength Indication (RSSI) or by using additional hardware to measure them.

Based on the application requirements, the number of sensor nodes can vary from tens to thousands. Usually WSN is capable to self-organize, which makes it suitable to build smart environments. This self-organize includes the automatic initialization, routing and self-hearing, etc. In the sensor networks, the sensor nodes are densely deployed and perform sensing in their sensing radius. Then the sensed data is collected from the nodes to the centralized control for future processing. Some data processing operations can also be performed in distributed manner in the network.

2.2 Radio Propagation

Since all information is transmitted between each node wirelessly, applied radio and short-range communication protocols are important components for sensor network functionality. Thus issues rising from the radiowave propagation must be taken into account in the network design.

The Free Space Propagation under idealized conditions follows the rule:

$$P_r = P_t \cdot G_r \cdot G_t \cdot \left(\frac{\lambda}{4\pi r} \right)^2 \tag{2.1}$$

where P_r is the power of the received signal, P_t is the power of the transmitted signal, G_r is the gain of the receiving antenna, G_t is the gain of the transmitting antenna, λ is the wavelength of the signal and r is the distance between the transmitter and the receiver.

In a realistic application, the received signal always includes additive noise N_0. This additive noise is independent of the position of the receiver and it is also uncorrelated with the desired signal. Usually it is also assumed that this noise is Gaussian distributed and white. The Signal to Noise Ratio (SNR) is an important parameter which represents the quality of the Signal. It is given by:

$$SNR = \frac{P_r}{N_0} \tag{2.2}$$

The free-space propagation model is highly idealized and it does not hold in a realistic environment. The fading and distortion are always caused by the effects of the physical medium. In the computation, the loss parameters can be divided to the ones modeling single path effects and the ones modeling multi path effects.

Two main single path effects are large-scale path loss and shadowing.

1) A **large-scale** path loss in dB is given by:

$$\mathrm{PL}(\mathrm{d}) = \mathrm{PL}(\mathrm{d}_0) + 10 \cdot \mathrm{n} \cdot \log\left(\frac{\mathrm{d}}{\mathrm{d}_0}\right) \tag{2.3}$$

where d is the distance between the transmitter and receiver and d_0 is the reference distance determined from measurements close to the transmitter. n is the power delay index, which varies based on the specific propagation environment. Normally in free space environment $\mathrm{n} = 2$.

2) **Shadowing** is that part of signal attenuation, which is caused by the objects that are big compared to the wavelength of the signal. Shadowing is given by a formula:

$$\mathrm{PL}(\mathrm{d}) = \mathrm{PL}(\mathrm{d}_0) + 10 \cdot \mathrm{n} \cdot \log\left(\frac{\mathrm{d}}{\mathrm{d}_0}\right) + \mathrm{X}_\sigma \tag{2.4}$$

where X_σ is a zero-mean Gaussian distributed random variable with a standard deviation σ .

In natural conditions, there are more than one path available for radio wave propagation. The multipath effects are caused by three different kinds of physical phenomena:

1) **Reflection** occurs when a propagation wave impinges an object which has large dimension compared to the wavelength. (Object $> \lambda$)
2) **Diffraction** occurs when a transmitted radio wave is obstructed by a surface which has irregular edges. The radio wave diffracts around the surface. It can happen even if line-of-sight path does not exist. (at the edge of impenetrable: Object $> \lambda$)
3) **Scattering** occurs when a radio wave meets an obstacle which is smaller than the wavelength. (Object $< \lambda$)

2.3 Applications

There is a wide range of applications for WSNs. Some common ones are listed below:

1) **Disaster Monitoring Applications:**

   In these kinds of applications, the sensor nodes are equipped whit temperature sensor and they can localize themselves. As an example, in a forest environment, the sensor nodes can report the warning message with their location if the temperature exceeds the warning level. Based on some special requirement, the sensor nodes can also detect the chemical situation around their positions.

2) **Military Applications:**

   Wireless sensor nodes can be deployed to the critical areas to form a network to track enemy troops and vehicle movements (Martincic & Loren 2005: 6). WSNs can also be applied to gather more detailed information about friendly and enemy forces, equipment and ammunition. They can also be used for targeting and for attack detection, if nuclear, biological or chemical weapons are used. Sometimes, the disaster monitoring applications can also be used for military purposes.

3) **Medical Applications:**

Nowadays, the WSNs are also used in healthcare. The patients can be equipped with sensor nodes. The hospital's WSN can monitor the patients' health conditions and also track their location (Martincic & Loren 2005: 9), if needed. By using such a system, a patient can also be monitored at home, if he of she is, for example, recovering from major surgery. If dangerous changes are detected in patient's conditions, the network can notify the doctor in time. But the system requirements of this kind of application are really strict, such as the reliability and compatibility with other systems in the hospital.

4) **Environmental Applications:**

The large number of sensor nodes can be deployed to the environment to collect a long-term data on a previously unattainable scale. By doing so the researchers are able to perform deeper data analysis than before. Through such applications, the researchers can follow the environmental pollution easier and save more natural resources than before.

5) **Industrial Applications:**

The idea is to attach sensor nodes to difficult-to-reach areas of machinery. Generally, the cable installation could form 80% of the total costs of the automation system. There are mainly two benefits to use wireless sensor nodes to instead of the traditional cable system. One is the savings compared to cable installation and the other one is accessibility to such a phase that can not be accessed by cables. But the cable maintenance is not that difficult or expensive. Also the throughput of WSN is still very low for many real-time control applications. On the other hand, the system lifetime is another issue. It should be at least 10 years after installation, which is a big challenge in WSN point of view.

Above five examples of the WSN's applications are currently under intensive research. There are also many other WSN applications, for example, the Green house monitoring and control, Logistics and Machine surveillance, etc. Base on several benefits they offer, the WSN applications are expected to become common.

2.4 Transmission Power Control

Two sensor nodes located within a communication range of each other use to have a direct radio link, so that they can communicate directly. A lot of power resources are needed to support long direct links. The number of direct links in WSN is changed every time when the transmission power changes, and this also effects to the network topology. Therefore, the tradeoff between transmission power and network topology must be taken into account

Remarkable savings in the power consumption can be achieved by optimizing the applied transmission power either globally or locally in each node. Generally, the power control can be classified to homogeneous and nonhomogeneous. The classification of power control is presented in Figure 2.2 (Santi 2005: 30).

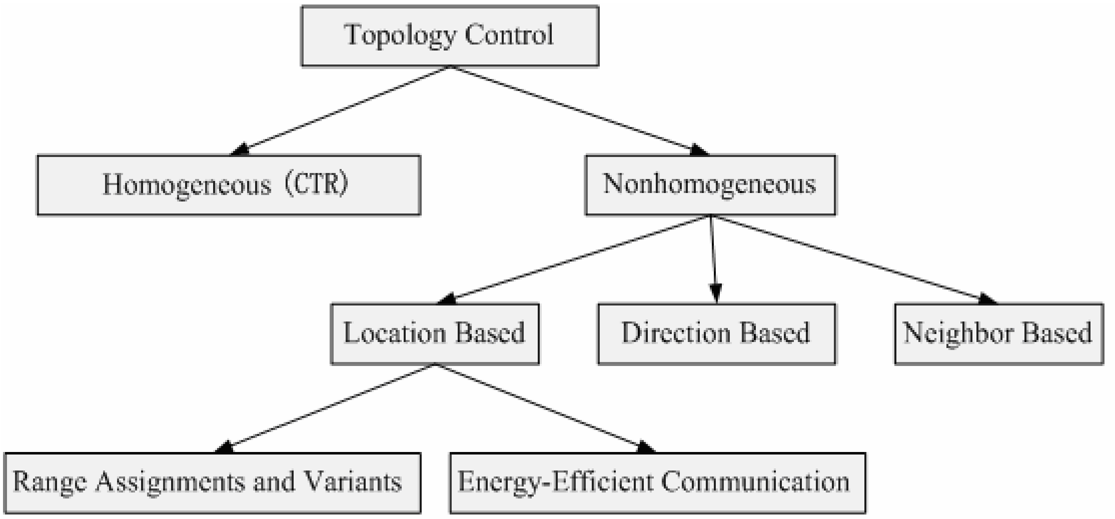

Figure 2.2. The Taxonomy of Power Control

In homogeneous case, all sensor nodes apply the same Critical Transmission Range (CTR). Transmission power is optimized such that the CTR satisfies the requirements of certain applications.

In nonhomogeneous case each sensor node adjusts its own transmission power based on local requirements. As it is shown in Figure 2.2, the nonhomogeneous power control can be further divided into three categories:

- **Location Based Power Control**

  In Range Assignments and Variants case, it is assumed that all sensor nodes know their locations. The centralized network control can utilize the location information to calculate and assign the transmission power. The computation can also be performed in a distributed or locally centralized manner.

- **Direction Based Power Control**

  In this case, it is assumed that the sensor nodes do not know their locations, but they can estimate relative directions to their neighboring nodes. That information is then exploited in power control.

- **Neighbor Based**

  In this case, it is assumed that the sensor nodes only know the IDs of their neighboring nodes and utilize that information in power control

Because of the limited power resources of the wireless sensor nodes, an advanced transmission power control is the most important way to extend the network lifetime.

2.5 Time Synchronization

In some cases, the network can operate in asynchronous time, but in time-critical applications the Time Synchronization (TS) is needed to make the system work properly.

The important requirements for TS in wireless sensor network are following (Elson & Römer 2002):

- Energy efficiency. The TS protocol should not cause too much overhearing.
- Precision, either the dispersion among a group of peers, or error with respect to an external standard must be as small as possible.
- The time period during which the TS holds the required accuracy.

Generally, the traditional time synchronization protocols always try to keep the nodes synchronized all the time meaning that they have to be repeated often and continuously. On the energy cost point of view, these traditional ways are not suitable to the WSN, because frequent periodic repeating consumes too much communication overheard (Karl & Savvides 2005: 206).

In a hierarchically organized WSN, TS should proceed from upper levels to lower levels. According to Martincic & Loren (2005: 14), nodes at the level below the sink node exchange messages with the sink node and adjust their clocks accordingly. In the lower levels, the nodes adjust their clocks with the nodes at the level above them.

There exist time-varying transmission delays as the TS proceeds. TS protocol should keep these time delays as short as possible because otherwise the delays may cause big errors to the synchronized time.

The first non-constant is the sending time, it means the process time starting from when the system plan to send the data to the sensor node ready to send. The second is access time, it means before the transmission starting, the node need wait the access for free wireless channel. The access time delay also depends on the load of the network. The third is propagation time, it means the time during signal transmitted from transmitter to the receiver. The forth is the receiving time, it means the time during the receiver reception of the signal to the whole arrived data. The fifth is the encoding and the decoding time. On the security point of view, it is necessary to cost the time for source node to encrypt the transmitted. At the same time, it also cost extra time for the destination node to decrypt the data. There is also one kind of delay caused by lost packet. In this case, it will require the node to retransmit.

2.6 Power Consumption

Many different types of power sources are used in the sensor nodes, but any types of them are not really optimal. In many WSN applications, the sensor nodes are deployed in such environments that the requirement of the frequent human maintenance is difficult. The life cycle of a sensor node depends on its battery life time. Hence, the energy consumption of the sensor nodes should be optimized to minimum and efficient energy management becomes one of the main challenges in WSNs.

Generally, most of the consumed energy goes to the message transmission and receiving. The main energy consumers in the wireless sensor nodes are shown in Figure 2.3 **(**Savvides 2005).

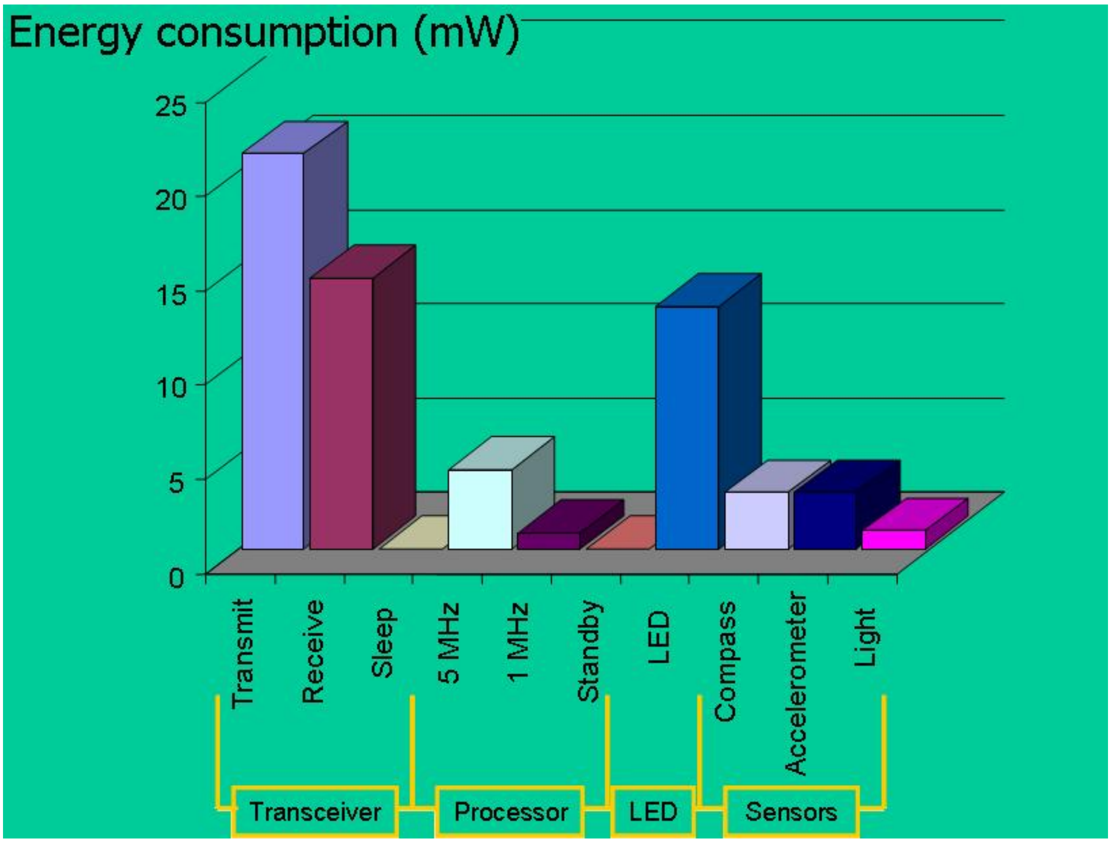

Figure 2.3. The energy consumption in wireless sensor nodes **(**Savvides 2005)

Microprocessor or microcontroller is the centre of the sensor node. It can be set in to three different modes to control its energy consumption. Normal mode means that the CPU works with full power capacity. Idle mode means that the operate state of the CPU is stopped. When this mode is interrupted, the nodes will continue to work. The smallest

CPU power consumption is achieved in the sleep mode. Only the clock runs in the sleep, and the CPU can be wakened up by timer.

