Institutionen för datavetenskap
Department of Computer and Information Science

Master's Thesis

An Energy-efficient Handover Algorithm for Wireless Sensor Networks

by

Fredrik Säveros

LIU-IDA/LITH-EX-A--13/056--SE

2013-10-24

Linköpings universitet

Master's Thesis

An Energy-efficient Handover Algorithm for Wireless Sensor Networks

by

Fredrik Säveros

LIU-IDA/LITH-EX-A--13/056--SE

2013-10-24

Supervisors: Joakim Mjårdner
Attentec

Fang Wei-Kleiner
Department of Computer and Information Science
Linköping University

Examiner: Associate Professor Niklas Carlsson
Department of Computer and Information Science
Linköping University

Abstract

The recent advancements in the communication area have enabled the Internet of Things, a paradigm which extends the Internet to everyday objects. The Internet of Things enables many new applications, but also comes with great challenges; effective communication under limited power supply being the perhaps most important one.

This thesis presents the design, implementation, and evaluation of an energy-efficient handover algorithm for the main building block in the creation of the Internet of Things: wireless sensor networks. Our low-power handover design is based on a careful breakdown and analysis of the potential power consumption of different components of the handover process. With the scanning part of the process being identified as the main drain of energy, the algorithm is designed to place the majority of the scanning responsibility on the mains powered access points, rather than on the low-power mobile nodes.

The proposed algorithm has been implemented and its functionality and low power consumption have been empirically evaluated. We show that the design can reduce the energy consumption by several orders of magnitude compared to existing handover algorithms for wireless sensor networks. In addition, interesting fading effects were discovered in a sparsely deployed network scenario with limited access point coverage; most likely due to multipath propagation. For this case the handover performance was greatly reduced, relative our more normal coverage scenario. While these results illustrate that the absolute energy savings will differ from scenario to scenario, the potential energy savings made possible by the proposed algorithm significantly reduce the battery requirements of the devices in the emerging landscape of the Internet of Things; potentially even opening the door for new devices to connect.

Keywords: handover, handoff, wireless sensor networks, low power, mobility

Acknowledgements

“Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.”

— Antoine de Saint-Exupéry

This report is my final thesis for the conclusion of the Computer Science and Engineering program at Linköping University. It is a part of a Master’s degree and corresponds to 30 ECTS. The work has been carried out at Attentec in Linköping and examined at Department of Computer and Information Science at Linköping University.

I am profoundly grateful to my examiner **Associate Professor Niklas Carlsson** for his expert guidance and continuous encouragement throughout to see that this thesis reaches its target since its commencement to its completion. I really appreciate the feedback from my opponent **Niklas Ekendahl** and my supervisor **Fang Wei-Kleiner** which led to a better report. In addition, a thank you to **Marjorie Carleberg**, who helped me improve the report from the linguistic point of view.

I also want to extend my gratitude to my colleagues at **Attentec** for their help and advice on various issues. Special thanks to **Anders Weiland** and **Malcolm Sundberg** for giving me the opportunity to do this thesis. I am also particularly grateful to **Joakim Mjårdner** for the support and feedback on the report.

Tran Duy Khanh and **Edwin Krikke** at **Open Controller** are appreciated for their help on the implementation part.

Last but definitely not least I want to thank **my family** for their love and support during my entire time at the university.

Linköping, October 2013

Fredrik Säveros

Contents

List of Figures	x
List of Tables	xi
List of Algorithms	xii
List of Abbreviations	xiii
1 Introduction	1
1.1 Connecting Everything	1
1.2 Problem Statement	2
1.3 Purpose	3
1.3.1 Scope	3
1.4 Method	3
1.5 Contributions	4
1.6 Prerequisites	4
1.6.1 Terminology	4
1.7 Thesis Structure	5
2 Background	7
2.1 Handover	7
2.1.1 Handover Categories	8
2.1.2 Connection Regions	9
2.2 Wireless Sensor Networks	10
2.2.1 Sensor Node	10
2.2.2 Network	11
2.2.3 Internet of Things	13
2.3 Related Work	14
3 Algorithm Design	17
3.1 Traditional Communication Systems: A Comparison	17
3.2 Considerations	18
3.2.1 Energy	18
3.2.2 Type of Handover	19
3.2.3 Position Knowledge and Mobility	20
3.2.4 Quality of Service	21
3.2.5 Scalability	21
3.2.6 Redundant Handovers	22

3.3	The Parts of a Handover Algorithm	22
3.3.1	Observe	23
3.3.2	Initiate	24
3.3.3	Scan	25
3.3.4	Select	25
3.3.5	Data Link Layer Switch	26
3.3.6	Network Layer Switch	26
3.4	Handover and the Protocol Stack	26
3.4.1	Data Link Layer Protocol	27
3.4.2	Network Layer Protocol	27
4	The Proposed Algorithm	29
4.1	Overview	29
4.1.1	Assumptions	29
4.1.2	Algorithm Mechanics	31
4.2	Link Monitoring	32
4.2.1	Parameters	33
4.2.2	Initiation	34
4.3	Intra-domain Access Point Selection	35
4.3.1	Scanning	35
4.3.2	Selection	36
4.4	Inter-domain Access Point Selection	39
4.4.1	Scanning	39
4.4.2	Selection	40
4.5	Switch	40
4.5.1	Data Link Layer Switch	40
4.5.2	Network Layer Switch	40
4.6	Timers	41
4.6.1	<i>handover_select</i> Time Out	41
4.6.2	<i>intra_domain_handover</i> Time Out	42
4.7	Alternative Approaches	43
4.7.1	Stability	43
4.7.2	Observation	43
4.7.3	Initiation	44
4.7.4	Selection	44
5	Implementation	45
5.1	Limitations	45
5.2	Hardware	45
5.2.1	Mobile Node	46
5.2.2	Access Point	47
5.3	Software	48
5.3.1	Operating System	48
5.3.2	Network	48
5.4	IEEE 802.15.4	48
5.4.1	Physical Layer	48
5.4.2	MAC Layer	49
5.4.3	Device Classes	49
5.4.4	Network	49
5.4.5	Network Management	50

5.4.6	Optimizations	50
5.5	Network Overview	50
5.6	Algorithm at Mobile Node	51
5.6.1	Required Functions	51
5.6.2	Command Line Interface	53
5.6.3	Finite State Machine	53
5.7	Algorithm at Access Point	55
5.7.1	Required Functions	55
6	Empirical Evaluation	57
6.1	Selected Evaluation Method	57
6.2	Instrumentation	58
6.3	Algorithm Parameters	58
6.3.1	Timers	58
6.3.2	Thresholds	59
6.3.3	Stability	60
6.4	Performance Metrics	61
6.4.1	Handover Delay	61
6.4.2	Number of Handovers and Packet Loss	62
6.4.3	Energy Consumption	67
7	Discussion and Conclusion	71
7.1	Handover in Wireless Sensor Networks	71
7.2	Algorithm Analysis	72
7.2.1	Energy Consumption	72
7.2.2	Quality of Service	73
7.2.3	Energy Efficiency	73
7.2.4	Weaknesses	73
7.3	Future Work	74
7.4	Final Words	75
A	The Proposed Algorithm: Parameters and Functions	77
A.1	Algorithm Parameters	77
A.2	Access Point and Mobile Node: Common Functions	78
A.3	Access Point: Functions	78
A.4	Mobile Node: Functions	79
B	The Proposed Algorithm: Flowchart	81
B.1	Monitor	82
B.2	Selection and Switch: Mobile Node	83
B.3	Selection: Access Point	84
B.4	Switch: Access Point	85
C	Command Line Interface	87
	Bibliography	89

List of Figures

1.1	Illustration of a handover	2
2.1	Handover's relation to the OSI model protocol stack	8
2.2	A typical sensor node architecture	11
4.1	Intra-domain handover algorithm	31
4.2	Intra-domain handover algorithm message formats	32
4.3	Relative velocity	34
4.4	Link quality fuzzy logic sets in relation to the connection regions	34
4.5	Neighbor channels message format	40
4.6	Timing of messages sent during an intra-domain handover	41
5.1	A mobile node	46
5.2	An access point	47
5.3	A gateway	47
5.4	Mobile node's finite state machine	54
6.1	Evaluation of message timings	59
6.2	Packet error rate vs. link quality	60
6.3	Evaluation of handover delays	62
6.4	Test setup, sparse deployment of access points	63
6.5	Evaluation of number of handovers and packet loss, test case 1	64
6.6	Probability distribution of where the handovers are done, test case 1	64
6.7	Access point fading	65
6.8	Test setup, dense deployment of access points	65
6.9	Evaluation of number of handovers and packet loss, test case 2	66
6.10	Probability distribution of where the handovers are done, test case 2	66
6.11	Access point density in relation to number of handovers and packet loss	70
B.1	The mobile node's monitor flowchart	82
B.2	The mobile node's selection and switch flowchart	83
B.3	The access point's selection flowcharts	84
B.4	The access point's switch flowchart	85
C.1	The mobile node's command line interface	88

List of Tables

1.1	The phases of the method used	4
2.1	An example protocol stack	13
4.1	Handover decision fuzzy logic matrix	35
4.2	Candidate access point decision fuzzy logic matrix	37
4.3	The proposed algorithm's times	42
4.4	Comparison of selection mechanisms	44
A.1	Algorithm parameters	77
A.2	Common functions required by the algorithm	78
A.3	Access point functions required by the algorithm	78
A.4	Mobile node functions required by the algorithm	79

List of Algorithms

4.1	The mobile node's observe procedure	34
4.2	The mobile node's handover initiation procedure	35
4.3	Procedure for executing a handover to the selected access point	37
4.4	The procedure for handling received messages at the access point	38
4.5	The procedure for handling received messages at the mobile node	38
4.6	The mobile node's inter-domain handover procedure	39
4.7	The access point's procedure for performing a network layer switch	40
5.1	The access point's and mobile node's main loop	51
5.2	Add link quality to window	52
5.3	Calculate average link quality	52
5.4	Handover initiation decision function	52
5.5	Procedure for finding the best access point	53
5.6	Candidate access point potential	56

List of Abbreviations

AP	Access Point
BER	Bit Error Rate
CSMA	Carrier Sense Multiple Access
FDMA	Frequency Division Multiple Access
FFD	Full Function Device
GW	Gateway
IoT	Internet of Things
IP	Internet Protocol
LQ	Link Quality
MAC	Medium Access Control
MN	Mobile Node
OSI	Open Systems Interconnection
PAN	Personal Area Network
PER	Packet Error Rate
QoS	Quality of Service
RFD	Reduced Function Device
RSSI	Received Signal Strength Indication
SNR	Signal-to-Noise Ratio
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Network

Introduction

THE technology of Wireless Sensor Networks (WSNs) has evolved at a furious pace thanks to a great deal of attention from both researchers and industry. The main reason for this attention is the many applications it enables. In the past, the main focus of WSN research studies and deployments have been on static sensor networks. However, many of the recent applications in the area require mobile sensor nodes, something that poses unique challenges such as topology control, routing, addressing, handover and security. Some examples include the use of WSNs for health parameters monitoring [1], military [2] and animal monitoring [3]. Without efficient mobility mechanisms, the application areas of WSNs will be limited.

1.1 Connecting Everything

The technology development of WSNs has enabled the cost for the sensor units to drop while the reliability increases. This creates a virtuous circle whereby increased volumes drive further technology development. The idea of connecting these sensor units not only locally but also globally via the Internet has led to the rise of a novel paradigm in communication: the Internet of Things (IoT). This new paradigm has gained a lot of attention as one of the most promising techniques of the 21st century. The IoT is the vision of “connecting everything”, e.g. cars, toys, animals and even flowers. This is done by integrating small computers (sensor nodes) within these things, having the ability to communicate with each other via the Internet. The vision is 50 billion connected devices by year 2020. We will give a more comprehensive overview of the IoT in Section 2.2.3.

Wireless sensor nodes are often battery powered and hence operate on an extremely frugal energy budget. The IoT vision requires their lifetime to be long since battery replacement is not an option for networks with thousands of physically embedded nodes. This leads to energy being one of the scarcest resources and this should be reflected in all parts of the design of these networks.

1.2 Problem Statement

Mobility and energy consumption are two of the most important problems to tackle in the WSN area. In this thesis we focus on handover, the mechanism of switching Access Point (AP) in a wireless network, thus enabling mobility. Handover algorithm, mechanism and process are used interchangeably throughout the thesis. Handover is needed when a Mobile Node (MN) moves out of range of the current AP (there may be other reasons for doing a handover). A typical handover is illustrated in Figure 1.1. Changing AP in a wireless network will cause degradation in the Quality of Service (QoS) by introducing for example delay and/or packet loss. It is the handover mechanism's task to mitigate these impacts. Conceptually, a handover is a simple process, but in order to make it efficient in terms of energy and QoS, several parameters and aspects need to be analyzed thoroughly.

Figure 1.1 Illustration of a handover.

There exist several definitions of handover in the literature, we define it as follows.

Handover. The operation of switching associated access point in a wireless network by a predefined process. □

We do not require a connection (other than to the network itself) to be maintained. However, a predefined process is required; the switch of AP shall (in most cases) not require a complete reconnect to the network.

Our main question at issue is:

How to keep the energy consumption at a minimum when doing a handover?

Moreover, this energy usage minimization is to be done while providing best possible QoS, or more formally, we want to minimize the energy usage given a certain expected QoS.