Normally, the energy consumption of the memory depends on the size of the memory. If the memory size is large, then it will require more energy resources to support the system reading and storing the data. The memories can be classified to On-chip and Off-chip ones. The On-chip memory consists of CPU and flash memories. It consumes more power than the Off-chip one.

The radio transceiver consists of radio transmitter and receiver. Transmitting and receiving data is the most remarkable source of power consumption as show in Figure 2.3. For this reason, the transceiver should be turned off most of the time and only powered up when needed.

Digital Signal Processing (DSP) needed to analyze the sensed data is also one remarkable source of power consumption. In addition, the sensors also consume some power, but usually their function is small compared to transceiver and DPS.

When the nodes are in the active mode, all components will consume the energy resources in every moment. Therefore, the efficient method to save power is a novel alternation between the normal mode, the idle mode and the sleep mode. It should target to set each node to the sleep mode in most of time and activate it only when it is required.

2.7 Operating Systems

The traditional Operating System (OS) is the software that operates between the application software and hardware. The most common operating systems are designed for the personal computers are workstations. Because of the limited resources of the sensor nodes, these kinds of operating systems are not feasible for them. However, also sensor node needs an operating system to support the hardware functionalities, but it must be designed to operate under the limited resources.

Because of the limited resource for WSNs, the sensor operating systems should be considered designed. Normally, the memory size of each node is limited, so it also requires the sensor operating systems should enough small. The sensor nodes often have memories of only tens or hundreds of kilobytes. On the time synchronization point of view, the sensor operating system should provide the real-time operation. For efficiently assigns the limited memory and CUP time, the resource should be assigned in fairness. The code of such operating system should reliable and efficient since the functionality performed by the sensor nodes may need to be changed after deployment. On the other hand, sensor operating system should provide the power consumption management. It helps to prolong the system lifetime and improve its performance. Also, the system programming interface should be designed follow the uniform standard to support the sensor middleware or application software.

2.7.1 TinyOs

The TinyOS operating system is designed by special software architecture with respect to the constrained resources of the sensor nodes. It allows the application software to directly operate the hardware when required. The system size of TinyOS is less than 400 bytes. According to Levis & Madden (2004), TinyOS is not an OS in the traditional sense; it is a programming framework for embedded systems and a set of components that enables one to built functionalities specific to each particular application.

Tiny OS includes in-build functions listed in Table 2.1. Generally, these functions support some services for the senor nodes by designed interface types. The most common service could be the transmitting or receiving data.

Table 2.1. Core interface support to TinyOS (Levis & Madden 2004).

Interface	**Description**
Clock	Hardware Clock
EEPROMRead/Write	EEPROM read and write
HardwareId	Hardware ID access
I2C	Interface to I2C bus
Leds	red/yellow/green LEDs
MAC	Radio MAC layer
Mic	Microphone interface
Pot	Hardware potentiometer for transmit power
Random	Random number generator
ReceiveMsg	Receive Active Message
SendMsg	Send Active Message
StdControl	Init, start and stop components
Time	Get current time
TinySec	Lightweight encryption, decryption
WatchDog	Watchdog timer control

The first generation of TinyOS operating system was programmed in C language. The infest versions of TinyOS was implemented in the NesC language. The strict checking at the programming time is the most benefits of NesC and it is easier for the designer to develop the TinyOS components.

2.7.2 Mate

Mate is a “Tiny Virtual Machine” which is designed to work on top of TinyOS operating system. It is a single component and sits on several system components of TinyOS including sensors, the network stack, and a nonvolatile storage (Levis & Culler 2002). Mate has two stacks, one is the operand stack and another is the return address stack.

In WSN, Mate functions can forward themselves via simple command. The functions are set in capsules with 24 instructions. All the capsules are classified into 4 subfields which include: message send, message receive, timer, and subroutine. Mate can be used not only as a virtual machine platform for application development, but also as a tool to manage and control the entire sensor network (Sohraby 2007: 277).

2.7.3 MANTIS

The MANTIS Operating System (MOS) is a larger sensor node operating system which is designed to behave similarly as UNIX and provide more built-in functionality than TinyOS (Shucker & Rose 2005: 138). MOS applications are programmed in C language and executed as threads; the MOS kernel includes scheduling and synchronization mechanisms. MOS also includes a framework to allow a high degree of integration between device drivers, network protocols, and the operating system.

2.7.4 SenOS

SenOS is an operating system for wireless sensor nodes developed by the Networked and Embedded Systems Laboratory (NESL) at UCLA (SOS 2.x Home Page 2008). It evolves the basic idea of traditional operating system in such a form that is suitable for resource limited wireless sensor nodes.

SenOS is an operating system which can be modeled as a finite state machine. It includes three components (Sohraby 2007: 280):

- A kernel that contains a state sequencer and an event queue. The state sequencer waits for an input from the event queue.
- A state transition table that keeps the information on state transition and the corresponding callback functions. Each state transition table defines an application. SenOS supports multiple applications in a concurrent manner, by using multiple state transitions and by switching among them.
- A callback storage of call functions. An incoming event will be queued in the event queue list. Base on the (First In First Out) FIFO queue, the first event in the event queue list is scheduled, which causes a state transition and correspondingly, invokes the associated functions.

There are also several other types of sensor operating systems, like: MagnetOS, OSPM and EYES OS, etc (Sohraby 2007: 278-279). The discussed four ones, Tiny OS, Mate, MANTIS and SenOS are currently the most commonly used operating systems in wireless sensor nodes. Each of the sensor node operating systems is still under intensive development to make them better suitable for WSNs.

2.8 Fault Tolerance

One common target in WSN design is that the networks must operate without continuous human intervention. The harsh environmental conditions may easily cause damages to the network. Furthermore, sensor nodes running out of power, it may also cause network malfunction. Therefore, the network must be designed in such a way that it tolerance local malfunctions and a loss of certain degree of nodes. The network should also be able to automatically recover from errors.

A typical source of error is an incorrect state of the hardware or program caused by a failure of a component; the types of the faults can be separated into (Koushanfar & Potkonjak 2008):

- **Permanent faults** which are the ones that is continuous and stable in time.
- **Intermittent faults** which are the ones that occur occasionally due to unstable property of the hardware or as a consequence of a malfunctioning in some part of the system.
- **Transient faults** which are the ones generated by a temporary malfunctioning of otherwise correct hardware.

The fault tolerances in WSN can be divided into two main categories: the Network Fault Tolerance and the Sensor Node Fault Tolerance.

The network Fault Tolerance, it can be considered as a “soft part” of error tolerance. The malfunctioning which occurs in this field is normally caused by the interference of the environment or by the bug in the software. The required level of error tolerance depends on the application. Thus, if the sensor network is used in a critical application, then the required level of fault tolerance is higher. On the software point of view, the usage of parallel versions of software can help to enhance the fault tolerance.

The Sensor Node Fault Tolerance can be considered as a “hard part” of error tolerance. The failures are mainly caused by the malfunctioning in sensor nodes. Also, the complex embedded operating system which is used in sensor node will increase the risk of system failures.

The sensor node malfunctioning can be divided into three categories (Koushanfar & Potkonjak 2008):

a) Systematic calibration error;
b) Random noise error;
c) Complete malfunctioning.

The first two can be addressed through time redundancy. The third one is enhanced using hardware redundancy. The complete malfunctioning is enhanced using hardware redundancy which currently is envisioned for actuators.

2.9 Communication Protocols

IEEE 802 is the official communication standards family which maintained by the LAN/MAN Standards Committee (LMSC). Generally, based on the different researching directions, the communication protocols are standardized by the individual working group. IEEE 802.15.4 is the most common standards for WSNs that defined by IEEE 802.15 working group，but there is not only IEEE 802.15.4 and also other protocols for supporting WSNs. On the other hand, another working group 802.11 is developed to coexist with wireless sensor network.

2.9.1 IEEE 802.11

IEEE 802.11 is the 11th working group of the 802-family. This standard specializes the Wireless Local Area Network (WLAN). Generally, IEEE 802.11 is also called Wi-Fi to instead the standard's name of WLAN, because Wi-Fi Alliance defines the Wi-Fi standard has the overlapping that compare with the IEEE 802.11 standards.

Nowadays, there exist several WLAN standards defined by the different task groups under the IEEE 802.11 working group.

802.11a

802.11a uses the same core protocol as the original standard of IEEE 802.11 legacy. It operates in the Industrial Scientific Medial (ISM) frequency band at 5 GHz with a maximum raw data rate of 54 Mbit/s 802.11a supports an indoor communication range up to 35 meters (Wikipedia 2008).

802.11b

802.11b has a maximum raw data rate of 11 Mbit/s and it uses the same medium access method defined in the 802.11 legacy. 802.11b is a direct extension of the modulation technique defined in the 802.11 legacy. It supports an indoor communication range up to 35 meters. The dramatic increase of throughput in 802.11b with simultaneous substantial price reductions led to the rapid acceptance of 802.11b as them wireless LAN technology (Wikipedia 2008).

802.11g

On June 2003, 802.11g was satisfied as the third communication standard. It uses the 2.4 GHz band but achieves a maximum raw data rate of 54 Mbit/s, or approximates 19 Mbit/s net throughputs. 802.11g hardware is fully backwards compatible with 802.11b hardware and the indoor communication range is the same 35 meters as 802.11b. Currently, IEEE 802.11 b/g are the most common WLAN standards in the commercial applications (Wikipedia 2008).

802.11i

802.11i is an amendment to the IEEE 802.11 standard specifying security mechanisms for wireless networks. It supersedes the previous security specification, Wired Equivalent Privacy (WEP), which was shown to have severe security weaknesses. Wi-Fi Protected Access (WPA) had previously been introduced by the Wi-Fi Alliance as an intermediate solution to WEP insecurities. WPA implemented a subset of 802.11i. The Wi-Fi Alliance refers to their approved, interoperable implementation of the full 802.11i as WPA2. 802.11i uses the Advanced Encryption Standard (AES) block cipher, whereas WEP and WPA use the RC4 stream cipher (Wikipedia 2008).

802.11-2007

802.11-2007 was defined by the task group “ma” and combined with many of the amendments of the 802.11 standards. It was created as a single document which includes 8 amendments (802.11 a, b, d, e, g, h, i, j) as the based standard. 802.11-2007 is also called 802.11ma or REVma (Wikipedia 2008).

802.11n

802.11n is a coming standard. It is a proposed amendment which differs from the previous 802.11 standards by adding Multiple-Input Multiple-Output (MIMO). The operating frequency band is 5GHz and/or 2.4GHz. The maximum data rate is 248 Mbit/s and the indoor communication range is about 70 meters (Wikipedia 2008).

2.9.2 IEEE 802.15

IEEE 802.15 is the 15th working group under the IEEE 802. This working group is specializing to Wireless Personal Area Network (WPAN). 802.15 working group is further divided to following six task groups (IEEE 802.15 2008):

Task Group 1

IEEE 802.15.1 was published on 14th of June, 2002. This task group is an additional resource for the existing Bluetooth devices. It defines the lower transport layers for Bluetooth wireless technology.

Task Group 2

IEEE 802.15.2 develops Recommended Practices to facilitate coexistence of Wireless Personal Area Networks (802.15) and Wireless Local Area Networks (802.11).

Task Group 3

IEEE 802.15.3 is chartered to draft and publish a new standard for high-rate (20Mbit/s or greater) wireless personal networks. It is separated into 3 subtasks:

a) WPAN High Rate Alternative physical layer

b) MAC Amendment

c) WPAN Millimeter Wave Alternative physical layer

Task Group 4

Task group 4 is chartered to investigate a low data rate solution which would enable from multi-month to multi-year battery life and a very low complexity. It is operating in an unlicensed, international frequency band which includes 16 channels in the 2.4GHz ISM band, 10 channels in the 915MHz and one channel in the 868MHz band. Potential applications are sensor networks, interactive toys, smart badges, remote controllers, and home automation. Currently, it is the most common standard for the wireless sensor networks. There are also five subtasks included into this group. They are:

a) Wireless Personal Area Networks Low Rate Alternative Physical layer

b) Revisions and Enhancements

c) Studying the recent Chinese regulatory changes which have opened the 314-316 MHz, 430-434 MHz, and 779-787 MHz bands for WPAN use within China.

d) Define an amendment to the existing standard 802.15.4-2006.

e) Define a MAC amendment to the existing standard 802.15.4-2006.

Task Group 5

In this group, it is chartered to determine the necessary mechanisms that must be present in the physical and MAC layers of WPAN to enable mesh networking which means a network providing a high-bandwidth communication over a specific coverage area.

Task Group 6

This group will define a standard for short range, wireless communication in the vicinity of, or inside a human body. Such a network is called the Wireless Body Area Network (WBAN).