Energy efficiency. A handover algorithm is more *energy efficient* than another handover algorithm if it uses less energy to provide the same quality of service. □

Furthermore, in order to design a handover algorithm, there are several questions that need to be answered, the most important ones being:

- When should the handover be initiated?
- How is the new access point chosen?
- How is the actual switch done?

1.3 Purpose

The purpose of the thesis is to design a handover algorithm for WSNs with focus on low power. The algorithm should be adaptable to (almost) any system and protocol stack. We also intend to evaluate the algorithm. In order to do this, the algorithm is implemented and run on a WSN. The implementation is instrumented to measure certain metrics of interest, such as number of handovers and energy consumption.

An in depth discussion on the topic of handover algorithm design for WSNs will be held. Hopefully our work is able to assist others in future designs of handover algorithms by referring to our discussion and considerations.

1.3.1 Scope

The focus is on the design of a low power handover algorithm. Our main concerns are handover decisions, such as deciding when to handover and the mechanics for deciding the next AP. The actual switch, especially the network switch, will only be looked at briefly. A handover might affect several things, such as changes in topology, addresses and routing. How these are handled is only of secondary concern. The optimal value for several of the algorithm's parameters can only be decided when system details are known. Further, knowing the system details, the algorithm can be tailored for that particular system and be incorporated as part of the protocols. These kinds of system dependent optimizations are out of scope of the thesis, due to our intention to keep the algorithm generic.

The more advanced handover decision models such as pattern recognition and machine learning will not be considered; there exist other decision models more suitable for sensor devices [4]. Finally, throughout the thesis, the network is assumed to consist of static mains powered APs, i.e. their energy consumption are not of interest.

1.4 Method

The strategy is to divide the work into four main phases, summarized in Table 1.1. The literature study will mainly consist of recent research papers in the WSN area. The focus will be on handover and mobility, but related topics such as energy and protocols will also be considered. In order to get an overview of handover and WSNs, surveys and books in respective area will be used. In addition, data sheets and standards (e.g. IEEE 802.15.4) will be used for the implementation part. The algorithm design will be general, whereas the proof-of-concept implementation and evaluation will be done on specific hardware.

Table 1.1 The phases of the method used.

Phase	Purpose
Literature study	Study the basics of handover and WSNs in general as well as existing techniques and work.
Design	Identify potential design weaknesses of existing handover algorithms for WSNs and design a general handover algorithm that addresses these weaknesses.
Implementation	Implement a proof-of-concept version of the general handover algorithm designed in the previous phase.
Evaluation	Empirically evaluate the implemented algorithm.

1.5 Contributions

The main scientific contribution is a proposed handover algorithm with remarkably low energy consumption at the MNs. The algorithm is system independent and thereby adaptable to many kinds of WSNs. The only requirement is that the APs are mains powered.

Further, a thorough discussion of the general design of a handover algorithm for WSN is provided. The main focus is on handover decisions; when should the handover be initiated and which AP should be the new AP?

1.6 Prerequisites

The reader is assumed to have basic understanding of computer science and engineering, particularly computer networks.

1.6.1 Terminology

In this section we define the terminology used throughout the thesis.

Access point Network unit (often static) with routing capabilities that provides network access to MNs.

Beacon A message sent by an AP with purpose of notifying MNs of its presence. A beacon often contains some sort of network information.

Candidate access point A potential AP for the MN to switch to in a handover.

Channel The ID of a domain, contains a set of IDs.

Data link layer Layer 2 in the Open Systems Interconnection (OSI) model.

Domain Subnetwork.

Fuzzy logic Multi-valued logic, allowing degrees of truth.

ID Identifier (unique within the domain) assigned to a unit to be used for communication, e.g. address, frequency channel or time slot.

Link quality Measurement of the current link, e.g. Signal-to-Noise Ratio (SNR), Bit Error Rate (BER), Received Signal Strength Indication (RSSI)—or a combination of these.

Mobile node The mobile end device of the network.

Multipath propagation Radio signals reaching the receiver by two or more (different) paths.

Network Interconnected APs. The (connected) MNs are also considered a part of the network.

Network layer Layer 3 in the OSI model.

New access point The AP *to* which the MN is about to switch in a handover.

OSI model A communication system model which divides the system into abstraction layers.

Personal area network A network of devices with short range, centered around a person.

Previous access point The AP *from* which the MN is about to switch in a handover.

Quality of service Defined measures of performance in networks, such as delay, throughput, packet jitter (variation in latency) and packet loss [5].

1.7 Thesis Structure

The outline of the thesis is as follows:

Introduction	Chapter 1 introduces the thesis (this chapter).
Background	Chapter 2 gives an introduction to handover and WSNs. It also contains an overview of related works.
Algorithm Design	Chapter 3 discusses the general design of a handover algorithm for WSNs.
The Proposed Algorithm	Chapter 4 presents our proposed handover algorithm.
Implementation	Chapter 5 describes the implementation of the proposed algorithm.
Empirical Evaluation	Chapter 6 evaluates the implemented algorithm.
Discussion and Conclusion	Chapter 7 concludes the thesis with a closing discussion and conclusion.

Background

THIS chapter gives an introduction to handover and WSNs. We begin by introducing handover in general terms and look at different handover categories. The most important aspects of sensor nodes are described and we look at possibilities and challenges when connecting these, thus creating WSNs. The IoT is described in terms of WSNs. A brief overview of related work concludes the chapter.

2.1 Handover

The term handover (also known as handoff) first appeared in cellular network and satellite communication to support mobility [6]. Handover in cellular networks allows a person to move in the network by seamlessly transferring an ongoing call from one base station to another. In satellite communication the mechanism is responsible for transferring satellite control from one earth station to another. Later, when the wireless technology of computer networks advanced, handovers were applied to support the transfer of a connection from one access point to another, e.g. in a Wi-Fi network.

The basic idea of handover is to allow a node (e.g. cell phone, satellite or laptop) to move in a network and still retain connection. This is done by having a process for changing to a new access point (e.g. base station or earth station) as the node moves out of range of the current access point. A general handover is illustrated in Figure 1.1.

Conceptually, a handover is a simple process, but one should not be fooled; there are many things to consider to make it efficient. Wireless communication is very unreliable with much variation and interference in the connection. Not only is handover a process that operates in the area of wireless communication; it does so in the worst possible scenario: when a connection is just about to break.

The handover algorithm can be divided in three sub-processes¹ (compare with the questions about the handover algorithm asked in Section 1.2):

1. Monitor: the current connection has to be observed by the means of different parameters to be able to decide whether a handover should be initiated or not.
2. Selection: candidate APs has to be found and one of them selected.
3. Switch: the actual transfer of the connection from the previous AP to the new one.

The main purpose of handover (in most communication systems) is to provide QoS. Continuous data streams (e.g. audio and video) are intolerant to delay and jitter, i.e. intolerant to traffic interruptions caused by a connection which no longer can be maintained. A complete reconnection to the network is in most cases not acceptable. Therefore, a handover process is used, with the task of switching AP as seamless as possible.

If the communication system in question is an OSI model system, the handover mechanism is considered to be a cross-layer function, see Figure 2.1. Moreover, it is concentrated around the data link layer and the network layers.

Figure 2.1 Handover’s relation to the OSI model protocol stack. Handover is a part of the mobility management plane. [7]

More details on the handover process (from a WSN perspective)—including division into more sub-processes, problems and considerations—will be given as we discuss the design of a handover algorithm in Chapter 3.

2.1.1 Handover Categories

Handovers are often classified into different categories, depending on context.

Mobile vs. Network Controlled

In a mobile controlled handover the MN is actively participating in the entire handover process. The complexity on the MNs has to increase but it reduces the load on the network.

¹There exists a plethora of ways of dividing a handover algorithm into sub-processes in the literature.

In a network controlled handover the mobile node has a passive role; the handover is initiated by the network. This reduces the complexity of the MNs but increases the load on the network. A combination of the two is also possible, called mobile assisted handover [8].

Intra-domain vs. Inter-domain

An intra-domain handover is a handover where the previous AP and the new AP are in the same domain whereas an inter-domain handover is a handover where the APs are in separate domains. Domain can refer to various things: one example might be to consider two base stations controlled by different mobile switching stations (in a cellular network) to be in two different domains.

Vertical vs. Horizontal

A vertical handover means that the MN changes the link layer technology used for the connection. This could for example be a change from a LTE to a Wi-Fi connection (or vice versa). If the new AP uses the same link layer technology as the previous AP it is a horizontal handover.

Hard vs. Soft

Another example of classification is hard and soft handovers. A soft handover implies that the radio can use multiple channels at the same time. This enables the node to be connected to both the new and the previous AP, thus enabling a continuous transmission without breaks. These kinds of handovers are also known as make-before-break; the new connection is established before breaking the old one.

In a hard handover only one radio channel is used which means that the node is connected to only one AP at all times. Thus, a break in the transmission is needed in a hard handover. These types of handovers are also known as break-before-make; the old connection has to be broken before establishing the new one.

Forward vs. Backward

In a forward handover the handover is executed via the radio link established with the new AP, while a backward handover is characterized by a prediction and execution of handover via the existing link [8]. Forward handover has an advantage over the backward handover in the case of old link deterioration.

2.1.2 Connection Regions

Studies have revealed the existence of three distinct reception regions in a wireless link [9].

Connected

In the connected region a stable connection between the MN and AP exists.

Transitional

The transitional region is often not considered in simulation tools. It has been shown to be quite large, especially in WSNs. It is characterized by high-variance in reception rates and asymmetric connectivity.

Disconnected

The disconnected region is pretty much what it sounds like, the region in which no connection exists.

2.2 Wireless Sensor Networks

In this section we give an overview of WSNs, sometimes called wireless sensor and actuator networks (WSAN). We will keep it brief and focus on the things most important for the thesis. In order to fully understand all aspects of WSNs a comprehension of a broad spectrum of areas in computer science, e.g. operating systems, computer architecture, signal processing and computer networks to mention a few, is needed. Moreover, understandings of a whole range of topics in the field of electrical and computer engineering is also necessary.

We start by describing a sensor node, followed by a description of possibilities and challenges when connecting these in order to create WSNs. This section is ended by a description of how these intra-networks of WSNs can be connected, creating (at least part of) the IoT paradigm.

2.2.1 Sensor Node

A sensor node (also called mote) is a hardware unit with the capability of *a*) gathering information about its environment, *b*) processing the gathered information and *c*) communicating. It is a unit with very limited resources; it is often a cheap device with small memory, slow processor and limited energy due to battery powering. The sensor nodes often have stringent constraints on various aspects [7]; they may for example have strict requirements on a very limited physical size.

The most common hardware architecture [7] is illustrated in Figure 2.2 and the different parts are described further in the following paragraphs.

Sensing Subsystem

This subsystem consists of sensors that link the node to the physical world, by capturing information about a physical object or process, including the occurrence of events (i.e. changes in state such as a drop in temperature or pressure). Many sensor nodes also include actuators which allow them to directly control the physical world. For example, an actuator can be a valve controlling the flow of hot water or a pump that controls the amount of fuel injected into an engine. Analog-to-digital converters are a common part of this subsystem.

Figure 2.2 A typical sensor node architecture.

Processing Subsystem

The data gathered by the sensing subsystem are processed by this subsystem, consisting of a microcontroller and/or microprocessor. This is the brain of the sensor node providing intelligence to the system, used for controlling all other subsystems. Its main task is to gather the sensed information from the sensing subsystem, process it (for example by applying signal processing algorithms) and forward the result to the communication subsystem. The memory subsystem is often considered to be a part of the processing subsystem, consisting of a non-volatile memory (usually a flash memory) for storing the program, and a primary memory.

Communication Subsystem

This subsystem's purpose is to provide communication capabilities to the sensor node, often over short range radio. The most commonly used band is the 2.4 GHz ISM (Industrial, Scientific and Medical) band. This is a crowded band used by for example Wi-Fi, Bluetooth, ZigBee and microwave ovens.

Power Subsystem

The task of the power subsystem is to power all other subsystems and manage the energy in the sensor node. A sensor node is commonly battery-driven.

2.2.2 Network

WSNs are created by connecting several sensor nodes to each other. WSNs originate from Personal Area Networks (PANs); they are more or less low-power and low-rate wireless PANs. Technology development in wireless communication and electronics has enabled the cost for the sensor units to drop while the nodes are provided with more

energy-efficient processing power. This enables large sensor networks with a large amount of nodes, which in turn enables a wide range of applications [7]:

- Home automation
- Industrial process control
- Traffic sensors
- Monitored animals
- Monitored people, for example in a hospital environment
- ...

The area of WSNs is an active research area with many unsolved problems [7], including fault tolerance, scalability, reliability and security. Several other groups are investigating efficient hardware/software system architectures, signal processing algorithms and network protocols [10]. Realization of WSNs means satisfying tough constraints including cost, environment and power consumption [7]. A problem with WSNs is the existence of numerous protocols and subsystems in the literature making very different assumptions on the rest of the system and how the parts should interact. In fact, Culler et al. [11] argue that the lack of an overall WSN architecture is the primary factor limiting progress in the field because of reduced synergy between research efforts, not any specific technical challenge.

Structured vs. Unstructured

WSNs can be structured or unstructured. Both of these have so called sink nodes, to which the other nodes send their collected data. The sink node often has larger capabilities than the other nodes in the network. The unstructured WSNs are wireless ad hoc networks, whereas the structured networks have a backbone of static nodes, often mains powered. We will only consider structured networks in this thesis.

Main Characteristics

In this section we list the most important characteristics of WSNs.

- Low energy consumption is one of the most important design constraints in WSNs.
- Sensor nodes often operate in harsh and inaccessible environments with very unreliable links [5].
- Continuous data transmissions are rare in WSN applications; small packet bursts occurring at a certain (often very low) frequency is the most common transmission pattern. This leads to low requirements on QoS (although real-time requirements might exist).
- Self-management capabilities are needed, e.g. adapt to failures and changes in the environment. Self-management also assists in deployment.
- The large scale networks makes it infeasible to rely on centralized algorithms (e.g. executed on the Gateway (GW)), therefore a decentralized management is important.

2.2.3 Internet of Things

The IoT is a novel paradigm rapidly growing in the modern wireless communication area. There is not any standard definition of the IoT—there exist several visions of it [12]—but a wide and general vision is to extend the Internet into the real world embracing everyday objects. It is often possible for these units to communicate with humans over the Internet but the major strength is the capability of communicating with each other to reach common goals.

In this section, we will focus on the IoT from a WSN perspective. WSNs arguably play a very important role as a building block in the creation of the IoT [12]. WSNs often has a GW connected to for example a local server or other networks, allowing them to disseminate their sensor data to remote processing, visualization, analysis, and storage systems. Taking this a step further, by connecting the GW to the Internet, creates (at least partly) the IoT paradigm.

There exists a wide range of applications for the IoT, ranging from simple monitor and control systems to advanced systems where a combination of different WSNs are cooperating with each other in order to perform complex tasks in an intelligent manner. Connected WSNs create an even wider range of applications than the WSNs themselves. These applications can be grouped into four domains [12]:

- Transportation and logistics domain
- Healthcare domain
- Smart environment (home, office, plant) domain
- Personal and social domain

One of the major problems with the IoT is the lack of standardized protocols and architectures, just as for WSNs. This problem is even more severe in the IoT, due to the vision of getting all kinds of systems to communicate and cooperate. However, several contributions for standardization exist [12]. For example, reference models and example architectures have been proposed for the IoT (e.g., [13]).

Communication in WSNs can, like most communication systems, be divided into several layers. Table 2.1 shows an example OSI model protocol stack for WSNs used in the IoT. Note that this protocol stack is not strictly enforced in any way: there exist several other possible protocol stacks for WSNs and the IoT. Also note that there are no standardized layer boundaries; researchers have produced monolithic solutions that cut across the boundaries of different layers. The protocols are often specialized to meet the requirements of the selected application. Roughly said, WSNs defines the lower layers (layer 1 and 2) whereas the IoT defines the higher layers (layer 3+).

Table 2.1 An example protocol stack.

Layer	Protocol
5: Application	CoAP (Constrained Application Protocol)
4: Transport	UDP (User Datagram Protocol)
3: Network	6LoWPAN
2: Data Link	IEEE 802.15.4
1: Physical	IEEE 802.15.4

IPv6

The latest Internet Protocol (IP) version, IPv6, is a really important advance in computer networks, enabling the evolution of the IoT. With the previous version, IPv4, it is only possible to address 4.3 billion devices. This clearly sets a limit for the IoT and the vision of 50 billion connected devices by 2020. The latest IP version provides an almost unimaginable number of IP-addresses ($3.4 \cdot 10^{38}$).

2.3 Related Work

As we have seen; there are neither general standardized architectures nor protocols for WSNs. Moreover, most WSN protocols (e.g. ZigBee) just support joining/leaving of nodes: handover (by our definition) is not supported. The mobility problem has been well studied in other communication systems but not in WSNs [14].

There exist two main approaches for enabling mobility in WSNs. The first one is to handle it at the network layer. Examples include MIPv6 [15], HMIPv6 [16], FMIPv6 [17] and PMIPv6 [18]. These schemes can help a MN to maintain an ongoing connection with a unit that is not a part of its WSN [19]. The problem with these IP solutions is that handover related messages are sent via IP and this is not feasible in WSNs due to the large overhead of IP packets. Some sort of middleware is required, transforming IP packets to a format more suitable for WSNs. These protocols also often require extensive communication with the MN, hence making them energy inefficient. Further, they seldom define how the actual low-level handover should be done.

The second approach is to incorporate mobility awareness in the Medium Access Control (MAC) protocol, hence making it responsible for detecting and handling moving nodes. Examples include MMAC [20], MA-MAC [21] and MS-MAC [22]. However, most of these protocols seem to have forgotten the fundamental properties of WSNs, such as unstable links and stringent requirements on low energy consumption. Namely, many of them trade off low energy consumption for QoS parameters such as latency and throughput and some assume good quality links with gradual and steady process deterioration [14]. Moreover, there is no real proof of their applicability to a demanding environment, nor any tested solutions that support moving objects in a critical environment [23]. Further, the network layer switch has to be added on-top of these MAC protocols.

Most solutions in both of these approaches focus on QoS aspects such as reliability and supporting real-time applications. Moreover, neither of these approaches has any particular consideration on energy efficiency in a handover situation. A general problem in the area of WSNs is the gap between research and real application. Many protocols and solutions have been proposed, but very few of them have actually been implemented and evaluated in a realistic scenario.

As our purpose for this thesis is to design a generic algorithm for WSNs, we will not focus on incorporating handover in existing protocols nor come up with a novel protocol. Instead, we want to design an algorithm which is adaptable for any WSN, no matter which protocols it uses. The most relevant works to us are:

- Smart-HOP [24]: empirically evaluates a handover algorithm and fine tunes parameters in order to optimize the algorithm.

- Fotouhi et al. [4]: describes a handover algorithm in detail (focusing on reliability). A fuzzy logic approach for evaluating several parameters in order to make good handover decisions is proposed.
- LoWMob [19]: proposes a network controlled handover algorithm for 6LoWPAN.

Additional relevant works are Inter-MARIO [25] and S-GinMob [23] which propose handover algorithms for WSNs with focus on increasing the QoS.

Algorithm Design

IN THIS chapter we elaborate on the main characteristics and design objectives of handover algorithms for WSNs. Our main focus is on low power. We begin with a comparison of traditional communication systems and WSNs by looking at their different requirements on handover. This is followed by the most important considerations when designing a handover algorithm—from a WSN perspective. A thorough analysis of the different parts of a handover algorithm is presented followed by an examination of handover's relation to the OSI model protocol stack.

3.1 Traditional Communication Systems: A Comparison

In general, the main focus of handover—in traditional communication systems—has always been on QoS aspects such as latency, reliability, throughput and fairness. The switch of AP should be as seamless as possible and a MN should have an upper limit for its disconnected time, thus supporting real-time applications. This allows for continuous data transmissions without interruptions and jitter, enabling multimedia applications such as voice and video. These handover requirements do for example apply for the two systems with the most extensive research in the handover area: cellular and Wi-Fi. Energy consumption is not a critical parameter in these systems; their protocols have high energy overheads.

The unique characteristics of WSNs (see Section 2.2.2) make techniques used in the traditional systems unsuitable. Contrary to traditional communication systems such as cellular and Wi-Fi, WSNs face other challenges due to their severely constrained resources. The simple low-power radio transceivers lead to highly variable wireless links, which requires a more careful evaluation of the handover parameters. Low energy consumption is of uttermost importance for WSNs. However, data rates are low in WSNs and the QoS required by WSN applications is often much lower than for applications that support streaming audio and video. Fairness is not an issue since all units work together to achieve common goals.

The centralized approach used in cellular networks, with the mobile switching center as controller, should be avoided in large scale WSNs since it will scale poorly. A decentralized

approach is more suitable. Wi-Fi is not built with mobility in mind; handover support is more like an extension to allow users to move in the network. On the contrary, movement is a part of many WSN applications. Communication between APs in Wi-Fi is limited; the handover is mobile controlled. Involving the APs in the handover, i.e. the network, allows for smarter handovers and lower energy consumption on the MNs since they get assistance.

To conclude: QoS parameters such as throughput, high data rates and fairness are the most important ones for traditional communication systems. These parameters are only of secondary importance for most WSNs. On the other hand, low energy consumption is the most important parameter for WSNs and this is of minor importance for the traditional communication systems. Hence, protocols and techniques used in the traditional communication systems are not viable for WSNs.

3.2 Considerations

There are many things to consider when designing a handover algorithm. In this section we address the most important ones from a WSN perspective. The only assumption made on the WSN (except for the known characteristics stated in Section 2.2.2) is mains powered APs. Hence, energy is not a concern for the APs; they are assumed to have unlimited power.

3.2.1 Energy

In order to minimize the use of a certain resource, a good starting point is (as in all kinds of optimizations) to find out which part of the system consumes most of the resource in question. When this is known, the effort should be in reducing this part's consumption of the resource. The subsystems of a sensor node (see Section 2.2.1) affected by the handover are mainly the processing and communication subsystems. Comparing these two subsystems in terms of energy consumption is not a trivial task, but one possible way is to look at the ratio of the energy spent on communicating 1 bit to the energy spent on executing 1 instruction. Example values for this ratio are in the range of around 1000 to 2500 [10], hence for the bits sent over the radio the processor is able to execute several orders of magnitude more instructions. Thus, the effort should be put in reducing the communication.

Communication Subsystem: From the Energy Point of View

The power-performance characteristics of microcontrollers and microprocessors have been studied extensively. Most of them have a power management unit with tasks like shutting down idle components and dynamic voltage scaling. However, this is not the case for the transceivers. The small packets and low data rates require low-duty cycle radio electronics. However, it is technically challenging to design such circuits to be energy efficient [26].

There are several important things to note about the transceiver. To begin with, intuition might tell that transmit is much more energy consuming than receive. But, in fact, many of today's transceivers consume almost equal amount of energy in transmit mode as in receive mode. A specific example is the Atmel AT86RF233 [27] transceiver which

consumes 11.8 mA in receive mode and 13.8 mA in transmit mode. Moreover, this is the case for the maximum transmit power; in closer range less power may be used. This requires some rethinking when designing protocols for WSNs as the conventional network protocols often assume receive power to be negligible.

Another important thing to note is that (on many nodes) operating in idle mode results in significantly high power consumption, almost equal to the power consumed in the receive mode [28]. Thus, it is important to completely turn off the transceiver—rather than using the idle mode—when no data are transmitted or received as operating the transceiver in idle mode does not provide any advantage in terms of power.

A third thing to notice is that a significant amount of power dissipation is caused by the changes in the transceiver's operating modes [10]. For example, when switching from sleep to transmit mode the transceiver circuit consumes a large amount of power just for starting up the transmitter.

To conclude: in order to save energy the communication subsystem should be turned off as much as possible, the other states do not provide any advantages in terms of power consumption. Thus, it is of great importance to communicate as little as possible (on the nodes where energy consumption is critical) when doing a handover. If possible, the load should be put on the processing subsystem rather than the communication subsystem.

3.2.2 Type of Handover

In Section 2.1 we gave a brief overview of handover and divided it into different categories. In this section we look at what impact these categories have on handover in WSNs.

Mobile vs. Network Controlled

The key to saving energy on the MN in a handover situation is to put as much responsibility as possible on the network, thus doing a network controlled handover. If the network is able to handle the entire handover it will require almost zero energy from the MN. A network based handover algorithm is especially important in low data traffic networks (the majority of WSNs) since the energy spent on handover compared to ordinary data traffic will be high.

In networks with high data rates, e.g. networks with much sensor or position data to be transmitted, an inefficient handover will not be as noticeable. Moreover, in high data traffic networks the APs will already be loaded with routing network traffic, adding handover responsibilities to the APs will make the load on them even higher. If these conditions hold, a mobile controlled handover might be the better choice.

Intra-domain vs. Inter-domain

It is often desirable to have different strategies for intra-domain and inter-domain handovers. It is likely that more knowledge is available in an intra-domain handover than in an inter-domain handover, e.g. position knowledge. A good approach may be to start by trying to do an intra-domain handover and change to an inter-domain handover if it fails.

Vertical vs. Horizontal

Vertical handovers are not likely in WSNs; the sensor nodes rarely have support for more than one link layer technology for accessing a network.

Hard vs. Soft

The simple transceivers the sensor nodes possess are often not able to operate on several channels at the same time, thus making soft handovers impossible. Another thing to consider in this context is how to handle transmissions in progress. Since WSN applications are often tolerant to jitter a simple way to handle it might be to abort the transmission and resume it through the new AP when the handover is done, by buffering the data at the previous AP. Throwing away the packets and consider them as lost might also be viable thanks to the often low QoS requirements. Other alternatives include completing the data transmission before the handover or negotiate for a dynamic rate adaptation (so that data can be transmitted at a higher speed before the link breaks) [14].

Forward vs. Backward

In WSNs forward handovers is—in most cases—the better choice. The unreliable and rapidly deteriorating links make a backward handover strategy very precarious. This is especially true for indoor applications where obstacles such as walls could cause an almost instant loss of connection. Moreover, Fotouhi et al. [24] show that handover should be made in the lower end (close to the disconnected region) of the transitional region (at least in sparse WSNs). Hence, we cannot trust the link to the current AP as it is just about to break.

3.2.3 Position Knowledge and Mobility

Position knowledge can be used for aiding the handover process, if available. For example, the selection of AP can be supported by the MN's location and moving direction in relation to candidate APs. The velocity—recall that velocity is both speed and direction—can be used to estimate the deterioration speed for a link. Localization algorithms is beyond the scope of the thesis, example of such are triangulation, angle of arrival and signal strength indicator.