2.9.3 IEEE 802.15.4 and ZigBee

As a communication standard for WSNs, IEEE 802.15.4 only provides the service in Data Link Layer and Physical Layer. ZigBee is one option to organize the higher layer services on top of IEEE 802.15.4. The structure base on the Open System Interconnection (OSI) model is shown in Figure 2.4:

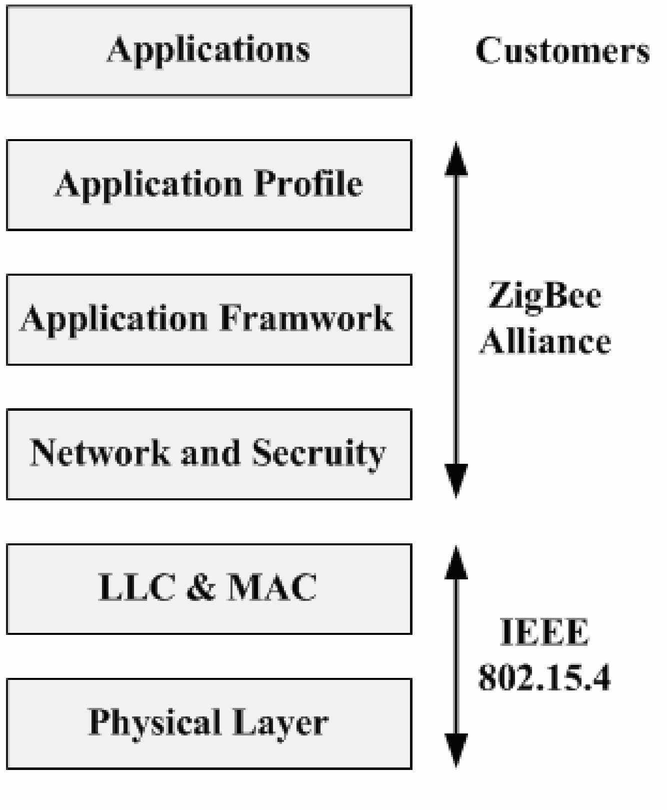

Figure 2.4. The reference model of IEEE 802.15.4 and ZigBee

This picture shows how ZigBee is build on top of IEEE 802.15.4 communication protocol.

802.15.4

IEEE 802.15.4 is the standard specified by task group 4 of 802.15 working group. It is target to low data rate wireless personal area networks. The standard specified Radio Frequency (RF), MAC and physical layer. In IEEE standard, the RF specification includes modulation type, coding, spreading, symbol/bit rate, and canalizations. In physical layer, the standard mainly to manage the radio channel and control the packet data flow. Also the medium access control sub-layer contains over two dozen primitives that allow data transfer, both inbound and outbound, as well as management by higher-level entities of the RF and physical layer.

According to Callaway (2004: 247), for maximizing the utility of the standard, the 802.15.4 task group had to balance the requirement to achieve the small size, low-cost and low-power consumption sensor nodes based on the desired standard that is required by a wide variety of market applications. The functionalities of the resulting standard can be divided to two physically different device types:

- **Full Function Device** (FFD), which can operate in any of the network roles (PAN coordinator or device). It must have sufficient memory to store routing information as required by the algorithm employed by the network.

- **Reduced Function Device** (RFD), which is a very low cost device, with minimal memory requirements.

ZigBee

As it is shown in Figure 2.4, IEEE 802.15.4 does not standardize the communication protocols above the Data Link Layer. Therefore, based on the IEEE 802.15.4 standard, ZigBee is defined by the ZigBee Alliance to carry out the operating of the higher communication protocols (ZigBee Alliance 2008).

ZigBee uses IEEE 802.15.4 beacon technique in which a node continuously transmits small packages alerting the neighbors with its presence in the network and works on routing discovery protocol (Benkic & Planinsic 2007). When a node needs to communicate with another node, it sends a route request packet in broadcast communication type. Generally, there exist four basic topologies in ZigBee network: Peer to Peer, Star, Mash and Cluster Tree. The architectures of these four topology types are shown in Figure 2.5.

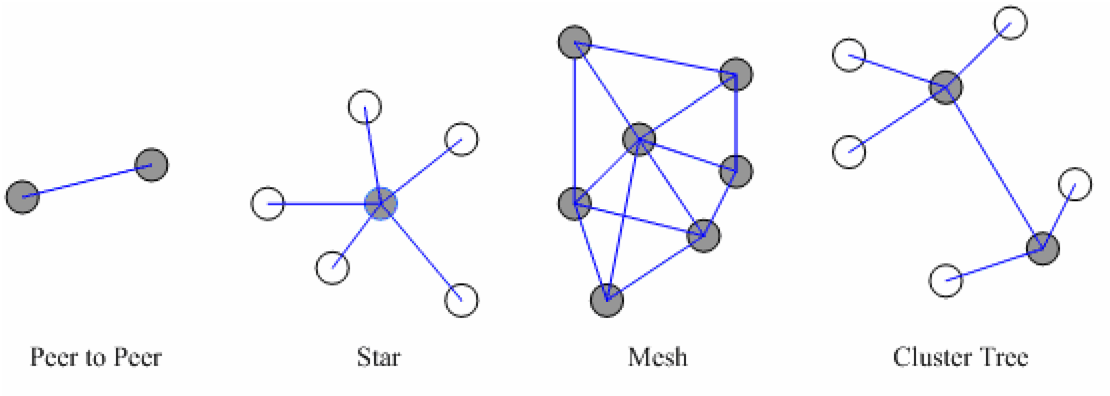

Figure 2.5 ZigBee topology types.

A node in a ZigBee network can act in three different modes:

a) ZigBee Coordinator(ZC)

b) ZigBee Router (ZR)

c) ZigBee End Device (ZED)

2.10 MAC Protocols

In OSI model, the Medium Access Control (MAC) protocol belongs to the Data Link Layer, the second layer of OSI model. MAC protocol is responsible for the channel access in wireless communication and it is directly on the top of the Physical Layer.

2.10.1 Common MAC Protocols

Generally, the medium access control protocol can be divided into three common methods: Fixed assignment, Demand assignment and Contention access.

Fixed assignment protocol

Typically, the protocols for fixed assignment can be divided into Frequency-Division Multiple Access (FDMA), Time-Division Multiple Access (TDMA), and Code-Division Multiple Access (CDMA).

FDMA uses radio systems to share the radio spectrum. In this case, the available radio spectrum is divided into many orthogonal sub-channels. For reduce the interference between each device, the overlapping can be minimized by large separation

TDMA uses digital transmission technology that allows several wireless devices to access a single radio-frequency channel without interference. It means each transmitter send the data in a very short time then stop for a while to give the opportunity to others to send.

CDMA utilizes the Spread Spectrum (SS) techniques which spreads the original signal in frequency domain using the independent signal to the information signal. Generally, CDMA is the most efficient multiple access method from many point of views.

Demand assignment protocol

The main idea of demand assignment protocol is to optimize the utilized of the channel to each wireless device. The typical protocol of demand assignment can be divided into three kinds of methods which are polling, reservation and turnking.

In the polling method, a master device is applied to query the each slave device whether it has data to transmit. If the queried device has the data to transmit then the master device will allocate the channel to the ready device and it is allowed to use the full data rate of the channel to transmit.

The second method is reservation. It means before transmitting, the slave devices reserve the transmitted time from the master device. After the master device receives the request, the slave device will start to transmit data based on the allowed transmission order.

Turnking method is a multiple-access scheme that dynamically assigns communication requests to available logical channels. Any fixed-assignment MAC protocol (FDMA, TDMA, and CDMA) can be employed, with the goal to substantially improve the channel efficiency without causing the degraded quality of services (QoS) to any user.

Contention access protocol

In this type of protocol, the master device will not operate any control to assign the channel access to the slave devices. If the slave devices have not get the access then it can try again later. The typical protocols of contention access include: ALOHA, CSMA, CSMA/CD and CSMA/CA.

ALOHA is a simple random assignment protocol developed for packet radio communication among uncoordinated contending users. Slave devices start to send the packets until they get the traffic to transmit. In case of collision, the devices will back off for random time and try again.

Carrier Sense Multiple Access (CSMA) improves upon the relatively poor channel capacity of ALOHA. It means when the transmitter ready to send the data, it first to listen the carrier frequency; if the channel is occupied then transmitter will wait and start to send the data until the end of last transmission.

To solve the disadvantage of CSMA methods and efficiently reduce the collision, Carrier Sense Multiple Access with Collision Detect (CSMA/CD) is applied to extend the capabilities of a wireless device. If the wireless device detects the collision, it will stop the transmission immediately and wait for a random duration of time then try to transmit again.

Base on the method of CSMA, the Carrier Sense Multiple Access with Collision Avoid (CSMA/CA) utilize the RTS (Request To Send) & CTS (Clear To Send) handshake to check the condition of the communication between the source and destination. It efficiently avoids the hidden and exposed device problems.

2.10.2 MAC Protocols for WSN

Because of the critical requirement of the wireless sensor network, the random access-based protocol and schedule-based protocol are applied to support the WSN, especially for control the energy consumption of the wireless sensor nodes.

Random access-based protocol

In some cases, the random access-based also called the contention-based; in this types of protocol, there is no coordination controlling between the master device and the sensor nodes for accessing the channel. The original ALOHA, CSMA, CSMA/CD and CSMA/CA are still employed to solve the hidden and exposed device problems, but there some additional protocols are proposed to provide the extra MAC protocol service for WSN.

The Sparse Topology and Energy Management (STEM) protocol, it is an efficient energy management method. It uses two different radio channels: the wake up and data radio channels. STEM is one kind pseudo asynchronous scheduled scheme. It means, the sensor node turn off the data radio channel and wake up when it is required. If the node has the data to transmit, it will first use the wake up channel. When all of its neighbor nodes received the wake up signal, they will keep awake state for enough long period to receive all the transmitted data. After receiving all the packets, the node will change to sleep mode again.

The Power Aware Multi-Access with Signaling (PAMAS) is originally designed for ad hoc networks (Karl & Savvides 2005: 131). It uses the separate signaling channel to avoid the overhearing between the neighboring sensor nodes and ignore the idle listening problem. It allows the node to use RTS and CTS transmission and still keep the nodes' transceivers in power off mode. It can efficiently reduce the energy consumption.

Schedule-based protocol

Generally, the schedule-based protocols can be divided into Self-organizing medium-access control for sensor networks (SMACS) and Low-Energy Adaptive Clustering Hierarchy (LEACH).

SMACS is a distributed protocol that incorporates features of FDMA, TDMA, and CDMA (Edgar & Callaway 2007: 239). It is achieved by randomly assigning a channel, selected from a large number of channels (FDMA), or spreading code (CDMA) to each link. Using the super frame structure, each node maintains its own time slot schedules with all its neighbors, and nodes are required to tune their radios to the proper frequency channel or CDMA code to achieve communication (Sohraby 2007: 144).

LEACH takes a hierarchical approach and organizes nodes into clusters (Sohraby 2007: 144). It uses TDMA to achieve communication between nodes and their cluster head. The master node computes a schedule bases on TDMA and sends it to all cluster node. It can avoid collisions among data messages. Base on this method, the node can determine its active time period. It allows the cluster nodes, without the master node, can turn off their transceiver until it is assigned the time slot.

2.11 Wireless Sensor Nodes

Wireless Sensor Networks are built by small devices called sensor nodes. Basically, the senor nodes are equipped with 5 components; there are a microprocessor, a radio transceiver, an external memory, the power supply and the sensors. In a typical sensor network, the wireless sensor nodes perform the computations, sensing and communications.

2.11.1 Spec Node

Spec is a typical miniaturized sensor node. It is produced in 2003.The required size was designed as 2.5mm x 2.5mm of silicon. It equips a 4-8MHz CPU, a 3K RAM memory and the data rate is at 50-100Kbps. The peak power is 3mW and about 3uW in idle mode. It uses a 50 kbps radio bandwidth. As an example application, it could trigger an alarm when an asset leaves a facility without authorization (Hill & Horton 2004).

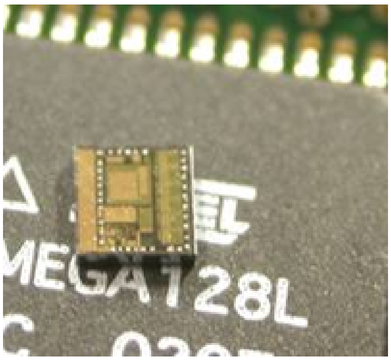

Figure 2.4 The Spec Node (Howstuffworks 2008).

2.11.2 Rene Node

Rene node was developed in 1999 (Hill & Horton 2004). It has an ATMEL8535 CPU and two 51-pin connectors for interacting to sensor and programming boards. It uses a 916MHz radio transceiver with bandwidth about 10Kbps, a 512B RAM and an external flash memory in 8KB. It supports the TinyOS operating system. An additional antenna provides the communication range up to 30 meters. The peak power is 60mW and consume about 0.036mW when the node in sleep mode.

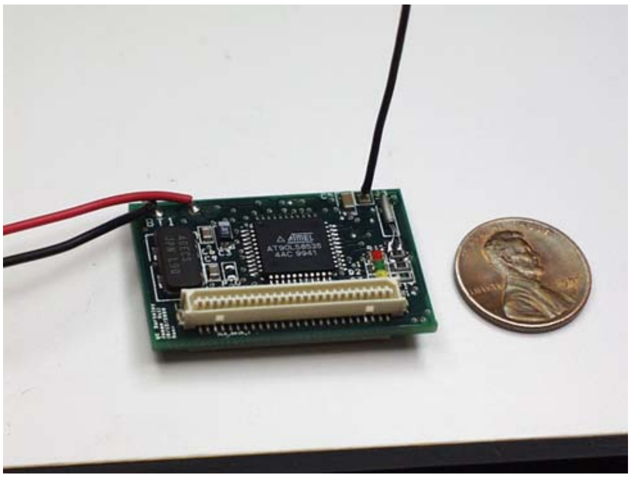

Figure 2.5 The Rene Node (UCLA 2008).

2.11.3 Mica Mote

Mica mote was developed based on the Rene node. It has dimensions of 1.25× 2.25 inches and the applied operating system is TinyOS. It can be used as a sensor platform in self-configuring multi-hop wireless networks (Hill & Culler 2002). The mote is equipped with TAtmel ATMEGA 128L 8-pin CPU and running at 4MHz. The bandwidth of the mote is capable to reach 40 Kbps. Mica includes 128KB of program memory and 4KB of data memory. The power source is a pair of AA batteries.