Position knowledge can be used to decide mobility patterns in the network which in turn can be a tremendous advantage when doing a handover. We will use the remainder of this section to look at two mobility aspects: degree and patterns.

Degree

A lot of movements in the network might result in many concurrent handovers. If this is the case the handover should be a simple process, especially if it is a network controlled handover. The speed of the MNs is also important to consider; if they move fast and the handover is a lengthy process the current handover might not be done before it is time to initiate a new one.

Patterns

A mobility model is a formal mathematical description generalizing the characteristics of mobility patterns. Knowing how and when a MN moves in the network allows for intelligent handover algorithms. Depending on the degree of predictability in the mobility pattern, it is more or less easy to implement artificial intelligence algorithms such as learning algorithms and path searches. However, due to the MN's limited resources (e.g. processing power and memory) these kinds of algorithms seldom are feasible to implement. A thorough examination of mobility models and patterns is out of scope of the thesis.

3.2.4 Quality of Service

The QoS requirements of WSNs have already been discussed in various contexts throughout the thesis. As we have seen; the requirements are often very low. However, it is important to know the application's requirements on QoS. If these (contrary to expectations) are high, great care must be taken when designing the algorithm. If a MN is required to be reachable at all times, i.e. always connected to the network, some sort of monitoring must be used. If an application for streaming continuous data (e.g. audio and/or video) is present the handover might have to be fast and seamless as well.

3.2.5 Scalability

In order to support several thousands of devices the cells have to be small, i.e. have a large amount of APs. This is similar to the usage of micro- and picocells in cellular networks for supporting more devices and higher data rates. However, a network with dense deployment of APs (smaller cells) will require more frequent handovers, leading to higher energy consumption on the MNs. On the other hand, few APs (relative to the number of MNs) will make the communication channels congested. This could have a negative effect on the MN's energy consumption depending on the MAC protocol used. For example, a Carrier Sense Multiple Access (CSMA) protocol may require the MN to listen on the channel several times just to discover that it is busy.

The use of a centralized unit for managing the handovers, like the mobile switching centre in cellular networks, should be avoided since it will be a limiting factor for the scalability. Another thing worth considering is the differences between a network and a mobile controlled solution. If the number of MNs grows in the network without increasing the density of APs (i.e. making the cells smaller) a network controlled handover will scale poorly compared to a mobile controlled handover. However, in terms of energy, the network controlled solution might be the better choice since it reduces the MN's communication and thereby avoids the phenomena of congested channels described in the previous paragraph.

3.2.6 Redundant Handovers

A redundant handover is an unnecessary handover and should be avoided since they waste resources. We define it as follows¹.

Redundant handover. A MN η executes a handover from AP λ , to AP ζ . If η then, *within* a short time frame ϵ , executes another handover to AP μ , the handover from λ to ζ was a *redundant handover*. $\square$

The Ping-pong Effect

The ping-pong effect is a special case of redundant handover where $\mu = \lambda$ and several consecutive redundant handovers take place at a rapid pace. The reason for this might be due to sudden fluctuation in link qualities [24] (which is especially common in WSNs). This usually happens when the MN is inside the coverage of more than one AP but in the frontiers of these. The density of APs is an influencing factor; dense deployments have abundant good links to choose from whereas sparse deployments often lead to selection of bad links (in absence of better). Thus, the ping-pong effect is more severe in sparse networks. A properly designed handover algorithm can avoid the ping-pong effect (and redundant handovers in general) by taking care when doing the initiation and selection decisions (more details on this will be given in Section 3.3).

3.3 The Parts of a Handover Algorithm

This section intends to give guidance in answering the three questions regarding handover given in Section 1.2:

- When should the handover be initiated?
- How is the new access point chosen?
- How is the actual switch done?

In order to do this, we divide handover into three sub-processes (as in Section 2.1), each responsible for answering the corresponding question. Further, these three parts are divided into two parts each, thus yielding six parts:

1. Monitor
 - (a) Observe
 - (b) Initiate
2. Selection
 - (a) Scan
 - (b) Select
3. Switch
 - (a) Data link layer switch
 - (b) Network layer switch

¹Similar to how the Swedish Meteorological and Hydrological Institute defines seasons; it is only possible to determine a change of season some days after its occurrence.

3.3.1 Observe

To be able to answer the first question, information about the state of the connection must be gathered. Two questions come to mind. Which parameters to monitor? How to monitor these parameters? This section intends to answer these questions.

Parameters

Link Quality (LQ) parameters such as RSSI, SNR and BER are the most valuable when making a handover decision since they describe the quality of the connection. They are also the most commonly used parameters [4]. However, due to the harsh environments and unreliable links in WSN, the LQ parameters are unstable. Thus, it is important to make the parameter readings reliable; great care should be taken when designing the mechanism for estimating the LQ. A sampling window can be used for stabilizing a parameter. Using a window of for example five samples and calculating the average of these will give a more reliable and stable reading than each sample considered separately. However, deciding the optimal window size is not a trivial problem. If the window is too large it will provide outdated values and if it is too small it might provide unstable readings. Moreover, relying on only one (unreliable) parameter (e.g. considering the RSSI criterion individually) may lead to inappropriate handover decisions [4]. Ideally, the LQ value is based on several LQ parameters.

In order to make a good handover decision, other parameters (in addition to the LQ) may be used, e.g. velocity, transmit power and the ratio of the used capacity to the total capacity for the APs. Deciding which parameters to use is not an easy task. LQ should always be used but other than that it depends on what the application provides. Adding a parameter to be used only for the handover may be a bad idea; an energy hungry positioning only used for deciding whether to handover or not will probably cost much more than it saves (when avoiding a redundant handover or two).

Observation Types

In order to monitor the selected parameters, we consider two observation strategies:

1. Discrete observation: passive monitoring, parameter values are gathered only at events, e.g. receiving data or calculation of acceleration (initiated by the application).
2. Continuous observation: active monitoring, parameter values are frequently observed (no matter what the application does).

The discrete observation strategy leaves no guarantee for connection since it is not possible to know whether the MN is connected or not in the absence of events. It uses very little additional energy since it piggybacks on the application's tasks, no extra actions are performed.

The continuous strategy uses more energy but guarantees to notice a disconnected MN. This can be made adaptive, e.g. if the MN is close to the AP and not moving the monitor frequency can be lowered. How much energy it uses depends on the observation frequency and how much energy a sample requires. The link layer MAC protocol and application will have a great impact on this. It is of course possible to combine a continuous with a discrete observation (e.g. by only using the continuous observation in the absence of events).

Another thing to consider is whether the MN or AP should have the task to monitor. Letting the AP do all observation would be optimal since this would require zero energy consumption by the MN. This is however not viable since most of the parameters are measured on the MN and would require it to communicate the data to the AP. Communication is, as we saw in Section 3.2.1, the most energy consuming part of a MN.

3.3.2 Initiate

The task of the Initiate sub-process is to analyze the parameters gathered by the Observe sub-process and decide whether a handover should be initiated or not. From an energy point of view, the analysis of the parameters should not require any comprehensive calculations since these might occur frequently, depending on the observation strategy used (see Section 3.3.1). However, imprecise calculations can lead to a bad decision which in turn can cause a redundant handover.

Strategies

There exist two types of handover decision models [4]:

1. Standard models: consider only one parameter by comparing it with a threshold value.
2. Heuristic models: the strategies that do not belong to the standard models.

The standard models should be avoided since they can lead to bad handover decisions if the parameter is unstable and/or unreliable. Instead, one of the heuristic models should be used. This category includes a whole range of different strategies, e.g. dynamic programming, pattern recognition, and artificial intelligence. Often, several parameters are involved and considered together. However, many of the heuristic models are too heavy to be feasible for the simple MNs. Fotouhi et al. [4] argue that fuzzy logic is the best one for WSNs. An example is F-LQE [29] which takes several LQ parameters into account by a fuzzy logic approach.

Ultimately, most strategies in both models compare parameters with threshold values. Finding the optimal thresholds, especially when several parameters are considered together, is an arduous process requiring extensive empirical evaluations. Fotouhi et al. [24] do an empirical evaluation of a few LQ related parameters, showing that a handover in the lower end of the transitional region (close to the disconnected region) is better than one in the connected region. This might be true for a network with sparse deployment of APs and minimal overlap between them. This is however not necessarily true in a network with dense deployment where the MN is guaranteed to have a good LQ with at least one AP. In such network a handover high up in the transitional region might in fact be better, since it avoids unnecessary unreliable communication in the transitional region.

A handover may of course be initiated on the detection of a lost connection too, i.e. the MN is no longer able to communicate with its associated AP.

Initiation Responsibility

As we saw in Section 3.3.1; the MN is the one most suitable for observing the parameters and should primarily be the one who decides whether a handover should be initiated or not. However, if the AP is observing some parameters it too should be allowed to initiate a handover.

3.3.3 Scan

Once the Monitor part of the handover algorithm has decided that a handover should be initiated, it is time for the Selection part. In order to select an AP, candidate APs have to be found. This is the task of the Scan sub-process. The Scan part is both the most time and energy consuming in most of the existing handover algorithms. Scanning for several seconds is not uncommon. During a scan the radio is turned on and as we saw in Section 3.2.1 the communication subsystem is the most energy consuming one.

The most common approach is for the MN to scan all possible channels (e.g. frequency channels or time slots, depending on the MAC protocol) and listen for beacons. This is extremely time and energy consuming. Therefore, some optimizations should be used. An example is to reduce the numbers of channels to scan (e.g. by knowing all nearby channels). This approach is commonly suggested in Wi-Fi optimizations. Another optimization might be to reduce the time needed for scanning a channel (while still ensuring that all APs on that channel are found). This can for example be done by increasing the APs' beacon frequency. In fact, both of these optimizations actually increase the QoS (by reducing the handover delay) *and* reduce the energy consumption, thus improving the energy efficiency. In Chapter 4 we present a (to our knowledge) novel approach for the Scan sub-process.

3.3.4 Select

The task of the Select sub-process is to analyze the parameters of the candidate APs found by the Scan sub-process and select one of them. Most communication protocols (e.g. Wi-Fi and existing WSN protocols) do not specify how an AP is selected. We suggest using the same parameters and evaluation strategies as in the Initiate part of the algorithm, see Section 3.3.2. However, one difference is that this decision is expected to be made far more infrequently than the decision in the Initiate part. This allows for more advanced calculations.

The Initiate and Select parts are in fact related. It is important to consider the threshold for initiating a handover and the threshold for an AP to be selected, at the same time. Fotouhi et al. [24] show that the difference between them, known as the hysteresis margin (defined below), has a great impact on the number of redundant handovers. Fotouhi et al. [24] also argue that a large hysteresis margin should be used—due to the high variability of links in WSNs—in order to avoid redundant handovers.

Hysteresis margin.

$$LQ_{candidate_threshold} - LQ_{init_threshold}, \quad (3.1)$$

where $LQ_{candidate_threshold}$ is defined as the minimum LQ for becoming a candidate AP and $LQ_{initiation_threshold}$ is defined as the maximum LQ for initiating a handover. $\square$

3.3.5 Data Link Layer Switch

This sub-process defines the protocol used for the actual switch of AP. This part is tightly coupled to the MAC protocol in use. In fact, most of the MAC protocols for WSNs provide an association (and disassociation) procedure which take care of the data link layer switch. The association might for example set the correct frequency, store the new AP's address and give the MN a new address.

Fotouhi et al. [24] argue that WSNs should avoid explicit disconnections from the previous AP, due to the (possibly) rapidly deteriorative links. Avoiding them will also lead to reduced energy consumption as the messages sent during the disassociation are no longer needed.

3.3.6 Network Layer Switch

The Data link layer switch part of the handover algorithm is responsible for doing the actual switch of AP, thus retaining a connection to the network. The Network layer switch part is responsible for (if required) retaining a point-to-point connection, e.g. a connection over the Internet. Another task is to perform a re-route if necessary. Most (probably all) network protocols for WSNs have routing and re-routing procedures. However, care should be taken; some algorithms rebuild the entire network when performing a routing procedure. Because of the small network change a MN handover is, this is often unnecessary. Another thing to consider is how data packets—addressed to the MN—sent to the previous AP during a handover (hence not sent to the MN), should be handled. A possibility is to buffer the packets and then let the previous AP send them to the new AP, i.e. creating a tunnel. This is how protocols like HMIPv6 [16] and PMIPv6 [18] handle mobility.

For energy reasons, the traffic between the APs and the MN should be reduced as much as possible. All context data related to the MN possessed by the previous AP (and needed by the new AP) should be transmitted via the network instead of having the MN sending the same data to the network once more. Example of such data is authentication information.

3.4 Handover and the Protocol Stack

Handover is tightly coupled to the OSI model protocol stack; the importance of the stack cannot be over-emphasized. As mentioned in Section 2.1; handover is a cross-layer function centered on the data link and network layers, see Figure 2.1. In order to get even close to an optimal handover mechanism it has to be embedded as a part of the network protocol stack.

A handover can for example be more effective if support, such as dedicated handover messages (perhaps with higher priority), is provided by the protocols. Moreover, a handover might require the protocol to perform an “update”, e.g. re-schedule time slots and/or perform a re-route. The protocols' abilities to do these kinds of changes effectively (thus incorporate mobility awareness in the protocols) are of great importance in order to make the handover algorithm energy efficient. These kinds of optimizations are out of scope of the thesis since we intend to keep the handover algorithm generic; the protocols should not matter. Some work on mobility aware protocols has been made, see Section 2.3.

3.4.1 Data Link Layer Protocol

It is particularly the MAC sublayer of the data link layer that is interesting for a handover mechanism. The task of the MAC layer is to share the media between units ensuring that no two nodes are interfering with each other's transmissions and deal with the situation when they do. The MAC protocol's design has a great impact on the energy consumption since it has full control over the radio (the most energy consuming part of a sensor node).

Many of the existing MAC protocols only support limited mobility in the network; almost all protocols are able to deal with a slow change in a network's topology, e.g. a joining or failing node [14]. They do not have a predefined process for doing a switch of AP (thus do not fulfill our definition of handover).

3.4.2 Network Layer Protocol

The network layer protocol is often simple and light weight in an isolated WSN. However, in order to enable connection to the Internet (IoT, see Section 2.2.3) the protocol has to be compatible with the IP protocol. This is (in WSN standards) a very heavy weight protocol. In order to connect the data link layer protocol to the IP protocol an adaptation middleware layer has to be used. An example task is to transform the large IP packets into smaller messages more suitable for the data link layer protocol used by the WSN. Example of such middleware protocol is 6LoWPAN [30]. However, existing implementations of 6LoWPAN do not support mobility [15]. For more details on the network layer protocol's relation to handover, see Section 3.3.6.

The Proposed Algorithm

IN THIS chapter we present our proposed handover algorithm. Energy consumption is the main criterion and it is designed with support of the guidance and knowledge gained in Chapter 3. First, we state the assumptions made on the WSN the algorithm is designed for. Then a brief overview of the basic mechanics of the algorithm is given. This is done to make the rest of the chapter, which is dedicated to a detailed explanation of the algorithm's parts, easier to follow and understand. We conclude with a brief overview of alternative approaches.

4.1 Overview

This section gives an introduction to the proposed algorithm. In terms of classifications (see Section 2.1.1), it is best described as a hybrid of forward and backward handover: the handover is predicted using the existing link (in order to ensure connectivity and avoid retransmissions due to a bad connection) but is initiated and executed through the new AP (to be able to handle rapidly deteriorating links). A network controlled strategy is used for intra-domain handovers, whereas a mobile controlled strategy is used for inter-domain handovers.

4.1.1 Assumptions

These are the assumptions made on the WSN:

1. The network consists of static mains powered APs with MNs as end devices.
2. The network is divided in domains (at least one).
3. The MNs' require a very low energy consumption.
4. The APs know the channels used by its neighbor domains.
5. The requirements on QoS are low.
6. The MNs must ensure connection to the network, i.e. the connection has to be frequently checked.

7. Some kind of LQ parameter is available.
8. The network units have location knowledge; an AP knows its location and a MN knows its location as well as the associated AP's location. It is however possible to tweak the algorithm if location knowledge is not available.

Please note that many kinds of WSN systems meet these requirements; we are not restricted to a particular system. The proposed algorithm is supposed to be generic enough to be adaptable to many kinds of WSNs. One example is a WSN used for surveillance. Such system consists of a network of static mains powered APs with MNs as end devices. The system's main task is to monitor the location of the MNs. The users (e.g. people, machines or objects) wear sensor tags which are registered and monitored by a supervisor system. The system may for example be used in a mine, e.g. in order to ensure that no people are in the area of a rock blasting.

Some parameters and functions have to be defined by the implementation of the algorithm, a summary of these can be found in Appendix A. These functions are marked with an asterisk (*) in the algorithms which are about to be presented.

MAC Protocol

The algorithm requires the MN's MAC protocol to provide the following primitives¹:

1. `MAC_SCAN(channels)`: find available networks (returns a list containing the APs found).
2. `MAC_SEND(data, ID)`: send data.
3. `MAC_RECEIVE(ID)`: receive data.
4. `MAC_BROADCAST(data)`: send data to all network units in range (with energy consumption in the same order of magnitude as send and receive).
5. `MAC_ASSOCIATE(ID)`: interconnect a MN with an AP (and exchange necessary information about each other, e.g. ID and location).
6. `MAC_SYNCHRONIZE(ID)`: ensure that the MN is still able to communicate with its associated AP (returns the LQ).

These are very common primitives and should be provided by most MAC protocols (especially for WSNs), even the simplest ones. Associate and synchronize can be implemented with the send and receive primitives (how this is done is out of scope of the thesis, but a hint is that communication requires an implicit synchronization). Note that the algorithm does not require a particular medium access type, such as Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA).

Network Protocol

The algorithm requires the network protocol to provide the following primitives:

1. `NW_SEND(address)`: send data.
2. `NW_RECEIVE(address)`: receive data.
3. `NW_ROUTE(ID)`: update the route to a network unit.

¹The AP is only required to support the two most fundamental primitives: send and receive.

Figure 4.1 Messages sent in the proposed handover algorithm during an intra-domain handover. The MN performs a handover from AP1 to AP2 (AP4 does not send a *handover_notify* message since it is not good enough to become a candidate AP).

4.1.2 Algorithm Mechanics

In this section we give a brief overview of the algorithm. The rest of the chapter is dedicated to the detailed description of the algorithm's parts; such as handover initiation and selection decisions. Knowing the basic mechanics of the algorithm will make these easier to understand. The messages sent during a handover are shown in Figure 4.1 and their formats are shown in Figure 4.2. The proposed algorithm in flowchart form can be found in Appendix B. This is the algorithm in brief:

1. The MN monitors the connection. If it decides that a handover should be performed the MN initiates the handover by broadcasting a *handover_request* message to all APs in range (more details will be given in Section 4.2).
2. When an AP receives a *handover_request* message it decides whether it is a candidate AP or not (more details will be given in Section 4.3.2). If it concludes that it is, the AP sends a *handover_notify* message to the MN's AP. The *handover_request* message is ignored if the AP is associated with the MN.
3. When the MN's AP receives a *handover_notify* message it checks if this candidate AP is the best so far (more details will be given in Section 4.3.2). If it is, the candidate AP's information is stored.
4. When the MN's AP has received the *handover_notify* messages from all candidate APs (more details will be given in Section 4.6.1), it sends a *handover_select* message to the AP stored in the previous step.
5. When an AP receives a *handover_select* message it sends a *handover_execute* message to the MN. The *handover_select* message contains information about how to communicate directly with the MN.

<i>handover_request</i>	Associated AP's ID	MN's locations	MN's Δt	Stability data
<i>handover_notify</i>	MN's ID	Candidate AP		
<i>handover_select</i>	MN's ID			
<i>handover_execute</i>				

Figure 4.2 Formats of the messages used by the intra-domain handover algorithm. The first field in each message tells which type of message it is. The sender ID is an implicit field in all message formats (not shown in the figure). The *MN's locations* field contains two location points and the *MN's Δt* field contains the time difference between these two points. These two fields are used to calculate the relative velocity (see Section 4.2.1). The *Stability data* field (partly) decides the length of the message (more details will be given in Section 4.3.1). The *Candidate AP* field contains the candidate AP's potential and number of associated MNs (more details will be given in Section 4.3.2).

6. When a MN receives a *handover_execute* message it associates with the AP it received the message from.
7. If the MN does not receive a *handover_execute* message in a reasonable time (more details will be given in Section 4.6.2) after a handover has been initiated, it changes strategy to an inter-domain handover. In this strategy the MN scans (ordinary MAC protocol provided scan) the neighbor channels of the current domain (more details will be given in Section 4.4). If it still cannot find a suitable AP it scans all channels until one is found. When the MN finds an AP it associates with it.
8. When the association is done the newly associated AP calls the network protocol's routing procedure.

4.2 Link Monitoring

The algorithm intends to initiate a handover before the connection to the current AP breaks. This to save the energy wasted on re-sending packets when the connection is lost (or so bad that sending a package requires several tries). Further, due to assumption number 6 (see Section 4.1.1), a continuous observation strategy (see Section 3.3.1) is used for connection monitoring. How often an observation is done is decided by the *observe_frequency* parameter and the value of this is up to the implementation to decide (depending on the application's requirements on the QoS). The application is responsible for calling the observe procedure (at the frequency given by the *observe_frequency* parameter) when connected to an AP, unless a handover is already initiated.

4.2.1 Parameters

To select the parameters used for a handover initiation is not an easy task. As discussed in Section 3.3.1; relying on only one—perhaps unreliable—parameter might be a bad idea. Therefore, two parameters are taken into consideration when a handover decision is about to be made (the selection of parameters is inspired by Fotouhi et al. [4]):

- Link quality
- Relative velocity (optional)

Link Quality

The most valuable parameter for monitoring a connection is the LQ since it gives the current connection quality. The proposed algorithm does not specify which kind of LQ to be used. However, a recommendation is to use error rate parameters (e.g. BER), rather than signal strength parameters (e.g. RSSI). This recommendation is because signal strength parameters only measure the signal strength—no attempt to decode the signal is made—hence this signal could come from anywhere, even an interferer.

For further improvement, the LQ parameter is stabilized. This is done by using a sampling window for measuring the LQ, as discussed in Section 3.3.1. The number of observation samples (the window size) is defined by the *LQ_window_size* parameter. The purpose of the LQ window is to compensate for a temporary deterioration, e.g. due to an unreliable link or being behind an obstacle for a short moment. For deciding an appropriate window size, please see Section 3.3.1.

Relative Velocity

The second parameter used is the MN's velocity in relation to the associated AP; the relative velocity. This can be derived knowing the AP's and MN's location points (which are assumed to be known, see Section 4.1.1).

Relative velocity. Let capital letters denote (location) points, let $\vec{XY}$ denote a vector with initial point X and terminal point Y and let $||\vec{XY}||$ denote the length of vector $\vec{XY}$. Further, let B be the (location) point of MN η at time t_1 , let B' be the (location) point of η at time t_2 and let A be the (location) point of AP λ . Then, the *relative velocity* ($v_{\eta,\lambda}^-$) is defined as

$$v_{\eta,\lambda}^- = \frac{||\vec{AB'}|| - ||\vec{AB}||}{\Delta t},$$

where $\Delta t = t_2 - t_1$. □

Thus, the relative velocity (see Figure 4.3) is an average velocity between two time points for the MN's movement away from (positive sign on $v_{\eta,\lambda}^-$) or to (negative sign on $v_{\eta,\lambda}^-$) an AP. Moving around the AP in a circle yields zero relative velocity.

With knowledge about the relative velocity it is possible to determine how fast the LQ is expected to deteriorate. It also tells whether the MN is moving towards or away from the AP. The LQ in conjunction with the relative velocity will enable a good decision whether to handover or not. If location knowledge is not available, the algorithm uses only the LQ parameter. Algorithm 4.1 shows the Observe part of the algorithm (recall that asterisks mark functions provided by the implementation, see Appendix A).

Figure 4.3 The relative velocity measures how fast the length of vector $\mathbf{v}_r$ changes.

Algorithm 4.1 The MN's observe procedure.

```

1: procedure OBSERVE(void)
2:    $LQsample \leftarrow \text{MAC\_SYNCHRONIZE}(associated\_AP)$ 
3:    $LQ \leftarrow \text{ADD\_LQ\_TO\_WINDOW}(LQsample, self.LQwindow)^*$ 
4:   if Location data available then
5:      $relative\_velocity \leftarrow \text{CALC\_RELATIVE\_VELOCITY}(A, B, B', \Delta t)^*$ 
6:      $\text{INIT}(LQ, relative\_velocity)$  ▷ Defined in Algorithm 4.2
7:   else
8:      $\text{INIT}(LQ)$  ▷ Defined in Algorithm 4.2
9:   end if
10: end procedure

```

4.2.2 Initiation

The decision of initiating a handover is, as mentioned in the previous section, based on the LQ and the relative velocity (another reason for initiation a handover is of course a lost connection, i.e. being unable to communicate with the AP, e.g. when trying to send data or performing a locating procedure). The LQ and relative velocity are combined using a fuzzy logic approach.

Fuzzy Logic

Whether to initiate a handover or not is decided by fuzzy logic. The reason for this is discussed in Section 3.3.2. Table 4.1 shows the fuzzy logic matrix. Fotouhi et al. [24] suggest that a handover should be performed in the lower end of the transitional region (close to the disconnected region); therefore, we are very patient with the decision to

Figure 4.4 LQ fuzzy logic sets in relation to the connection regions.

Table 4.1 Handover decision fuzzy logic matrix.

LQ	Relative velocity	Handover
Low	–	Yes
Medium	Low and towards AP	No
Medium	Medium and towards AP	No
Medium	High and towards AP	No
Medium	Low and away from AP	No
Medium	Medium and away from AP	No
Medium	High and away from AP	Yes
High	–	No

handover. Converting the parameters' values to the fuzzy sets is up to the implementation. High, Medium and Low LQ are illustrated in relation to the connection regions in Figure 4.4. If only the LQ is available a handover is initiated if this value is mapped to the Low LQ fuzzy set; thus, in practice, just compared with a threshold value: $LQ_{init_threshold}$. The Initiate part of the algorithm is shown in Algorithm 4.2.

Algorithm 4.2 The MN's handover initiation procedure.

```

1: procedure INIT( $LQ$ , [ $relative\_velocity$ ])
2:    $handover \leftarrow$  FUZZY_EVAL_INIT( $LQ$ , [ $relative\_velocity$ ])*
3:   if  $handover = \text{Yes}$  then
4:      $new\_message \leftarrow$  CREATE_MESSAGE( $handover\_request$ )*
5:     MAC_BROADCAST( $new\_message$ )
6:     Start  $intra\_domain\_handover$  timer ▷ See Section 4.6.2 for details
7:   end if
8: end procedure

```

4.3 Intra-domain Access Point Selection

The proposed algorithm uses the following strategies when doing a handover (it proceeds with the subsequent strategy only if the current fails):

1. Intra-domain handover (network controlled)
2. Inter-domain handover (mobile controlled)
3. Reset and join network (mobile controlled)

We start with describing the intra-domain handover (AP selection).