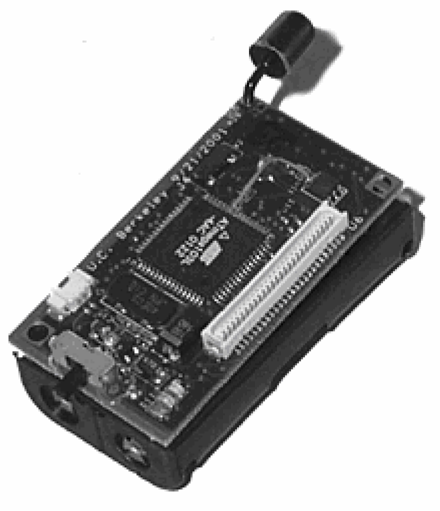

Figure 2.5 The Mica Mote (Hill & Culler 2002).

2.11.4 Mica-2 Mote

Mica-2 mote was developed in 2001. It has an ATMEGA 128 CPU, 868/916 MHz Multi-Channel Radio Transceiver and more than one year battery life on a pair of AA batteries (Crossbow 2008a). Mica-2 also equips a 4KB program memory, 128KB external flash memory and supports the TinyOS operating system (Hill & Horton 2004).

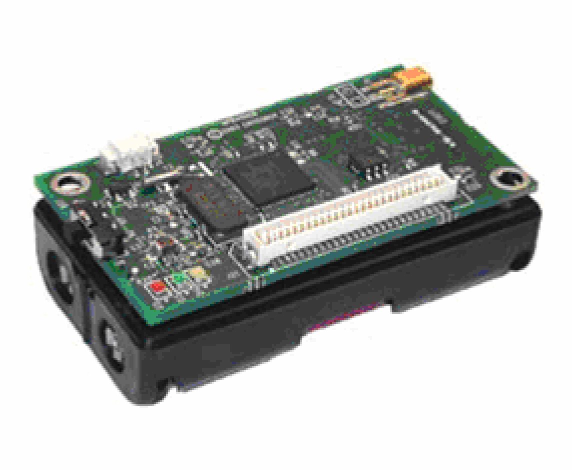

Figure 2.6 The Mica-2 Mote (Crossbow 2008a).

2.11.5 Mica-Z Mote

Mica-Z mote was developed in 2004. It uses the same CPU as Mica-2 and has an 802.15.4/ZigBee compliant RF transceiver. The data rate is in 250 Kbps. It supports TinyOS, SOS and MantisOS operating systems. Crossbow offers a variety of sensor and data acquisition boards for the MICAz Mote (Crossbow 2008e). All of these boards connect to the MICAz via the standard 51-pin expansion connector.

Figure 2.7 The Mica-Z mote (Crossbow 2008b).

2.11.6 TelosB

TelosB supports the open source operating systems; it has an IEEE 802.15.4 compliant radio transceiver that operates in 2.4GHz band and it reaches a high data rate of 250Kbps; The CPU is 8MHz TI MSP430 Microcontroller. The node is also equipped with USB interface to provide the data collection and programming (Crossbow 2008c).

Figure 2.8 The TelosB (Crossbow 2008c).

2.11.7 T-mote Sky

The T-mote Sky, produced by MoteIV, improves the features of the TelosB Mote: it has a TI MSP430F1611 microcontroller, 10 KB of SRAM memory, 48 KB of program memory, and 1024 KB of external flash memory (University of New South Wales 2008). It uses the same bandwidth and radio as the TelosB.

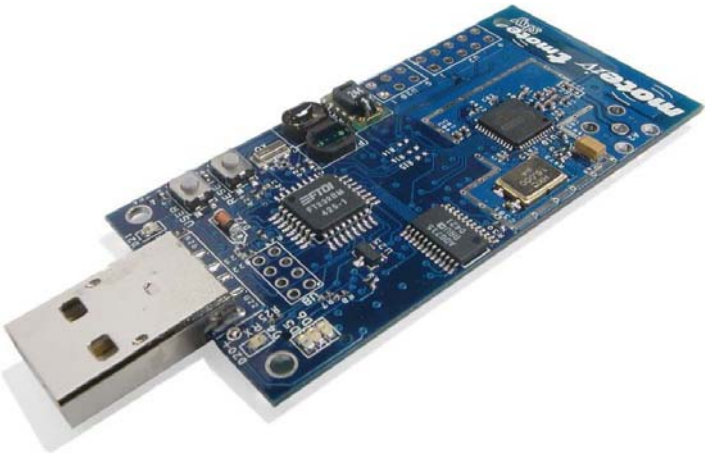

Figure 2.9 The T-mote Sky Node (University of New South Wales 2008).

2.11.8 XYZ

XYZ is a new versatile general-purpose wireless sensor node; it uses an OKI ARM ML67Q500x microprocessor and a CC2420 ZigBee compliant radio from Chipcon which provide 256Kbps high data rate; it also has 32KB of RAM and 256KB of external flash memory (Lymberopoulos & Savvides 2008). The node uses SOS operating system (Lymberopoulos & Savvides 2008).

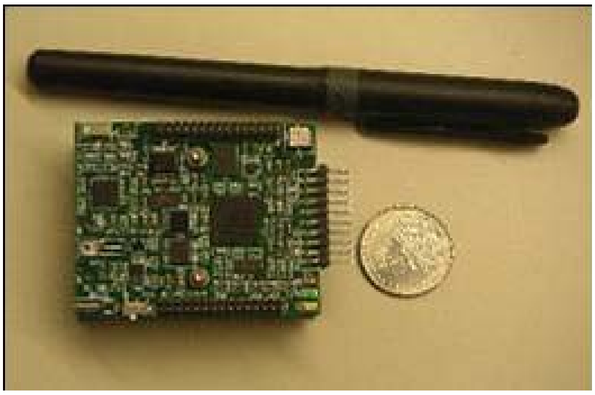

Figure 2.10 The XYZ Node (Lymberopoulos & Savvides 2008).

2.11.9 BT Node

BT Node is equipped with Atmel ATmega 128L that operates at 8MHz frequency, a Chipcon CC1000 radio and also Bluetooth. The node uses 64KB SRAM, 128 external flash memory and 4KB EEPROM. t support the TinyOS operating system and can also be programmed by standard C.

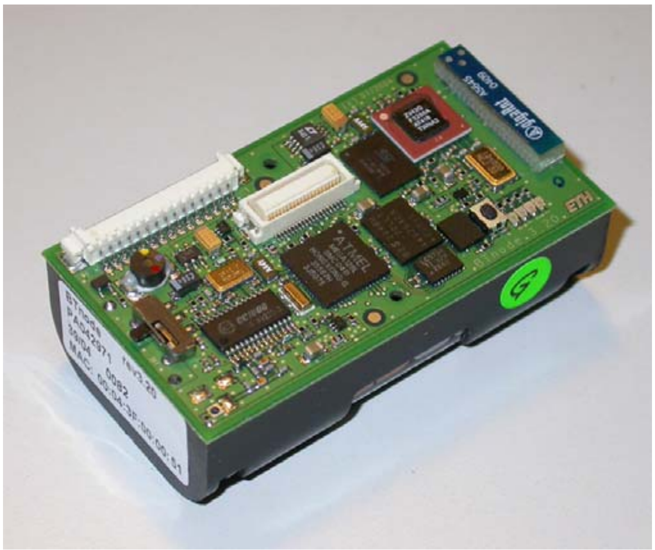

Figure 2.11 The BT Node (BTnode 2008).

2.11.10 Intel Imote

Intel Imote, developed by Intel Research, is designed to be a high-bandwidth sensor platform (Hill & Horton 2004). It has an ARM core CPU that operates at 12MHz. The mote uses Bluetooth radio support the Bluetooth within a range of 30 meters. It uses 64KB SRAM and 512KB external flash memory. Imote 1.0 is the higher version of Imote. It is equipped with ARM 7TDMI that operates at 12-48 MHz and provides Bluetooth 1.1. It uses Bluetooth for the radio provides increased link reliability and security

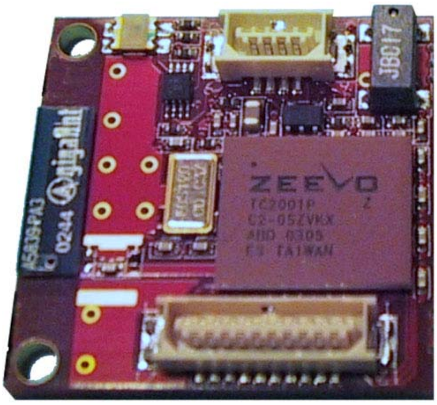

Figure 2.12 The Imote Node (BTnode 2008).

2.11.11 Stargate

The Stargate is a representative gateway node developed on 2003. Its dimensions are 9.53cm x 6.33cm x 1.86cm and it is equipped with Intel PXA255 CPU, which operates in 400MHz, a 51-pin SDC400CA Daughter Card, a 64MB SDRAM and a 32MB flash memory (Crossbow 2008d). The daughter card provides the extra interfaces, including RS-232 serial, 10/100 Ethernet, USB host and Joint Test Action Group (JTAG). The Stargate node supports open source operating system which is an embedded version of the Linux. In addition to providing a range of system capabilities, Linux provides a suite of device drivers for enabling gateway nodes to bridge to legacy networks (Hill & Horton 2004).

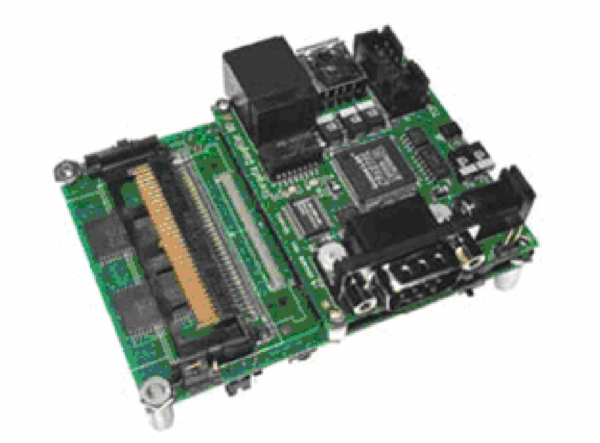

Figure 2.13 The Stargate Node (Crossbow 2008d).

2.11.12 Intrinsyc Cerfcube

Intrinsyc Cerfcube node is a high-performance, low-power design that is the perfect starting point for developing internet devices; it is equipped with the same Intel processor as the Stargate node and with the Intel StrataFlash and fast SDRAM. The node also supports extra interfaces, including the Ethernet, one serial port, JTAG and flexible digital I/O. It also can provide the additional Bluetooth support, wireless LAN/WAN support, or up to 6GB of local storage with an IBM Microdrive (ARM 2008).

Figure 2.14 The Intrinsyc Cerfcube Node (ARM 2008).

2.11.13 Future Development

The key target of the future development is how to efficiently increase the performance of the sensor nodes under the limited resources. There several tasks should be considered to provide the future development. For examples, the processing power of the CUP should be efficiently increased to support the digital signal processing and localization applications; the reliability of the radio propagation should be improved to decrease the packet loss rate; the dimensions of each node should be designed as small as possible for suitable more applications; finally, the cost also should be decrease for large deployment of nodes in required environments and for the widely usage of the WSNs.

2.12 Security

Generally, there are four main tasks to consider in WSN security: key management, encryption and decryption, secure routing and prevention of denial of service.

2.12.1 Key Management

Generally, key management in sensor networks can be divided into 4 different categories (Ren 2006):

- **Individual Key**. Each node has its unique key that it shares pair wise with the base station. This key is applied for secure communication between the sensor node and the gateway node.
- **Cluster Key.** A key is shared between a node and its neighboring nodes, which belong to the same cluster. The cluster key is applied to secure securing locally

broadcasted messages.

- **Pairwise Shared Key**. Each node shares a pairwise key with each of its nearby neighboring nodes. This key is applied to secure such communication, which requires privacy or source authentication.
- **Group Key.** A globally shared key is used by the gateway node to encrypt messages that are broadcasted to the whole group.

On the power consumption point of view, to preserve the energy efficiency, the encryption and decryption processes are usually kept simple

2.12.2 Secure Routing

The main attack type is called external attack in which the attacker tries to partition the network or increase the traffic load in the network by injecting wrong routing information or by replaying the old routing information. Such external attacks can be efficiently repulsed by using encryption.

2.12.3 Network Malfunctioning

Usually network malfunctioning is caused by node failures, software errors, critical environments or by the resource exhaustion, etc. As the basic performances of the sensor node, the network automatic initialization and self-healing can support the WSNs automatic recovering from network malfunctioning.

2.13 Data Processing

The two main steps of data processing are the data selecting and combing. Normally, the nodes should combine the original data, erase the useless data and select the valuable data to send. This process can keep the reliability of the transmitted data. Typically the data processing can be further divided into data aggregation, data storage and data fusion.

2.13.1 Data Compression and Aggregation

Usually, the end user wants to get the sensing or measuring data from the WSNs. These data are gathered by the sensor nodes and reported to a sink node. The sink node requests readings from the entire sensor network with appropriate request messages. The data aggregation protocol has to combine end-to-end characteristics (e.g. how many sensors are there, how important is a given piece of information, does it come from a covered, etc.) with local decisions (e.g. how to treat the lost packets, who to use the energy to avoid packet losses, etc) (Karl & Lobbers 2003). It efficiently optimizes the energy requirement to obtain the data from the source node.

According to Karl & Lobbers (2003), the mainly process of data aggregation starts at a leaf node which plan to transmit a single reading to its parent node. This transmission should be combined whit three values:

1) The measuring value of the sensor readings.

2) The number of sensors that have contributed to the measuring value.

3) The approximated area that is covered by this reading.

These recursion steps are performed in an intermediate node. Base on the collected value of the relevant readings coming from the child nodes, the intermediate node calculates these new values:

- The aggregation function based the received values and its own sensor reading.
- Sum up the numbers of contributing sensors.
- A new approximation of the covered area.

After these three values have been computed, the whole packet is passed on to the link layer, where a redundancy control module decides how to transmit the packet.

The historical data collected by individual nodes must be stored in some places for the future query processing. External storage, local storage and data-centric storage are the mainly three types of data storage (Chen & Hou 2005: 450). The external storage means the sensing or measuring data will be relayed to some external storage which locates outside the network, such as a gateway node. It not requires the nodes to consume the extra energy to store the historical data. The local storage means the data are stored at the local storage of each sensor node. It not requires the initial communication cost. However, each sensor needs to process all queries and a query is flooded to all sensors. The data-centric storage means the data will be routed to a predefined location. First the sensor maps an event it detects to a suitable location via the geographic hash function and then routes the data to a node according to the predefined location.

2.13.2 Data Fusion

Generally, data will be gathered by various sources and transmitted through multiple hops to a sink node. Based on the limited power resources, data fusion is applied to solve this problem. Instead of transmitting all the data to a centralized node for processing, data are processed locally and a concise and useful data is transmitted to sink node. Data fusion optimizes the usage in bandwidth and energy. Its benefits become manifest, especially in a large-scale network.

3 PiccSIM Simulation Platform

As the main target of this Master’s Thesis, the results of the mobility control simulation will be obtained from a virtual WSNs. The applied simulation system is modified based on the PiccSIM simulation platform. This platform is developed for modeling, design, simulation and implementation of networked control systems. It integrates the mobility control processing based on the simulation tools available in MATLAB with the NS2 (Nethi & Pohjola 2007).

Three different simulation scenarios for wireless automation applications were developed by using the PiccSIM simulation platform. This platform combines both the control system and networking simulator. The processing runs in real-time with control loops over simulated networks, which makes it possible to research different networking protocols and congestion scenarios in a realistic Network Control System (NCS) setting (Pohjola & Nethi 2007).

3.1 Target Tracking

Target tracking is one scenario that is related to the mobility control. As the definition, the static wireless sensor nodes are scattered in a grid distribution. The distances between each node are 200 meters and the distance measurement range is defined as 300 meters. If the mobile node closes the static node less then 300 meters, the distance between them will be measured and transmitted to the computation center for estimating the position of the mobile node.

Generally, in the realistic environment such as a factory, the sensor nodes may not be deployed in a grid distribution. For this reason, we modify the scenario of target tracking from TKK. As an evolvement, the simulated environment of the WSN is complicated. The static wireless sensor nodes will be randomly deployed in the decided area. It means the positions of these static nodes are not manually defined. An efficient localization method will provide the randomly deployed static nodes to localize themselves. Based on these changes, the simulated WSN will be more close to the realistic situation. The obtained results are more valuable to support the future development.

3.2 Structures

Basically, the PiccSIM simulation platform consists of two computers (xPC Target and NS2) which are connected to each other by using the local area network. Figure 3.1 shows the connective structure between the xPC Target and the network simulator.

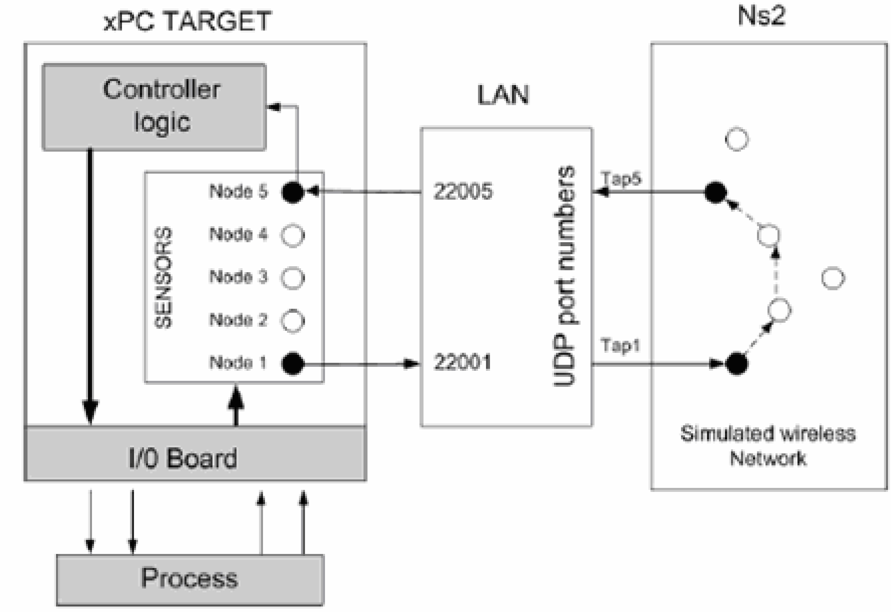

Figure 3.1 Connective structure of the PiccSIM platform (Nethi & Pohjola 2007).

xPC Target: xPC target runs a real-time operating system, the main responsibility of this computer is to measure and control the processes based on the user specified algorithms, made with a Simulink model in MATLAB, then the xPC transmits signals (i.e. UDP packets) to the network simulator (Pohjola & Nethi 2007).

Network simulator: A network simulation runs on a Linux system, on which NS2.30 is installed. By using the Network Simulator Emulator (NSE), the computer for network simulation, captures the User Datagram Protocol (UDP) packets and injects them into the simulated (wireless) network model. These entities are in the simulator. Typically, this network simulator is focused on simulating the wireless communication network. For example, the packet loss and the routing protocol in WSN.

4 Simulations

4.1 Simulation Setup

By using the PiccSIM simulation platform, the scenarios will be simulated in a virtual WSN. The following steps show the whole process of the simulations. They also show how the PiccSIM simulation platform is modified and how to achieve the realistic surrounding.

4.1.1 Deploying the Sensor Nodes

The first step is to randomly deploy the static sensor nodes in the decided area. As the assumption, 25 static wireless sensor nodes will be randomly deployed in a 500 x 500 area; the communication range of each sensor node is 200 meters and the distance measurement range is assumed as the same as the communication range. All sensor nodes have radio module for communication and are equipped with an ultrasound transceiver for the distance measurement (Nethi & Pohjola 2007). It is also assumed that each node has at least three one-hop neighbors. This is done to get unique realization for the network scenario. Figure 4.1 is an example that 25 wireless sensor nodes are randomly deployed in the decided area and the blue circles show the ground-truth positions of these 25 static nodes.

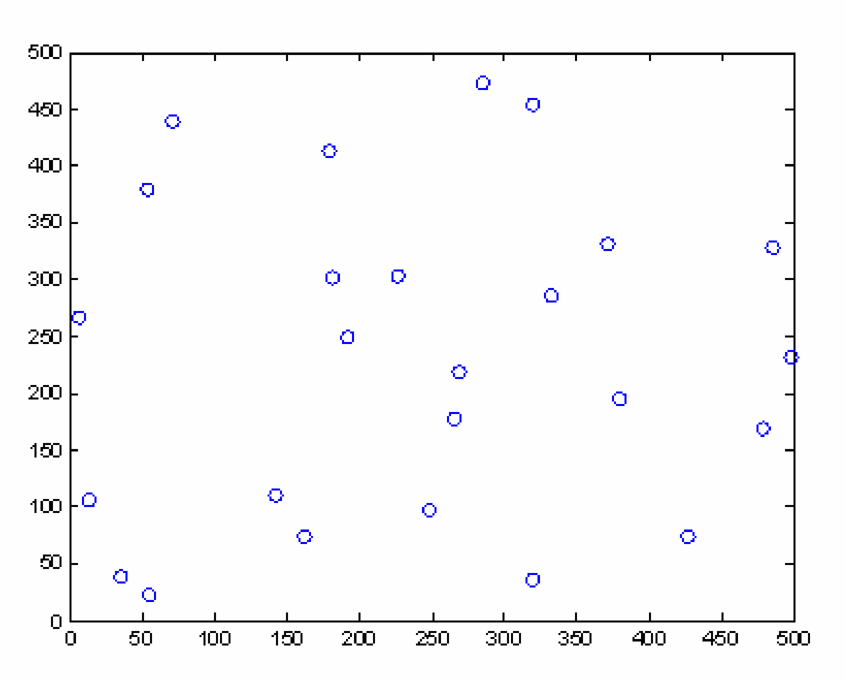

Figure 4.1. Randomly deployed 25 nodes with 200 communication range in the decided area.

4.1.2 Static Node Localization

After the 25 static sensor nodes are randomly deployed, a centralized algorithm is executed to localize the static nodes. In an outdoor environment it would be too expensive to equip every node with GPS. For an indoor environment GPS can not be used at all. Multidimensional Scaling (MDS) is one kind of localization method to solve a distance-based localization problem. It is a data analysis technique that transforms proximity information into a geometric embedding and an efficient way for nodes localization in wireless communication networks, where the task is to use the distance information between nodes to determine the coordinates of nodes in 2D or 3D space (Shang & Ruml 2003a).

Generally, MDS can be classified according to whether the similarity data is qualitative (non-metric MDS) or quantitative (metric MDS) (Shang & Ruml 2003b). Classical MDS is one variation of the MDS method and also the simplest case: the data is quantitative and the proximities of objects are treated as distances in a Euclidean space and if the proximity data ware measured without error in a Euclidean space, then Classical MDS would exactly recreate the configuration of points (Shang & Ruml 2003b).

The Classical MDS provide the static sensor nodes to localize their positions by themselves. The Classical MDS method can be separated into two main procedures: the MDS computation and the coordinate transformation. Also, during the simulation process, the distance measurement range is assumed as 200 meters, and it is equal to the communication range.

Computing Relative Localization

Normally, there are three different input values required to support the Floyd-Warshall method. The first parameter is the number of static sensor nodes. The second one is a connective matrix which means if the distance between a pair of nodes is in the distance measurement range, then the value in the matrix is 1 otherwise the value is 0. The last one is the distance matrix, which means if the distance between a pair of nodes are in the distance measurement range then the distance will be stored in the matrix otherwise the value will be infinite. For the pair-wise distances that exceed the distance measurement range, it can be calculated by using the Floyd-Warshall algorithm.

After the shortest Euclidean path is computed by using the Floyd-Warshall algorithm, a new distance matrix with the measured distance will be made. Then it can be applied to provide the MDS method:

$$B = -\frac{1}{2} \cdot \left(I - \frac{1}{n} U \right) \cdot D^2 \cdot \left(I - \frac{1}{n} U \right) \tag{4.1}$$

Where

D new distance matrix with the measure distance

B a symmetric square matrix

n number of sensor nodes

U n x n distance matrix

I n x n identity matrix

The ortogonalization of a square symmetric matrix B is a special case of SVD. Thus, the coordinate matrix can be computed by following formulas:

$$B = USV^T = USU^T = XX^T \tag{4.2}$$

$$X = US^{\frac{1}{2}} \tag{4.3}$$

Finally, a matrix X is calculated, and the first 2 columns in this matrix are the x and y coordinates for the computed positions of all sensor nodes via the Classical MDS method. Figure 4.2 shows the relative locations provided by MDS.

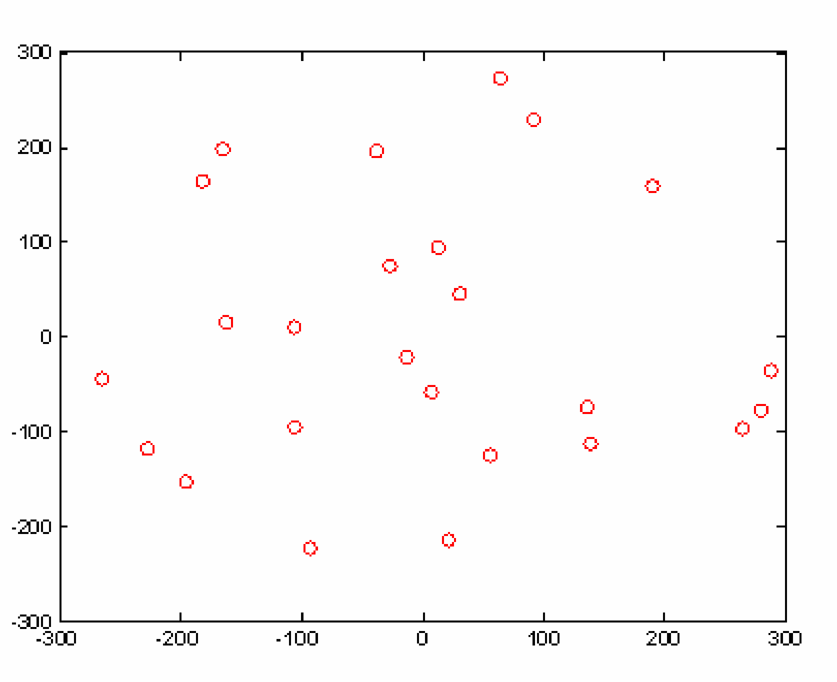

Figure 4.2. Relative locations of sensor nodes.

Transform to Reference Coordinates

Obviously, the coordinates in Figure 4.2 are different than the absolute coordinates. Hence, for transforming the relative coordinates to the original absolute coordinates, three beacons will be chosen to provide the transformation method. Figure 4.3 shows the ground-truth locations compared with the transformed relative locations and the three nodes with the green star are the three beacons.

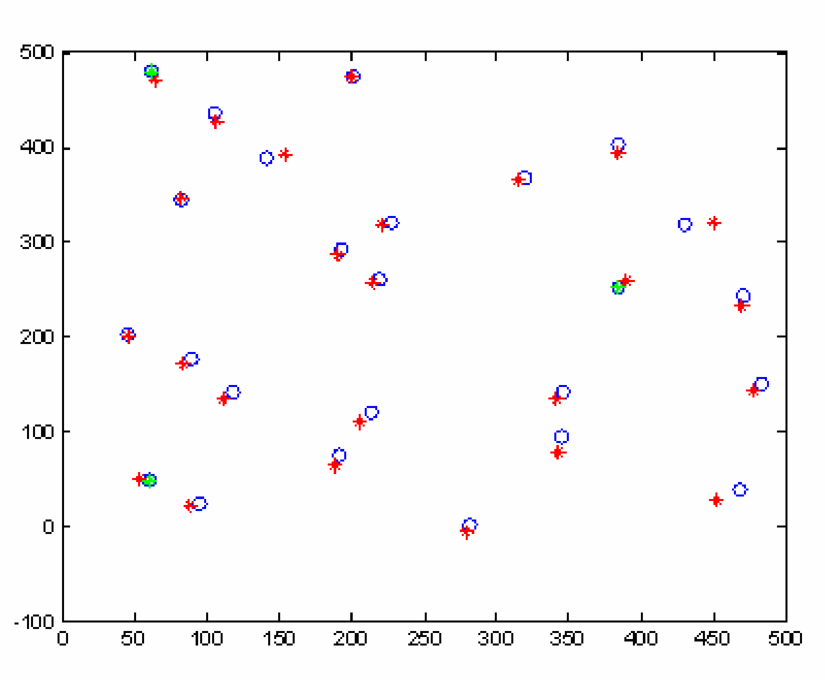

Figure 4.3. Ground-truth and transform relative locations

4.1.3 Mobile Node Trajectory

Once the positions of all sensor nodes are computed by using Classical MDS, the sensor nodes can localize their location based on the new coordinates which appear as red stars in Figure 4.3. After this step, a reference path will be created and the mobile node will be guided to move along this reference path.