4.3.1 Scanning

The core purpose of the proposed algorithm is to use as little energy as possible on the MNs. In order to do this, the costly scan procedure performed by most mechanisms for switching AP is avoided. Instead, a network controlled solution is used, which is—as we have seen throughout the analysis of the different parts of a handover algorithm—the key to saving energy. When a MN decides to initiate a handover it broadcasts a

handover_request message which is received by all nearby APs (in the current domain). The network now has knowledge about the LQ for all APs in range of the MN and will now make the decision of which AP the MN should handover to. The MN continues to operate as normal until it receives a message from the new AP (more details will be given in Section 4.3.2).

The length of the *handover_request* message (partly decided by the stability data field, see Figure 4.2) will have impact on the scan stability. A longer message will provide a more reliable LQ but will also take longer time to transmit and use more energy. Fotouhi et al. [24] show that if the handover is initiated in the lower end of the transitional region (close to the disconnected region) and an AP is allowed to become a candidate noticeably higher in the transitional region (hence having a wide hysteresis margin, see Equation (3.1)), a thorough observation of the stability will not provide a better handover than a more brief one. As we follow this advice (see Section 4.2.2 and Section 4.3.2), the stability data field can be held short, thus saving energy. The *request_stability* parameter defines the length of the stability data field in the *handover_request* message.

4.3.2 Selection

When doing an intra-domain handover, the selection of AP is divided into two parts:

1. Selection of candidate APs
2. Selection of new AP

Each AP, receiving a *handover_request* message, decides whether it is good enough to be a candidate AP or not (how this decision is made is described in further detail below). If an AP concludes that it is a candidate, it sends a *handover_notify* message to the MN's AP which then decides which of the candidate APs to select (also described in further detail below).

Candidate Access Points

When selecting whether an AP should be a candidate or not the same parameters as in the handover initiation are used (LQ and relative velocity), for the same reasons. The LQ is measured during the reception of the *handover_request* message. The AP calculates the MN's relative velocity by using the MN's locations and time difference between these locations (Δt) provided by the *handover_request* message (see Figure 4.2). An AP uses a fuzzy logic strategy (as described in Section 4.2.2) to decide whether it should be a candidate AP or not. The fuzzy logic matrix is shown in Table 4.2. The implementation is responsible for mapping the parameters' values to the fuzzy sets; the LQ should be mapped roughly as shown in Figure 4.4. The Medium set might be reduced by moving values in the left end into the Low set, thus to ensure a sufficiently large hysteresis margin.

An AP becomes a candidate if it has at least Medium potential. As the fuzzy logic matrix shows; we have high requirements on an AP to become a candidate. Since the handover is initiated in the lower end of the transitional region (see Section 4.2.2) the hysteresis margin becomes large, thus redundant handovers will be avoided. If location knowledge is not available, only the LQ is used to decide whether an AP is a candidate or not. If this is the case, an AP candidate potential is the same as the LQ fuzzy set value. If an AP decides it is a candidate it sends a *handover_notify* message (see Figure 4.2) to the

Table 4.2 Candidate AP decision fuzzy logic matrix.

LQ	Relative_velocity	Candidate AP potential
Low	–	Low
Medium	Low and towards AP	Low
Medium	Medium and towards AP	Medium
Medium	High and towards AP	Medium
Medium	Low and away from AP	Low
Medium	Medium and away from AP	Low
Medium	High and away from AP	Low
High	–	High

MN's AP containing, among other things, the candidate potential (i.e. Low, Medium or High) and its load, represented as the number of associated MNs (an implementation may change this to another parameter, e.g. traffic congestion).

New Access Point

When an AP receives a *handover_notify* message it checks the candidate's potential and if it is the best so far the new candidate AP is stored. If the current best one and the newly received one have the same candidate potential the load is used to distinguish the two. This enables load balancing in the network. When the AP has received the *handover_notify* messages from every candidate AP (more details will be given in Section 4.6.1) it sends a *handover_select* message containing information of how to communicate with the MN (i.e. the MN's ID) to the selected AP and disassociates with the MN, see Algorithm 4.3. This AP in turn sends a *handover_execute* message to the MN. Algorithm 4.4 shows how the AP handles the different message types used by the algorithm. Algorithm 4.5 shows ditto for the MN.

Algorithm 4.3 Procedure run on the AP for handover of a MN to a selected new AP.

```

1: procedure EXECUTE_HANDOVER( $MN_x$ )
2:    $selected\_AP \leftarrow MN_x.candidate$ 
3:   DISASSOCIATE( $MN_x$ )*
4:    $new\_message \leftarrow CREATE\_MESSAGE(handover\_select)^*$ 
5:   NW_SEND( $new\_message, selected\_AP$ )
6: end procedure

```

Algorithm 4.4 The procedure for handling received messages at the AP.

```
1: procedure HANDLE_RECEIVED_MESSAGE(message)
2:   switch message do
3:     case handover_request
4:       if Location data available then
5:          $B \leftarrow \text{message}.\text{(MN's locations (B))}$ 
6:          $B' \leftarrow \text{message}.\text{(MN's locations (B'))}$ 
7:          $\Delta t \leftarrow \text{message}.\text{(MN's } \Delta t)$ 
8:          $\text{relative\_velocity} \leftarrow \text{CALC\_RELATIVE\_VELOCITY}(A, B, B', \Delta t)^*$ 
9:          $\text{candidate\_potential} \leftarrow$ 
10:        FUZZY_EVAL_CANDIDATE( $\text{message.LQ}$ ,  $\text{relative\_velocity}$ )*
11:       else
12:          $\text{candidate\_potential} \leftarrow \text{FUZZY\_EVAL\_CANDIDATE}(\text{message.LQ})^*$ 
13:       end if
14:       if  $\text{candidate\_potential} = (\text{Medium or High})$  then
15:          $\text{new\_message} \leftarrow \text{CREATE\_MESSAGE}(\text{handover\_notify})^*$ 
16:         NW_SEND( $\text{new\_message}$ ,  $\text{message}.\text{(Associated Ap's ID)}$ )
17:       end if
18:     end case
19:     case handover_notify
20:        $MN_x \leftarrow \text{message}.\text{(MN's ID)}$ 
21:        $\text{candidateAP} \leftarrow \text{message}.\text{(Candidate AP)}$ 
22:       HANDLE_HANOVER( $MN_x$ ,  $\text{candidateAP}$ )*
23:     end case
24:     case handover_select
25:        $MN_x \leftarrow \text{message}.\text{(MN's ID)}$ 
26:        $\text{new\_message} \leftarrow \text{CREATE\_MESSAGE}(\text{handover\_execute})^*$ 
27:       MAC_SEND( $\text{new\_message}$ ,  $MN_x$ )
28:     end case
29:   end switch
30: end procedure
```

Algorithm 4.5 The procedure for handling received messages at the MN.

```
1: procedure HANDLE_RECEIVED_MESSAGE(message)
2:   switch message do
3:     case neighbor_channels ▷ Described in Section 4.4.1
4:        $\text{self.neighbor\_channels} \leftarrow \text{message}.\text{(Neighbor channels)}$ 
5:     end case
6:     case handover_execute
7:       MAC_ASSOCIATE( $\text{message.sender}$ )
8:     end case
9:   end switch
10: end procedure
```

4.4 Inter-domain Access Point Selection

If the intra-domain handover fails, e.g. due to no candidate AP in range, the algorithm switches to an inter-domain handover. This—unlike the intra-domain handover—is a mobile controlled handover. The reason for this is to avoid the complexity and load a network coordinated handover across domain boundaries would require. The inter-domain handover procedure is shown in Algorithm 4.6 and explained in detail in the following paragraphs.

Algorithm 4.6 The MN’s inter-domain handover procedure.

```

1: procedure INTER_DOMAIN_HANOVER(void)
2:    $APlist \leftarrow \text{MAC\_SCAN}(self.neighbor\_channels)$ 
3:   while True do
4:     while  $APlist.length = 0$  do
5:        $APlist \leftarrow \text{MAC\_SCAN}(\text{ALL\_CHANNELS})$ 
6:     end while
7:      $AP\_best \leftarrow \text{FIND\_BEST\_AP}(APlist)^*$ 
8:     if  $APbest.LQ > LQ_{candidate\_threshold}$  then
9:       break
10:    end if
11:     $APlist \leftarrow \text{MAC\_SCAN}(\text{ALL\_CHANNELS})$ 
12:  end while
13:   $\text{MAC\_ASSOCIATE}(APbest)$ 
14: end procedure

```

4.4.1 Scanning

The MN uses the scan procedure provided by the MAC protocol to scan for nearby networks. In order to reduce the energy, a concept of neighbor channels is used. This is very similar to the selective scanning in Wi-Fi through neighbor graphs suggested by Park et al. [31]. With neighbor channels, only the neighbors to the current domain’s channels are scanned. The channel of the current domain is also scanned; an intra-domain handover could fail due to failure in the associated AP. As previously mentioned, the Scan part is the most energy consuming process of the handover, reducing this with the help of neighbor channels will lead to a significant reduction of the energy consumption. A MN requests the neighbor channels of its associated AP every time it joins a new domain. This is done by setting a flag in the MAC association primitive (which may require a small change in the MAC protocol). An alternative might be to request the neighbor channels in a separate message, if the MAC protocol cannot be changed. The neighbor channels are sent in a *neighbor_channels* message; the format is shown in Figure 4.5 and is handled according to Algorithm 4.5. If even the inter-domain handover fails, the MN’s last resort is to reset itself and search for a network to join by scanning all channels, until it finds a suitable AP.

Figure 4.5 Neighbor channels message format.

4.4.2 Selection

In an inter-domain handover (or network join) the MN selects the AP with the highest LQ from the list of APs generated by the MAC protocol's scan primitive. For an AP to be select it must have a LQ larger than the $LQ_{candidate_threshold}$. If such AP cannot be found the MN performs another scan—even if APs with LQ values below this threshold were found. Connecting to a weak AP will probably just trigger a handover soon again and this puts load on the network and risk leading to redundant handovers (maybe even the ping-pong effect). Since the QoS requirements are low it is better to search until a good enough AP is found. If there is even a slightest risk for a MN to not be near any AP with a LQ above $LQ_{candidate_threshold}$, a sleep should be put between each consecutive scan to not exhaust the battery.

4.5 Switch

The process of the actual switch is pretty much the same for the intra-domain and the inter-domain handover. Context data should be transferred between the previous and the new APs instead of involving the MN (as described in Section 3.3.6).

4.5.1 Data Link Layer Switch

The data link layer switch is done by the MN associating with the chosen AP, using the MAC protocol's association primitive. Location information is exchanged as a part of the association.

4.5.2 Network Layer Switch

To do the network layer switch the newly selected AP simply calls the routing primitive provided by the network protocol. If the handover is an inter-domain the MN requests the AP's neighbor channels. The procedure for performing a network layer switch at an AP is shown in Algorithm 4.7 and is run after a successful association.

Algorithm 4.7 The AP's procedure for performing a network layer switch.

```

1: procedure NETWORK_LAYER_SWITCH( $MN_x$ )
2:   if  $MN_x$  required neighbor channels then
3:      $new\_message \leftarrow \text{CREATE\_MESSAGE}(neighbor\_channels)$ 
4:      $\text{MAC\_SEND}(new\_message, MN_x)$ 
5:   end if
6:    $\text{NW\_ROUTE}(MN_x)$ 
7: end procedure

```

4.6 Timers

Particularly two parameters related to timing are of interest: the *handover_select* time out and the *intra_domain_handover* time out, see Figure 4.6 (this figure illustrates the same scenario as Figure 4.1). Both of them will have a great impact on the handover delay (also illustrated in Figure 4.6), defined as follows.

Handover delay. The time elapsed from sending a *handover_request* message to association with a new AP. □

Figure 4.6 The timing for messages sent in the proposed handover algorithm during an intra-domain handover. The MN performs a handover from AP1 to AP2.

4.6.1 *handover_select* Time Out

An AP cannot know for sure (in a simple and efficient way) that it has gotten the *handover_notify* message from all candidate APs. Therefore, a timer is used, started at the reception of the first *handover_notify* message for a MN. When this timer times out the decision of which AP to handover to is made, regardless of whether the AP has received the *handover_notify* message from all candidate APs or not. A *handover_notify*

message received after the time out for a MN is simply ignored. When the *handover_select* timer times out, the procedure for handling the selected AP (Algorithm 4.3) is run.

The value of the timer is implementation dependent. One way to find a suitable value is to determine the average time for all parts of the algorithm, see Table 4.3 (every function depends on the network load l , or might depend on the network load $[l]$). The definitions of ρ , σ and τ are all dependent on the MAC or network protocol used. In most implementations α , β and γ will be negligible; it is the communication that takes the most time.

Table 4.3 The proposed algorithm's times.

Function	Description
$\rho([l])$	Time for sending a message between an AP and a MN.
$\sigma(l)$	Time for broadcasting a message from a MN to APs.
$\tau([l])$	Time for sending a message between two APs.
$\alpha(l)$	Time for handling a <i>handover_request</i> message.
$\beta(l)$	Time for handling a <i>handover_notify</i> message.
$\gamma(l)$	Time for handling a <i>handover_select</i> message.