As well as the initial definition, a reference trajectory will be randomly created. Three random points will be randomly chosen in the decided area. Based on these three points phenomena, the reference trajectory can be created for the mobile node. Figure 4.4 is an example of the randomly created path in the decided area.

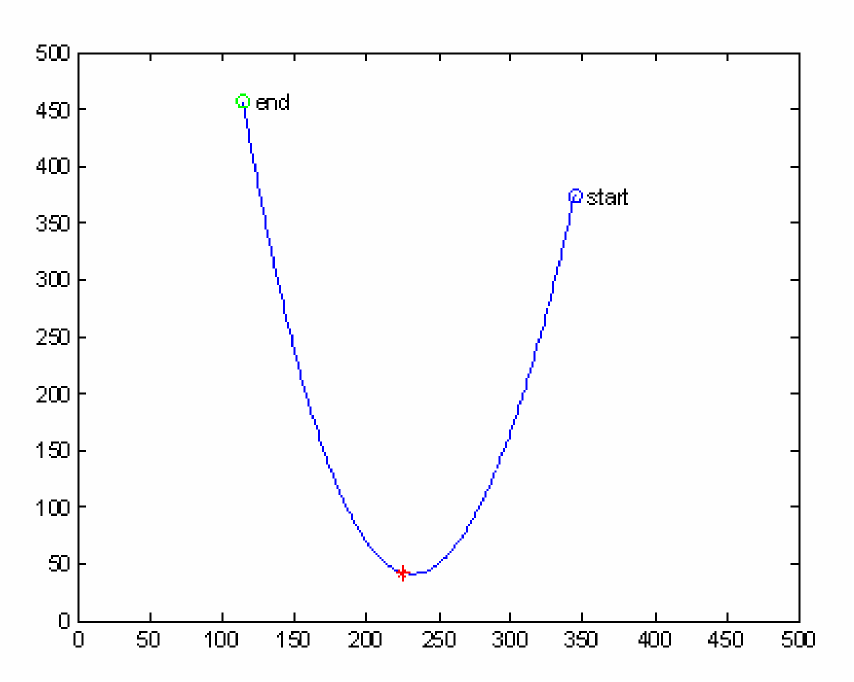

Figure 4.4. Reference path for the mobile node.

As can be seen in Figure 4.4, the blue circle is the starting point; the green circle is the end point and the red star is the middle point. In the simulation, the initial position of the mobile node is based on the first pair of x and y coordinates of the reference trajectory (x and y coordinates of the blue circle).

4.1.4 Communication Range

Wireless communication range is another important parameter. It mainly affects the network topology. Based on the different communication range level, it changes the quality of the simulation which means how well the mobile node moves along the pre-defined trajectory.

According to the scenarios, the wireless communication range is assumed as the same as the distance measurement range. In NS2, the radio propagation type is defined as

‘Two Ray Ground’ and the communication range is translated to the received power of each sensor node.

The main function for translating the communication range in meter to the receiving power can be divided in two parts. When the communication range is less than 9.28 meters, the formula is returned as:

$$\mathrm{Pr} = \frac{Pt \cdot Gt \cdot Gr \cdot \lambda^2}{(4\pi d)^2 L} \qquad (4.4)$$

if the communication range is larger than 9.28 meters, the formula is returned as:

$$\mathrm{Pr} = \frac{Pt \cdot Gt \cdot Gr \cdot hr^2 \cdot ht^2}{d^4 L} \qquad (4.5)$$

where the Pr is the receiving power, Pt = 0.28 W which is the transmitting power, Gt=Gr=1 which is the gain of the antenna, λ = 0.328 which means the wave length, hr=ht=1.5meters which means the height of the antenna, L =1 which means the parameter of the loss and d is the communication range in meter.

4.1.5 Information Sharing

Before starting the simulation, there some useful information need to send from the xPC Target to the NS2 for the wireless communication network simulation. Basically, they include:

1) **Coordinates:** It means the transformed relative locations of the static sensor nodes are computed by Classical MDS methods and also the initial position of the mobile nodes that is the first pair of x and y coordinates from the reference trajectory.
2) **Communication Range:** The different communication ranges are needed to update in the NS2 computer if it has been changed.
3) **Trajectory:** The reference path for the mobile node should be created before the simulation.

For transmitting the information to the NS2 computer, the ‘Samba’ server is installed in the Linux system. Thus, every time before the simulation starts, the above parameters will be automatically passed to the Linux system and provide the wireless network simulating in NS2.

4.1.6 Mobile Node Simulation

The main issues of this simulation is the target tracking which means the mobile node will be tracked and guided following the reference path by the command center. The key feature of this process is the target’s position estimation. It means the distances between the mobile nodes and the static sensor nodes will be measured, and then these measured distances will be sent to the command center. Based on the received parameters, the position of the mobile node can be computed.

Position Estimation

It is assumed that (u, v) is the position of the mobile node. If the distance between the static node and the mobile node are in the distance measurement range, then the static node will start to measure the distance with the target. The more distances have been measured by the static nodes, the more precisely the position of the mobile nodes will be estimated. In this case, we assume 12 nodes as the maximum number for supporting the distance measurement at one sample time.

The equations can be return as (Pohjola & Nethi 2007):

$$
\begin{aligned}
&(x_1 - u)^2 + (y_1 - v)^2 = d_1^2 \\
&(x_2 - u)^2 + (y_2 - v)^2 = d_2^2 \\
&\ldots \\
&(x_n - u)^2 + (y_n - v)^2 = d_n^2
\end{aligned}
$$

where x_i and y_i and (i=1, 2…n) are the x and y coordinates of the static sensor nodes. Combining the first two equations, we can get (Pohjola & Nethi 2007):

$$2 \cdot (x_2 - x_1) \cdot u + 2 \cdot (y_2 - y_1) \cdot v = x_2^2 - x_1^2 + y_2^2 - y_1^2 - d_2^2 + d_1^2 \tag{4.6}$$

and then the position of the mobile node can be computed by solving the linear equation:

$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \\ . & . \\ . & . \\ 2(x_n - x_1) & 2(y_n - y_1) \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 - d_2^2 + d_1^2 \\ x_3^2 - x_1^2 + y_3^2 - y_1^2 - d_3^2 + d_1^2 \\ . \\ . \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 - d_n^2 + d_1^2 \end{bmatrix} \tag{4.7}$$

The coordinates (u, v) can be computed by solving the equation (4.7) which is the position of the target. From a realistic environment point of view, the distance measurement will contain the noise, so the solution is obtained with the error. A standard Kalman Filter (KF) is applied to filter the noise. In some cases, when the computation center receives the measured distance less than 3, the KF can also be applied to solve this problem. The mobile node is modeled as a continuous-time 2D double integrator. The discrete-time model can be returned as (Pohjola & Nethi 2007):

$$x(k+1) = F \cdot x(k) + B \cdot u(K) + w(k) \tag{4.8}$$

$$y(k) = H \cdot x(k) + v(k) \tag{4.9}$$

used by the KF for the mobile node, is the discredited version of the continuous-time model with:

$$F = \begin{bmatrix} 2 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} h \cdot h/m & 0 \\ 0 & 0 \\ 0 & h \cdot h/m \\ 0 & 0 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

where h=0.2s is the sample time and m=1kg is the mass of the mobile node. The process and measurement noise terms w (k) and v (k) in the KF are Gaussian distributed with covariance Q = 10diag (4) and R = diag (2)/sqrt (n), respectively, where n is the number of received distance measurements. If there are less than 3 measured distances received by the computation center, only the update part of the Kalman filter is executed.

Control

After the computation center computes the location the mobile node, then it will guide the mobile node. Here, two Proportional Integral Derivative (PID) controllers are applied to control the mobile node; one for the x- acceleration and another is for the y-acceleration. The controllers are tuned by simulation to give a good reference tracking.

4.2 Scenarios

After completing the simulation setups, the different data and parameters should be obtained to provide the results analysis. The scenarios of the data collection are divided in to two directions. One is the analysis of the static sensor nodes and another is the mobile node.

4.2.1 Static Node Localization

This scenario is mainly consisting of two subtasks. It focuses on analyzing the quality of the static sensor nodes localization. Depending on different factors, the effect of the localization can be analyzed based on the collected data.

1) The first testing chooses 20 different examples of the randomly deployed static sensor nodes. The effects of the Classical MDS method causes by the different noise level will be analyzed. It means, in every example testing, the noise level will be increased from 0% to 55% in steps of 5%.The average error of the pairwise distances, the error standard deviation, the maximum and minimum value of the errors will be collected.

2) The objective of second testing is to analyze the effect of static nodes localization by selecting different beacons. Based on the same relative coordinate example, 10 different groups of beacons will be selected to support the coordinate transformation. In each case, the average error, error of standard deviation, the maximum and minimum value of the errors between the ground truth positions and the location estimates will be collected.

4.2.2 Mobile Node Tracking

There are three subtasks included in this scenario. By changing the factors of the scenario, the quality of the simulation will be proved. How well the simulation is running is based on the collected data from the following three subtasks.

1) In first task, there are 20 different randomly pre-defined reference trajectory will be applied to provide the simulating in the same static node setup. The times of the system succeeds to guide the mobile node from the beginning to the end will be recorded. The average number of participation static nodes is recorded to analyze the performance of this simulation system.

2) The target of second one is to show how well the mobile node follows the pre-defined trajectory. There are 10 different simulation setups which will be simulated. In every sample time, the distances between the mobile node and the reference point, the shortest distances between the mobile node and the reference trajectory and the summation of above two parameters will be recorded. Based on the collected data, it shows the quality of the mobility control by this simulation system.

3) In the last task, the communication range level for each static node will be increased from 40 meters to 200 meters in steps of 20 meters. All these communication range levels will be applied in the same simulation setup. The recorded data will show the performance of the simulation in different communication range level and what the limited communication range level for the system succeeds to guide the mobile node.

4.3 Results

After completing the testing scenarios in Section 4.2, the target data is recorded. As the results of the simulations, all the recordings will be analyzed in detail. It shows the performances of whole simulations.

4.3.1 Noise in Static Node Localization

By increasing the noise level in 20 different examples, the Figure 4.1 is plotted based on the obtained data.

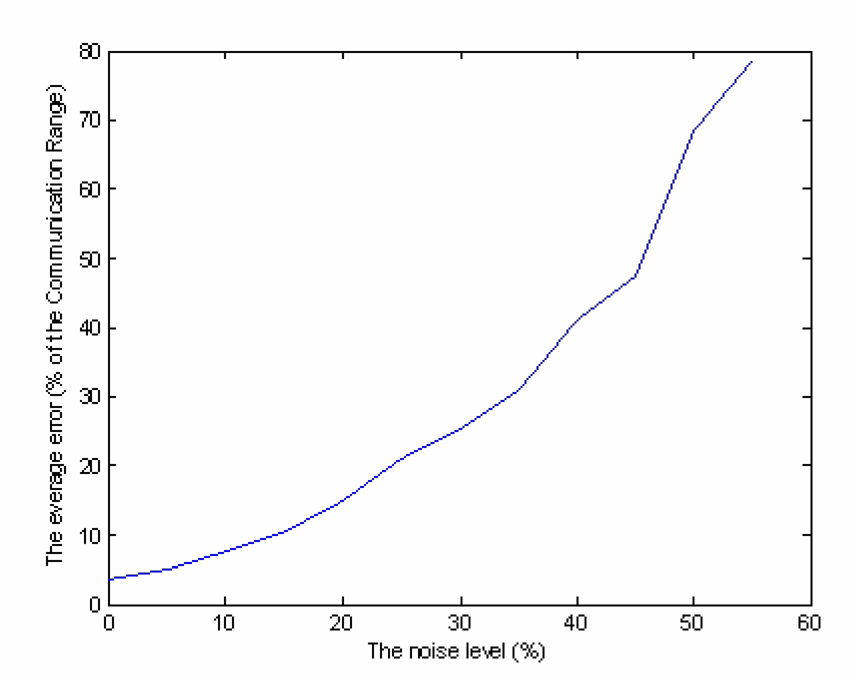

Figure 4.5. The noise effect in different noise level.

In Figure 4.5, the x label is the different noise level in percentage and the y label is the average error of the pairwise distances in percentage of the communication range. The minimum average error is 3.63% in the noise level of 0% and the maximum average error is 78.4% in the noise level of 55%. The localization algorithm breaking point is in the noise level around 45%. It means that if the noise level is continually increasing at this point, the average error of pairwise distances will increase faster than before and the quality of the static nodes localization becomes worse.

4.3.2 Beacon Selection in Static Node Localization

For transforming the relative coordinates of the static node location to the reference coordinates, 3 beacons are chosen to provide this transformation method.