The value of the *handover_select* time out can be calculated as follows

$$\text{handover_select}(l) = d(\tau([l]) + \beta(l)), \quad (4.1)$$

where d is the average number of candidate APs (an AP can only receive and handle one *handover_notify* message at a time²).

In order to get a static value, let l be e.g. 90 % of the (expected) maximum load on the network. Another possible way to arrive to a static value is by an empirical evaluation. If the QoS requirements allows for a large handover delay a large safety margin can be added to this value. If the requirements require a very low handover delay the value has to be analyzed more thoroughly in order to optimize it. Another solution is to use a dynamic value if the network load l is known (or at least approximated) when the system is running. Then Equation (4.1) can be used to derive the value of the *handover_select*.

4.6.2 *intra_domain_handover* Time Out

The MN must have a way of knowing if the intra-domain handover failed. This is the task of the *intra_domain_handover* timer. This timer is started when the MN broadcasts a *handover_request* message. If it does not receive a *handover_execute* message before this timer times out it switches to an inter-domain handover. When the *intra_domain_handover* timer times out, the inter-domain handover procedure (Algorithm 4.6) is run.

²It might however be able to handle several *handover_notify* messages at the same time if multiple processors are available (but it will still only be able to receive one message at a time).

The value of this timer is, just as for the previous timer, implementation dependent and can be decided in a similar manner. The equation is as follows

$$intra_domain_handover(l) = \sigma(l) + \alpha(l) + handover_select(l) + \tau([l]) + \rho([l]), \quad (4.2)$$

where all functions are described in Table 4.3 and *handover_select(l)* given by Equation (4.1).

4.7 Alternative Approaches

In this section we briefly look at some alternative algorithm solutions.

4.7.1 Stability

We stabilized the parameters used to make the initiation decision using a sampling window for the LQ in Section 4.2.1. This can be drawn further. In order to stabilize a single observation another parameter can be used; *number_of_syncs*, telling how many times the MN should synchronize with the associated AP each observation. The purpose for doing several synchronizations is to compensate for temporary interference. The average of the LQ values measured is stored as a sample in the sampling window. A large value on *number_of_syncs* will provide more stable readings but will use more energy; a compromise between stability and energy consumption has to be made. The time between each synchronization is defined by the *sync_frequency* parameter and its value depends on the environment (e.g. interference).

In Section 4.3.1 we talked about stabilizing the LQ used for deciding whether an AP should be a candidate or not by adjusting the length of the *handover_request* message. To make this even more stable, several *handover_request* messages could be sent and stored in a sampling window at each AP. This will obviously use more energy at the MN.

However, the stability provided by the algorithm should be more than enough to avoid most redundant handovers. Both of the approaches described above will provide a little more QoS but will (most likely) not be worth the extra energy consumption. Hence, the energy efficiency will be reduced. A redundant handover is not that severe since our proposed algorithm uses very little energy at the MN. It does however put some unnecessary load on the network.

4.7.2 Observation

If location data is available, the observation procedure can be made adaptive, i.e. by lowering the *observe_frequency* parameter if the MN is stationary.

As we discussed in Section 3.2.3; incorporating mobility patterns in a handover algorithm for WSN is rarely feasible due to the MNs' limitations. However, if the application has a very predictable pattern this can of course be worth considering. For the use case mentioned in Section 4.1.1 (surveillance) the pedestrian mobility pattern is the closest choice. This is a pattern manifested by its limited speed, obstacle avoidance, and chaotic nature [14]. Due to the unpredictability of this pattern it is not viable to be used for a handover.

4.7.3 Initiation

In the proposed algorithm the MN is the one that makes the handover initiation. A possible addition is to also make the AP able to initiate a handover, e.g. by sending a *handover_init* message to the MN. A reason for an AP initiated handover might be bad LQ in the direction from the MN to the AP (which, due to the harsh environments in WSNs, is not necessarily the same as the LQ in the opposite direction, i.e. asymmetric connectivity). Another reason might be as a load balancing procedure if the AP is congested. However, this is very precarious and could lead to severe ping-pong effects if incautious. Moreover, it will increase the complexity of the network.

4.7.4 Selection

The procedure of selecting candidate APs performed by the proposed algorithm can be made more efficient if intelligence is added to the network. One example is to only forward a *handover_notify* message if it is the best one for the MN in question, i.e. if an AP has forwarded a *handover_notify* message associated with a particular MN and then receives another one (associated with the same MN) with lower candidate potential, this new message is discarded. This will reduce the network traffic but increase the complexity of the network.

A brief comparison of some algorithms for selecting an AP (described below) is given in Table 4.4.

1. The proposed algorithm: the algorithm described in this chapter.
2. Standard: uses the MAC scan primitive (every time).
3. Location based: the network selects next AP based on the MN's location.
4. Broadcast: a MN broadcast every message, i.e. handover is not needed.

For most applications, the standard is not feasible due to the high energy consumption and the broadcast is not viable due to the high network load. The location based is not feasible due to the great risk of selecting a bad AP since the LQ is not considered.

Table 4.4 Comparison of selection mechanisms. Handover quality is defined as the probability of selecting a good AP.

Algorithm	Handover Quality	Network Load	Energy Consumption
The proposed	High	Medium	Low
Standard	High	Very low	Very high
Location based	Low	Medium	Very low
Broadcast	Very high	Very high	Very low

Implementation

THIS chapter describes an implementation of the proposed algorithm given in Chapter 4. We begin by describing the implementation of the WSN, including the hardware platforms and software used. An overview of the chosen protocol stack, defined by the IEEE 802.15.4 standard, is given. The chapter is concluded with a description of how the algorithm is implemented on the AP and MN. Implementation details can be found in the well commented source code.

5.1 Limitations

In order to implement the proposed algorithm we need a WSN. This is built with the hardware platforms described in Section 5.2 and software described in Section 5.3. Only the two lower levels of the OSI model protocol stack are used. This to keep the network complexity down; the interesting parts of the algorithm take place in the data link layer. A network layer is not implemented and consequently not the Network layer switch part of the algorithm. The network layer required functions are implemented using the corresponding functions provided by the MAC layer, e.g. `NW_SEND` uses `MAC_SEND` (the routing procedure is of course not implemented at all). Moreover, the relative velocity parameter is not considered when deciding handover initiation and candidate APs since implementing location knowledge in a WSN is far out of scope of the thesis.

5.2 Hardware

In this section we describe the hardware used for the MN and AP. It is possible to communicate with both the AP and MN via a PC using a serial interface, e.g. for sending commands and displaying logged data.

5.2.1 Mobile Node

In this section we describe the MN's hardware architecture (see Section 2.2.1). Figure 5.1 shows the MN's hardware platform.

Figure 5.1 A mobile node (with an attached battery).

Processing Subsystem

The MN is based on Atmel's AT91SAM3S ARM Cortex-M3 microcontroller [32].

Communication Subsystem

The communication subsystem uses Atmel's AT86RF233 2.4 GHz radio transceiver [27] and the Skyworks' SE2431L 2.4 GHz RF front end [33]. The transceiver is a very capable unit targeted for WSN applications (e.g. ZigBee and 6LoWPAN based applications), having special hardware support for the IEEE 802.15.4 standard protocols, including a MAC hardware accelerator. It has (with the front end) a range of around 100 meters indoor. It offers a programming method to specify the transmission power level during runtime and provides RSSI, energy detect and LQ indication for every received package. The LQ indication is implemented as the Packet Error Rate (PER) for the link.

Sensing Subsystem

The MN has a three axis digital accelerometer, a digital pressure sensor, a three axis digital output gyroscope and finally, a three axis digital magnetometer.

Power Subsystem

The MN is battery powered and uses AMS's AS1375 low power LDO (low-dropout regulator) [34].

5.2.2 Access Point

The AP's hardware platform is based on the same microcontroller and the same radio transceiver as the MN. However, a different front end is used: RFAxis' RFX2411 2.4 GHz RF front end [35]. Figure 5.2 shows the AP's hardware platform. The APs intercommunicates either via the radio or Ethernet. To enable Ethernet a hardware module has to be added to the AP, containing (among other things) an Ethernet controller and Atmel's AT91SAM9G20 microcontroller [36]. The AP communicates with the Ethernet module via a serial interface.

Figure 5.2 An access point.

A GW is an AP with an Ethernet module having an Ethernet connection to outside the domain. Figure 5.3 shows the GW's hardware platform; an AP with an added Ethernet module. The AP is mains powered.

Figure 5.3 A gateway.

5.3 Software

In this section we describe the software used by the MN and AP.

5.3.1 Operating System

Both the MN and AP run the Contiki operating system [37]. It is an open source operating system for the IoT. Contiki is designed for low power units with cheap and simple hardware. It has a rich network stack with support for full IP networking, supporting standard IP protocols such as UDP and TCP as well as 6LoWPAN and RPL [38]. It also provides a more lightweight stack: the rime stack. In addition to this, several MAC protocols are provided. However, only a minimal version of the Contiki operating system (more or less just the scheduler) and a different network stack is used for our WSN.

The Ethernet module (added to some of the APs) runs embedded Linux.

5.3.2 Network

The protocol stack is based on an implementation of the IEEE 802.15.4 standard (further details on this standard will be given in Section 5.4), provided by Atmel [39]. The protocol stack provided by Contiki is not used. This is because we only need the two lower layers of the OSI model protocol stack and the Atmel provided protocol stack works better with the transceiver since it utilizes the IEEE 802.15.4 hardware support.

5.4 IEEE 802.15.4

In order to implement a WSN we need an understanding of the protocol stack used. In this section we briefly describe the essential parts of the IEEE 802.15.4 standard [40]. The standard defines the MAC (which is a part of the data link layer) and physical layer of the OSI model protocol stack. It is targeted for applications with characteristics such as low throughput data and small packet sizes and provides only a very limited QoS. It is designed to be run on simple, low cost and low energy devices. In other words: it is designed for WSNs. Many of the popular IoT protocols are built on top of it, e.g. ZigBee [41] and 6LoWPAN.

5.4.1 Physical Layer

The IEEE 802.15.4 standard supports data rates up to 250 Kbit/s and operates on a subset of 27 available radio channels on specific unlicensed bands, depending on region. The bands are: 800 MHz (Europe), 900 MHz (USA) and 2.4 GHz (global). Features include energy detection within the current channel and LQ for received packets. The standard does not define what kind of LQ to use but it shall provide a value between 0x00 and 0xFF. The data are sent using frames.

5.4.2 MAC Layer

The main channel access method used is CSMA with collision avoidance. However, the standard supports a TDMA scheme in beacon mode through the superframes. We will not go into details on this, if interested please see [40]. Two kinds of transmission modes are possible: *a*) the transmitter sends directly to the receiver (which must have its radio on all the time) and *b*) the receiver polls the transmitter for pending data (hence can shut down the radio between polls). There are four frame types: data, acknowledgement, MAC command and beacon. All of the MAC protocol primitives required by the proposed algorithm (see Section 4.1.1) are provided.

5.4.3 Device Classes

The standard defines two classes of devices:

1. Full Function Device (FFD): having routing capabilities, is able to talk with any other device and supports any topology. It implements the complete protocol set.
2. Reduced Function Device (RFD): a simpler device than the FFD, having a reduce protocol set and is limited to leaves in more complex topologies. It can only operate as end device (see Section 5.4.4).

5.4.4 Network

All units in a network have an IEEE 64-bit globally unique address. A 16-bit short address may be used within a PAN. Each PAN uses a unique PAN ID (unique for the current frequency channel).

Device Types

Three kinds of device types are defined by the standard:

1. Coordinator: a FFD with network device functionality that provides services to the network, e.g. forwarding of packets.
2. PAN coordinator: a FFD acting as the principal controller of the PAN. A network has exactly one PAN coordinator.
3. End device: a RFD which is only able to talk to a coordinator.

Topologies

The standard defines three kinds of topologies:

1. Star: a simple structured topology having a PAN coordinator as the central node which all other network devices communicates with (similar to Bluetooth). The nodes only communicate with the central node; there exists no other communication between the nodes. A leaf may be a FFD or a RFD.
2. Peer-to-peer: an extension of the star topology in which the devices are not limited to communication with the central PAN coordinator only; additional point-to-point communication is also allowed. This network structure is essential in supporting ad-hoc networks as well as for supporting networks covering larger geographic areas.

3. Cluster tree: a special case of a peer-to-peer topology consisting of interconnected star networks (note that the network still has exactly one PAN coordinator).

5.4.5 Network Management

A PAN shall be started by an FFD only after an active channel or energy detect channel scan has been performed. A channel based on the scan results is then picked as well as a PAN ID for the network (unique for the selected channel).

Before exchanging information with other devices of a PAN a device must be associated. To do this, a device that intends to join an existing PAN performs a scan and selects one of the coordinators found (the standard does not define how a coordinator is selected). The node then sends a request to associate with the chosen coordinator. A PAN coordinator may indicate its presence by transmitting beacon frames. If it does not, an active scan has to be performed in which a device requests a beacon from a coordinator.

For PANs supporting synchronization it is possible for an associated device to request a synchronization using the synchronization MAC primitive. The IEEE 802.15.4 standard has no support for handover or mobility whatsoever. It is entirely up to the application to handle. It may detect a lost connection either by the synchronize procedure or by not being able to send data. If this happens, the device may perform an orphan scan, trying to obtain synchronization with its associated AP again. If this fails the device has to perform a complete reconnection to a network by again performing a scan. However, this is up to the application; it is not an implicit behavior upon a lost connection.