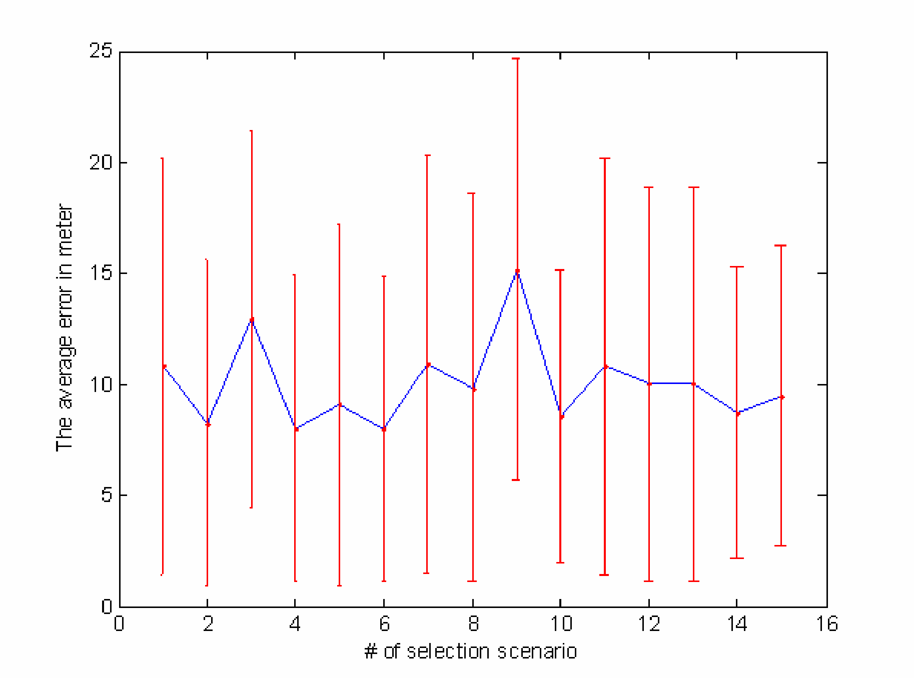

Figure 4.6. Beacons selection in static nodes localization

In Figure 4.6, the x label is the sequence number of 15 examples, the y label is the average error between the ground truth coordinates and the reference coordinates. The red lines are the errorbars of the Standard Deviation (STD) value for each example. Maximum average error is 15.18 meters and it occurs in the 9th sample. Minimum average error is 8.02 meters and it occurs in the 6th sample. Average value of these 15 examples is 10.056 meters. Maximum STD value is 9.47 and it occurs in the 9th sample. Minimum STD value is 6.57 and it occurs in the 10th sample.

4.3.3 The Number of Nodes Participating to Mobile Node Localization

By applying 20 different reference trajectory to simulate in a same static nodes setup, we recorded the number of static nodes participating to compute the temporal locations of mobile node. In these 20 samples, the average number of participating nodes is

5.3772, the maximum number is 6.9627 and the minimum number is 3.1079.

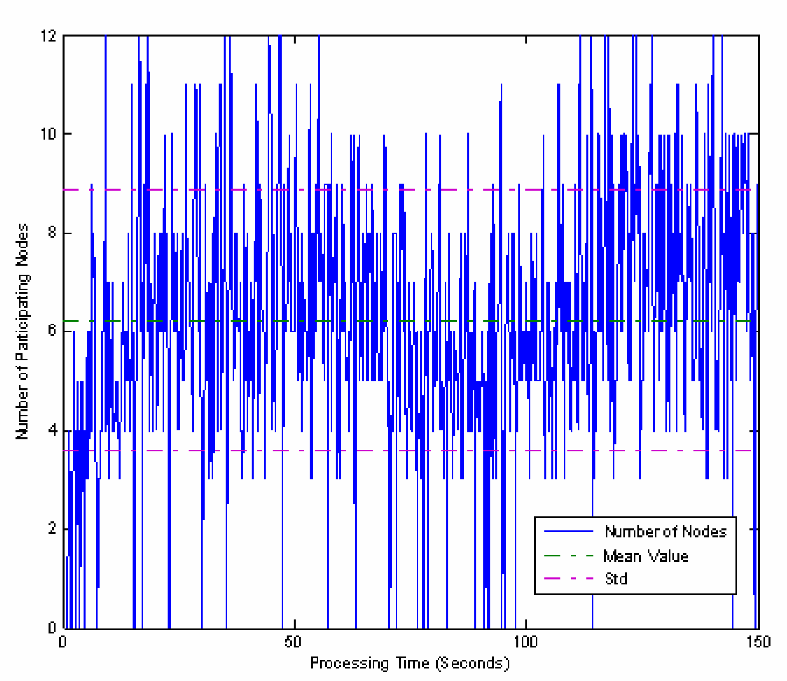

Figure 4.7. The recorded number of participating nodes in 12th samples

In 12th sample, the x label is the time slot from 0 to 150 seconds and the sample time is 0.2 seconds, the y label is the number of the participation nodes. The maximum number is 12 and the minimum number is 0. The average number is 6.226 which is the mean value in Figure 4.7. The STD value is 2.64.

4.3.4 Mobility Control Accuracy

In this scenario, we have collected the data from 10 different examples. Based on the obtained data, we analyzed the mobility control accuracy of the simulation system.

The following 3 plots show the obtained data from 6^{th} example. The blue lines are the difference of distances, the green lines are the average values and the red lines are the STD errorbars. In these 3 plots, the x label is the time slots from 0 to 150 seconds and the sample time is 0.2 seconds, the y label is the difference of distances in meters. The average number of the participating nodes in this example is 4.8455.

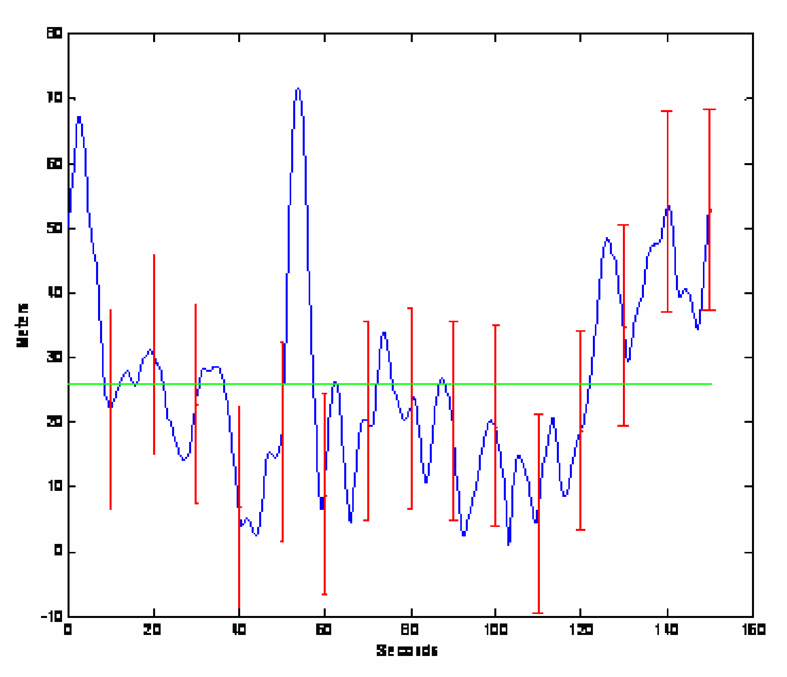

Figure 4.8. Mobile node distances to the reference point

Figure 4.8 shows the difference of the distance between the mobile node position and the reference point in each sample time. During the whole simulation process, the maximum distance is 71.6663 meters and the minimum distance is 0.8372 meters. The average value of the distances is 25.9345 meters and the STD value is 15.4381. Here should be notice that the reference point is not the point in reference coordinates. It is a pre-defined point which exactly moves along the reference trajectory in each sample time. By computing the distance between the mobile node position and the reference point, we can analyze how well the mobile follows the trajectory.

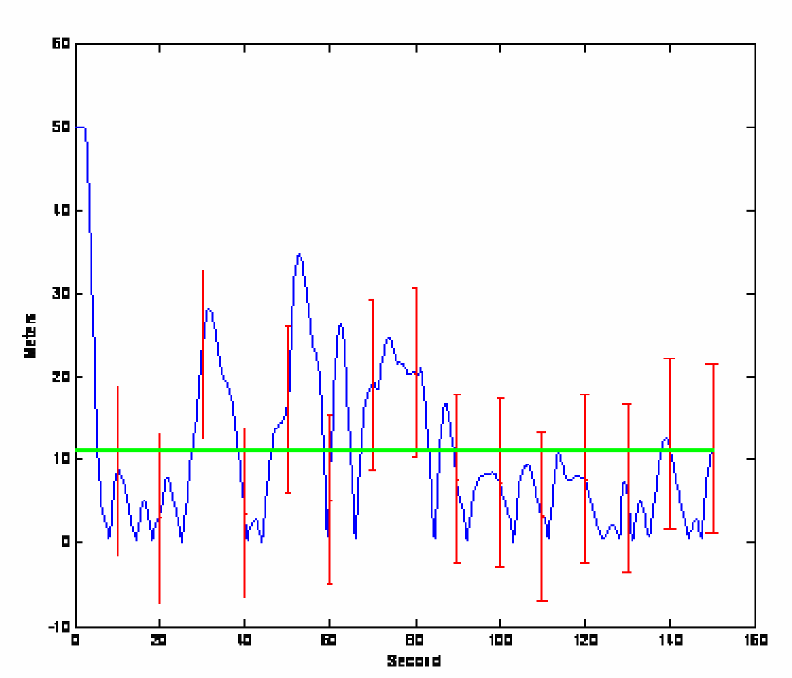

Figure 4.9. Mobile node distance to the reference path

Figure 4.9 shows the shortest distance between the mobile node position and the reference path in each sample time. During the whole simulation process, the maximum distance is 50 meters and the minimum distance is 0.0457 meters. The average value of the distances is 11.2524 meters and the STD value is 10.1593.

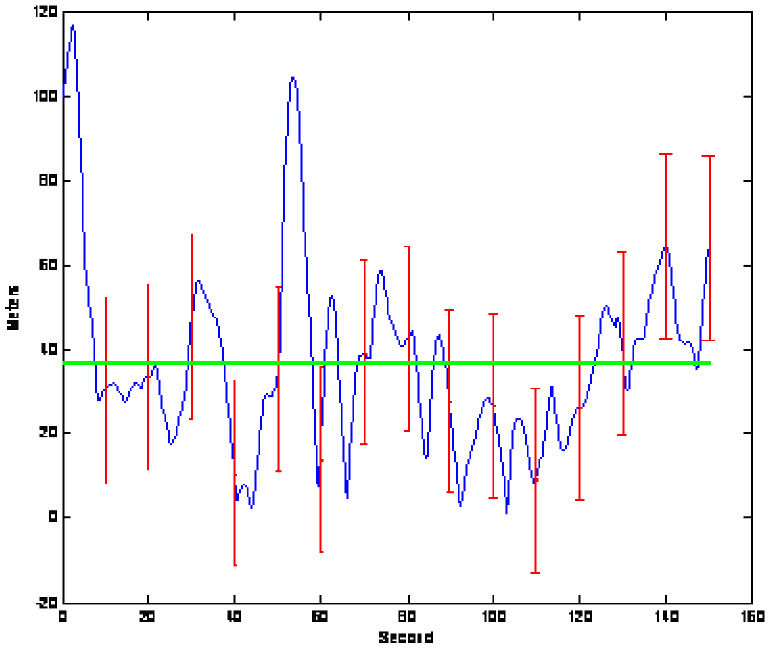

Figure 4.10. The summation value of the Dr and Dp

Figure 4.10 shows the summation value of the Figure 4.8 and Figure 4.9. The maximum value is 121.6663 meters and the minimum distance is 0.8829 meters. The average value of the distances is 37.1869 meters and the STD value is 21.8421.

Table 4.1. The obtained value from the ‘Mobility Control Accuracy’ scenario.

Scenario #	Max Dr	Min Dr	Max Dp	Min Dp	Mean Dr	Mean Dp	Mean Sum	Std Dr	Std Dp	Std Sum	mean n
1	72.5950	0.2071	50.1155	0.0295	26.4766	15.0323	41.5089	13.9241	12.2414	24.1674	2.5539
2	50.0000	3.9979	30.3085	0.0981	25.2016	9.6483	34.8499	11.1699	7.0123	13.7088	3.9068
3	57.3704	0.5009	50.0000	0.0717	23.7560	9.3960	33.1520	13.5854	9.9662	20.6916	6.0692
4	55.0174	0.9113	50.0000	0.0793	18.2181	10.6641	28.8822	9.2231	8.9618	16.4975	6.5326
5	54.3131	0.3984	37.2788	0.0911	21.7720	11.1957	32.9677	11.7904	9.0211	18.9029	4.0013
6	71.6663	0.8372	50.0000	0.0457	25.9345	11.2524	37.1869	15.4381	10.1593	21.8421	4.8455
7	90.2731	0.3160	50.0000	0.0407	34.7857	13.4256	48.2113	19.2490	9.8431	23.6299	3.6418
8	50.0000	0.9376	33.8662	0.0090	17.0478	6.2409	23.2887	8.4965	6.0303	11.6151	7.0333
9	71.4058	0.5805	50.0000	0.1731	21.0759	10.9154	31.9914	14.7299	12.4988	25.8749	3.4394
10	111.8351	1.3232	26.0803	0.1752	30.1705	7.0832	37.2536	26.8352	5.8948	28.9348	4.3941

In table 4.1, the values of ‘Max Dr’ and ‘Min Dr’ are the maximum and minimum value of the distance between the mobile node position and the reference point in each example, the values of ‘Max Dp’ and ‘Min Dp’ are the maximum and minimum value of the distance between the mobile node position and the reference path, the values of ‘Mean Dr’, ‘Mean Dp’ and ‘Mean Sum’ are the average value of the Dr, Dp and Summation of Dr and Dp, the values of ‘Std Dr’, ‘Std Dp’ and ‘Std Sum’ are the STD value of the Dr, Dp and Summation of Dr and Dp the values of ‘mean n’ are the average number of the participating static nodes. Except the values of the STD and ‘mean n’, all units for other values are in meters.

The average values of ‘Max Dr’ and ‘Min Dr’ are 68.4476 meters and 1.001 meters. The average values of ‘Max Dp’ and ‘Min Dp’ are 42.4679 meters and 0.0813 meters. The average value of ‘Mean Dr’ is 24.4439 meters, the average value of ‘Mean Dp’ is 10.4854 meters and the average value of ‘Mean Sum’ is 34.9293 meters. Finally, the average number of the participating nodes is 4.46418.

4.3.5 The Effect of the Communication Range

By changing the different communication range level, its effect to the simulations can be analyzed. As the assumption, the communication range level will be increased from 40 meters to 200 meter and each step is 20 meters.

Based on the collected data, when the communication range level is less than 100 meters, the mobile node does not move. The following plot shows the performance of the simulation in different communication range levels.

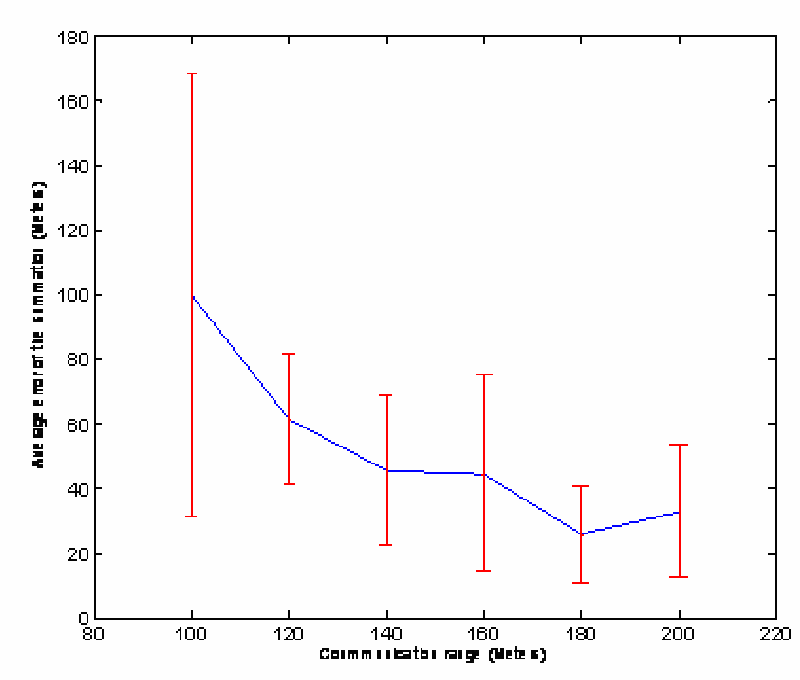

Figure 4.11. The performance of the simulation in different communication range level

In Figure 4.11, the x label is the different levels of the communication range and the y label is the average error from the ‘Mean Sum’. The maximum average error is 99.6644 meters and it occurs when the communication range level is 100 meters. The minimum average error is 25.8037 meters and it occurs when the communication range level is 180 meters. The red line is the errorbar of the STD value for each communication range level. The average value of the errors is 51.5887 meters.

Table 4.2. The obtained value from the 'Effect of Communication Range' scenario.

Range level	Max Dr	Min Dr	Max Dp	Min Dp	Mean Dr	Mean Dp	Mean Sum	Std Dr	Std Dp	Std Sum	mean n
40	424.2604	50	50	50	277.0369	50	327.0369	105.9173	1.28E-13	105.9173	0
60	424.2604	50	50	50	277.0369	50	327.0369	105.9173	1.28E-13	105.9173	0
80	424.2604	50	50	50	277.0369	50	327.0369	105.9173	1.28E-13	105.9173	0
100	196.963	8.6015	83.1832	0.0238	69.6018	30.0626	99.6644	49.4106	22.3583	68.5744	1.783
120	85.3181	17.4951	50	0.1522	42.4598	18.8569	61.3166	15.0462	10.5934	20.4966	3.0346
140	76.1682	1.5195	50	0.1691	31.5261	13.9426	45.4687	15.9201	10.9455	23.0271	2.7989
160	66.1306	0.9212	66.1186	0.0312	25.4353	19.059	44.4943	14.6183	16.6657	30.4297	4.0892
180	56.2298	0.403	50	0.0277	17.4047	8.399	25.8037	8.6029	7.469	14.9167	4.7483
200	60.3773	1.089	50	0.0552	21.3096	11.4751	32.7847	11.0491	10.6288	20.4986	4.4088

In Table 4.2, the definitions of the names for each value are the same as the definitions in Table 4.1. This table shows the whole obtained value of the communication range level from 40 meters to 200 meters.

When the communication range level is between 40 meters and 80 meters the mobile node is not guided by the command center, because the average number of the participating nodes is 0 for these three levels. It means that the connectivity of the network is in the worst situation. The average number of the participating nodes of the communication range from 100 meters to 200 meters is 3.4771. Starting from the 100 meters of the communication range level, the mobile node starts to move. Because of the average number of the participating nodes is still not enough, the value of the 'Max Dr' and 'Max Dp' are large.

5 Conclusions

In the past few years, WSNs has become a popular research direction in the context of wireless communication networks. Generally, Wireless Sensor Networks have several attractive properties. They can be applied in many different critical applications; the sensor nodes are in low-cost and interchangeable; the malfunction of some nodes does not affect the whole operation of the system, etc. Mobility control is one interesting research topic in the context of WSNs.

As the target of this Master's Thesis, we focus on the research of how a mobile node is guided to move in a WSN. By using the PiccSIM simulation platform, we have simulated the different scenarios and analyzed the obtained result.

According to the results of the 'Static Sensor Node Localization' testing scenarios, the tolerated noise level is about 45% and it is the localization algorithm breaking point as shown Figure 4.5. If the noise level is higher than 45%, the estimated positions of each static sensor node are useless.

The beacon selection is another important factor which will affect to the localization result. If the locations of the selected beacons are very near of each other and they are almost located to the same line, the estimated positions of the static nodes will have large errors compared to the absolute ground truth positions.

The average values of the distance to the reference point (Mean Dr) and the distance to the reference trajectory (Mean Dp) in Table 4.1 show how well the mobile node follows the trajectory. The maximum value of the Mean Dr is 34.7857 meters. It means the average error in 7th scenario is about 17.4% to the communication range of 200 meters. The average value of the Mean Dr is 24.44 meters and it is about 12.2% to the communication range. In Mean Dp, the maximum value is 15.0323 meters and it occurs in 1st scenario. It is about 7.5% to the same communication range. The average value of the Mean Dp is 10.49 meters and it is about 5.24% to the communication range. The average number of static nodes participating to temporal position computation is 4.46418. It proves that the mobile node has enough distance measurements to get its temporal location computed in each sample time moment.

In addition, the different communication range levels also affect the results of the simulations. Once the communication range is less than 100 meters, the connectivity of the WSN is not enough to provide the distance measurement and the data transmission. If the communication range is larger than 100 meter or even equal, the mobile node will start to move along the reference trajectory and the quality of the mobility control mainly depends on the average number of the static nodes participating to mobile node temporal position estimation

In 2D environment, it requires at least 3 static nodes participating to estimate the temporal location of mobile node in each sample time. But in some cases, when the average number of the participating static nodes is less than 3, the system still succeeds to guide the mobile node from the beginning to the end. The 1st sample in Table 4.1 is a typical example. The most reasonable explanation is the applied KF, which provides an estimate to the missing temporal position.

REFERENCES

ARM (2008). Intrinsyc's Cerf Cube 255 [online] [cited 21.04.2008]. Available from World Wide Web:
<URL: http://www.arm.com/markets/ embedded_solutions/armpp/4897.html>

Benkic, Karl & Peter Planinsic (2007). Custom Wireless Sensor Network based on ZigBee. In: *ELMAR*, pp. 259- 262

BTnode (2008). *BTnodes: A Distributed Environment for Prototyping Ad Hoc Networks* [online] [cited 21.04.2008]. Available from World Wide Web:
<URL: http://www.btnode.ethz.ch/>.

Callaway, Edgar H. (2004). Wireless Sensor Networks: Architectures and Protocols. Florida: CRC Press.

Chen, WeiPeng & Jennifer Hou (2005). Data Gathering and Fusion in Sensor Networks. In: *HANDBOOK OF SENSOR NETWORKS ALGORITHMS AND ARCHITECTURES*, pp. 493- 526. Stojmenovic, Ivan. Canada etc.: John Wiley & Sons.

Crossbow (2008a). *MICA-2 Datasheet* [online]. Crossbow: Wireless Sensor Networks. [cited 21.04.2008]. Available from Internet:
<URL: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/
MICA2_Datasheet .pdf>

Crossbow (2008b). *MICA-z Datasheet* [online]. Crossbow: Wireless Sensor Networks. [cited 21.04.2008]. Available from Internet:
<URL: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/
MICAz_Datasheet.pdf>

Crossbow (2008c). *TelosB Datasheet* [online]. Crossbow: Wireless Sensor Networks. [cited 21.04.2008]. Available from Internet:
<URL: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/
TelosB_Datasheet.pdf>

Crossbow (2008d). *Stargate Datasheet* [online]. Crossbow: Wireless Sensor Networks. [cited 21.04.2008]. Available from Internet:
<URL: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/
Stargate_Datasheet.pdf>

Crossbow (2008e). *Micaz Datasheet* [online]. Crossbow: Wireless Sensor Networks. [cited 21.04.2008]. Available from Internet:
<URL: www.xbow.com/products/product_pdf_files/wireless_pdf/
6020-0060-01_a_micaz.pdf>

Edgar H. & Jr Callaway (2007). The Wireless Sensor Network MAC. In: *HANDBOOK OF SENSOR NETWORKS ALGORITHMS AND ARCHITECTURES*, pp. 239-276. Stojmenovic, Ivan. Canada etc.: John Wiley & Sons.

Elson, Jeremy & Kay Römer (2002). Wireless Sensor Networks: A New Regime for Time Synchronization. In: *Proceedings of the 1st Workshop on Hot Topics in Networks*, pp. 149- 154

Hill, Jason & David Culler (2002), Mica: a Wireless Platform for Deeply Embedded Networks, In: *IEEE Micro*, vol. 22, Issue 6, pp. 12-24

Hill, Jason & Mike Horton (2004). The platforms enabling wireless sensor networks. In: *COMMUNICATIONS OF THE ACM*, vol. 47, No. 6, pp. 41–46

Howstuffworks (2008). *A Typical Mote* [online] [cited 21.04.2008]. Available from World Wide Web:
<URL: http://computer.howstuffworks.com/ mote4.htm>

IEEE 802.15 (2008). *IEEE 802.15 Working Group for WPAN* [online] [cited 21.04.2008]. Available from World Wide Web:
<URL: http://www.ieee802.org/15/ >

Karl, Holger & Marc Lobbers (2003). *A Data Aggregation Framework for Wireless Sensor Networks* [online] [cited 21.04.2008]. Available from World Wide Web:
<URL: http://www.tkn.tu-berlin.de/publications/papers/TechReport_03_016
.pdf>

Karl, Holger & Willig Andreas (2005). *PROTOCOLS AND ARCHITECTURES FOR WIRELESS SENSOR NETWORKS*. England: John Wiley & Sons.

Koushanfar, Farinaz & Miodrag Potkonjak (2008). *FAULT TOLERANCE IN WIRELESS SENSOR NETWORKS* [online] [cited 21.04.2008]. Available from Internet:
<URL: http://www.ece.rice.edu/~fk1/ papers/ chapter-FT_04.pdf>

Levis, Philip & David Culler (2002). *Mate: A Tiny Virtual Machine for Sensor Networks*. In: Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS X)

Levis, Philip & Sam Madden (2004). *TinyOS: An operating system for wireless sensor networks*. Ambient Intelligence, New York

Lymberopoulos, Dimitrios & Andreas Savvides (2008). *A Wireless Sensor Node Architecture for Exploring Distributed Sensor Network Applications* [online] [cited 21.04.2008]. Available from Internet:
< URL: http://www.eng.yale.edu/enalab/publications/XYZ_paper.pdf>.

Martincic, Fernando & Schwiebert, Loren (2005). Introduction to Wireless Sensor Networking. In: *HANDBOOK OF SENSOR NETWORKS ALGORITHMS AND ARCHITECTURES*, pp. 1- 40. Stojmenovic, Ivan. Canada etc.: John Wiley & Sons.

Nethi, Shekar & Mikael Pohjola (2007). Simulation case studies of wireless networked control systems. In: *Performance monitoring and measurement of heterogeneous wireless and wired networks*, pp.100-104. Helsinki, Finland.

Pohjola, Mikael & Shekar Nethi (2007). Platform for Emulating Networked Control Systems in Laboratory Environments. In: *Proc. IEEE Inter-national Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM 2007)*, Helsinki, Finland.

Ren, Xiuli (2006). Security Methods for Wireless Sensor Networks. In: *Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on*, pp. 1925-1930.

Santi, Paolo (2005). *Topology Control in Wireless Ad Hoc and Sensor Networks.* England: John Wiley & Sons.

Savvides, Andreas (2005). *Physical Layer, MAC and theIEEE 802.15.4 MAC* [online] [cited 21.04.2008]. Available from World Wide Web:
< URL: http://www.eng.yale.edu/enalab/courses/2005f/eeng460a/lec10.ppt>.

Shang, Yi & Wheeler Ruml (2003a). Localization from mere connectivity. In: *ACM MobiHoc*, pp. 201–212. Annapolis, MD.

Shang, Yi & Wheeler Ruml (2003b). Improved MDS-Based Localization. In: *23rd Conference of the IEEE Communications Society*. Hong Kong, China.

Shucker, Brian & Jeff Rose (2005). Embedded Operating Systems for Wireless Microsensor Nodes. In: *HANDBOOK OF SENSOR NETWORKS ALGORITHMS AND ARCHITECTURES*, pp. 137- 198. Stojmenovic, Ivan. Canada etc.: John Wiley & Sons.

Sohraby, Kazem (2007). *WIRELESS SENSOR NETWORKS: Technology, Protocols, and Applications.* Canada, etc.: John Wiley & Sons.

SOS 2.x Home Page (2008). *Introduction to SOS* [online] [cited 21.04.2008]. Available from World Wide Web:
< URL: https://projects.nesl.ucla.edu/ public/sos-2x/doc/>

UCLA (2008). Testbed General Overview [online] [cited 21.04.2008]. Available from World Wide Web:
< URL: http://lecs.cs.ucla.edu/Resources/ testbed/testbed-overview.html>.

University of New South Wales (2008). *Mini Hardware Survey* [online] [cited 21.04.2008]. Available from World Wide Web:
< URL: http://www.cse.unsw.edu.au/~sensar/hardware/hardware_survey.html>.

Wikipedia (2008). *Standards for Wireless Local Area Network* [online] [cited 21.04.2008]. Available from World Wide Web:
< URL: http://en.wikipedia.org/wiki/IEEE_802.11>.

ZigBee Alliance (2008). *ZigBee Alliance* [online] [cited 21.04.2008]. Available from World Wide Web:
<URL: http://www.zigbee.org/en/index.asp>.