5.4.6 Optimizations

As previously mentioned: we want to keep the implementation of the algorithm as general as possible and will not optimize the protocols used to make them perform better in a handover situation. However, if the handover is about to be implemented in a “real” system, further attention must be placed on how to best optimize low-level protocol details and parameter tuning for each particular handover protocol and environment. Several optimizations have been proposed for the IEEE 802.15.4 standard, which could improve the handover performance, including the works by Staehle [42] and Garcia-Sanchez et al. [43].

5.5 Network Overview

As mentioned above; the WSN is built using the Contiki operating system in conjunction with the Atmel implementation of the IEEE 802.15.4 standard. The coordinators defined by the IEEE 802.15.4 standard (see Section 5.4.4) acts as APs, the end devices acts as MNs and finally, PAN coordinators acts as GWs. The cluster tree network topology is used. A PAN corresponds to a domain. We will in the remainder of this section briefly describe how the WSN is created.

An AP creates a Contiki process which initiates the hardware and the MAC layer. The process then calls a procedure which resets the MAC layer. When the MAC layer has been reset, its settings are changed; the radio is set to be always on, associations are allowed and the beacon is configured (a beacon enabled network is used).

When the MAC layer has been configured, the AP follows the procedure described in Section 5.4.5 to either start a new network or join an existing network. The first thing done is a scan. If the AP is configured to be a GW it will use the scan result to select a channel and a PAN ID unique for this channel. It then sets its short address and starts the PAN network. On the other hand, if it is configured to be an ordinary AP, it will use the scan result to select an existing PAN to join. It sends an association request and when this is accepted it begins doing its PAN tasks, e.g. sending out beacons. When an AP (or GW) receives an association request it generates a short address and sends an association response.

Both Contiki and the IEEE 802.15.4 standard are highly event driven; all MAC layer invocations are done by sending a request of some sort. When it has been handled a callback function is called. Callback functions are also used to handle external events such as a received message or association request. The main loop of the AP is shown in Algorithm 5.1.

Algorithm 5.1 The AP's and MN's main loop.

```

1: while True do
2:   PROCESS_WAIT_EVENT();           ▷ Wait for an event (Contiki)
3:   wpan_task();                     ▷ Handle MAC-layer events
4:   app_task();                      ▷ Perform the application tasks
5:   process_post(PROCESS_CURRENT(), PROCESS_EVENT_MSG, NULL);
6:                                   ▷ Post an event to ourself (Contiki)
7: end while

```

5.6 Algorithm at Mobile Node

In this section we describe the implementation of the MN's part of the proposed algorithm. The parameter values will be discussed in Chapter 6.

5.6.1 Required Functions

In this section we describe the implementation of the required functions, see Appendix A. The required MAC functions are provided by the Atmel implementation of the IEEE 802.15.4 standard. Since we do not implement the location part of the algorithm the $\text{CALC_RELATIVE_VELOCITY}(A, B, B', dt)$ function is not needed.

The $\text{CREATE_MESSAGE}(\text{message_type})$ Function

This function is divided into several functions; one for each message type. These functions create an ordinary IEEE 802.15.4 data frame containing the messages data fields (see Figure 4.2) as load. In addition to the handover related messages two other messages are added: one for requesting the AP to print certain data of interest (e.g. channel and evaluation data) via the serial interface and one for sending ordinary sensor data to the AP.

Algorithm 5.2 Functions for adding a LQ value to the LQ window.

```
1: function ADD_LQ( $LQ$ )
2:    $LQ\_window[window\_ptr] \leftarrow LQ$ 
3:    $window\_ptr \leftarrow (window\_ptr + 1) \bmod LQ\_window\_size$ 
4:   if  $no\_of\_rec\_pkgs < LQ\_window\_size$  then
5:      $no\_of\_rec\_pkgs \leftarrow no\_of\_rec\_pkgs + 1$ 
6:   end if
7: end function
```

The ADD_LQ_TO_WINDOW(LQ , LQ_window) Function

We implement the LQ window as a circular buffer of size LQ_window_size . Two additional variables are used: one for keeping track of the next free place in the window ($window_ptr$) and one for keeping track of the number of entries in the window ($no_of_rec_pkgs$). The function is split into two functions: one for adding a LQ value to the window (Algorithm 5.2) and another for calculating the average LQ (Algorithm 5.3).

Algorithm 5.3 Function for calculating the average LQ.

```
1: function CALC_AVG_LQ(void)
2:   if  $no\_of\_rec\_pkgs = 0$  then
3:     return  $0xFF$ 
4:   end if
5:    $acc \leftarrow 0$ 
6:   for  $i \leftarrow 1$  to  $no\_of\_rec\_pkgs$  do
7:      $acc \leftarrow acc + LQ\_window[i - 1]$ 
8:   end for
9:   return  $\frac{acc}{no\_of\_rec\_pkgs}$ 
10: end function
```

The FUZZY_EVAL_INIT(LQ , $[relative_velocity]$) Function

Since we do not have location knowledge we just compare the LQ with a threshold value, see Algorithm 5.4.

Algorithm 5.4 Function that decides whether a handover should be initiated or not.

```
1: function FUZZY_EVAL_INIT( $LQ$ )
2:   if  $LQ < LQ_{init\_threshold}$  then
3:     return Yes
4:   else
5:     return No
6:   end if
7: end function
```

The FIND_BEST_AP(*AP_list*) Function

To find the best AP we use Algorithm 5.5.

Algorithm 5.5 Procedure that finds the best AP from a list of APs.

```

1: function FIND_BEST_AP(AP_list)
2:    $AP_{best}.LQ \leftarrow -1$  ▷ AP with the best LQ
3:   for all AP in AP_list do
4:     if  $AP.LQ > AP_{best}.LQ$  then ▷ Check if the AP is the best this far
5:        $AP_{best} \leftarrow AP$ 
6:     end if
7:   end for
8:   return  $AP_{best}$ 
9: end function

```

5.6.2 Command Line Interface

The only task the application of the MN has is to handle user requests received via the serial interface. The command line interface can be found in Appendix C.

5.6.3 Finite State Machine

The proposed handover algorithm is highly event driven (the events are received messages and time outs), as well as the Contiki operating system and the MAC layer protocol. In order to keep track of when and how the MN should react on certain events it implements a finite state machine, see Figure 5.4. Only the state transition defined by the finite state machine are allowed; e.g. if the MN receives a *handover_execute* message in the Synchronize state it is ignored. The finite state machine is implemented simply by using a variable containing the current state.

Initiation State

In the Initiation state the MN creates a Contiki process which starts by initiating the hardware and the MAC layer. It then calls a reset function which resets the MAC layer. When this is done the MAC layer is configured. The neighbor channels concept is implemented by having a variable, *neighbor_channels*, to keep track of the neighbor channels. This is set to all channels in the Initiation state (the reason for this is because the Scan state starts by scanning the neighbor channels). The MN uses the same main loop as the AP, see Algorithm 5.1. When the initiation is done the MN moves to the Scan state, (see Figure 5.4).

Scan State

The algorithm given in Algorithm 4.6 is implemented by the Scan state. When a suitable AP has been found and associated the MN moves to the Connected state (see Figure 5.4). The neighbor channels are requested by setting a flag in the association request message. Every time a MN associates with a new AP the LQ parameters are reset (i.e. the window).

Figure 5.4 Mobile node's finite state machine.

Connected State

More or less nothing is done in the Connected state. In a real application it would perform the application tasks (e.g. sending/receiving data or positioning) and synchronize with the AP at a frequency given by the *observe_frequency* parameter. The reason for the avoidance of frequent synchronization is because we want to be able to control exactly when the MN synchronizes with its AP. This to enable repeatable test cases when doing the empirical evaluation. A *neighbor_channel* message received is handled according to Algorithm 4.5. Two state transitions are possible in the Connected state (see Figure 5.4):

1. Synchronization is explicitly requested by the user: move to the synchronization state.
2. Unable to communicate with the AP (e.g. no longer able to send data): move to the Handover state.

Synchronize State

The Synchronize state implements the algorithm given in Algorithm 4.1 and consequently also Algorithm 4.2. Two state transitions are possible (see Figure 5.4):

1. Synchronization successful: move to the Connected state.
2. Synchronization failure: move to the Handover state.

Handover State

In the Handover state the MN does more or less nothing. In a real application it would perform the application tasks. Two state transitions are possible (see Figure 5.4):

1. *handover_execute* message received: handle the message according to Algorithm 4.5 and move to the Connected state.
2. *intra_domain_handover* time out: move to the Scan state.

5.7 Algorithm at Access Point

The initiation and network join/start is described in Section 5.5. The neighbor channels are hard coded in every AP (just because it is the simplest implementation). The AP keeps tracks of its associated MNs with an array of MN structures, containing information (such as short address) about each MN.

When a handover is initiated for a MN, i.e. the AP receives a *handover_notify* message and is associated with the MN, it creates a handover structure containing a pointer to the MN's structure, current time and information about the candidate AP (such as address and candidate potential). These handover structures are stored in an array for current handovers. As long as there is at least one MN involved in a handover this array is regularly checked for entries for which

$$\text{current time} - \text{stored time} > \text{handover_select time out}$$

is true. If such entry is found, the AP sends a *handover_select* message to the stored candidate AP and disassociates from the MN. Received messages are handled in accordance to Algorithm 4.4.

5.7.1 Required Functions

As for the MN; the required MAC primitives are provided by the Atmel implementation of the IEEE 802.15.4 standard. Further, since we do not implement the location part of the algorithm the $\text{CALC_RELATIVE_VELOCITY}(A, B, B', dt)$ function is not needed. The $\text{CREATE_MESSAGE}(\text{message_type})$ function is handled in the same manner as for the MN, see Section 5.6.1. In addition to the handover related messages two more are handled; sensor data messages and a display message (requesting the AP to send data to a PC via the serial interface).

The FUZZY_EVAL_CANDIDATE(LQ , [$relative_velocity$]) Function

Since we only use the LQ all this function does is to map the LQ parameter to the LQ fuzzy logic set, see Algorithm 5.6.

Algorithm 5.6 Function that calculates an AP's candidate potential.

```
1: function FUZZY_EVAL_CANDIDATE( $LQ$ )
2:   if  $LQ < LQ_{candidate\_medium\_threshold}$  then
3:     return Low
4:   else if  $LQ < LQ_{candidate\_high\_threshold}$  then
5:     return Medium
6:   else
7:     return High
8:   end if
9: end function
```

The HANDLE_HANOVER(MN , $candidate_AP$) Function

If a handover is already initiated for the MN (hence, the AP already has an entry in the handover array), the candidates are compared as described in Section 4.3.2. If the new AP is better than the current the records are updated. If the AP is not associated with the MN it simply does nothing.

The DISASSOCIATE(MN) Function

This function simply sets MN 's position in the association array to unused.

Empirical Evaluation

AN empirical evaluation of the proposed algorithm is described in this chapter. We begin by motivating the choice of evaluation method. This is followed by an explanation of how the source code is instrumented in order to measure certain metrics of interest. This instrumentation is used to decide the parameter values for the algorithm implemented in Chapter 5. Finally, the empirical evaluation is described, including results and an analysis of these.

6.1 Selected Evaluation Method

In order to evaluate the algorithm, there are mainly two strategies to choose from: *a)* empirical evaluation and *b)* simulation. As the chapter title reveals, an empirical evaluation strategy is the one we selected. A simulation definitely has its advantages; e.g. easy to repeat tests with exactly the same circumstances and automation of the testing. However, a realistic model of the transitional region is very important and Zuniga et al. [44] argue that common network simulation tools often can be very misleading in this region, due to the perfect-reception-within range model. Moreover, since the algorithm is implemented on top of the Atmel IEEE 802.15.4 implementation it would require extensive work to adapt it to a simulation tool. An alternative would be to implement the proposed algorithm in the Contiki protocol stack and use the Cooja simulator [45], but this would not require much less work. Thus, the two main reasons for choosing the empirical evaluation strategy is *a)* expectancy of results much closer to the reality than a simulation would provide and *b)* avoidance of algorithm adaptation to a simulation tool.

6.2 Instrumentation

In order to measure metrics of interest, the source code is instrumented. Timing metrics—e.g. handover delay and time for sending messages—are measured by storing the current time when such event is started (t_s) and then subtract the current time when the event is finished (t_e) from this stored time, i.e. $t_e - t_s$. Discrete event metrics—e.g. number of handovers and number of lost packets—are measured simply with a counter for each event.

The metric of interest that is the most difficult to measure is the energy consumption. Although power consumption is one of the (perhaps even the) most important concerns for WSNs, few protocols and mechanisms have been empirically studied in terms of their energy consumption [46]. Contiki is provided with powertrace [46], a software based energy measuring mechanism which tries to be more exact than just measure the number of transmissions. The basic idea behind powertrace is to measure the energy consumption by measuring the time the transceiver spends in its various states (e.g. transmitting, receiving and idle). One great drawback with powertrace is that even though it tries to be exact, it does not consider the receiver's state transition which—as we saw in Section 3.2.1—can cause a significant amount of power dissipation. It is also hard to estimate the power consumption if more intelligent hardware is used, e.g. hardware with automatic acknowledgement and automatic retransmission.

Since our purpose is to evaluate the proposed algorithm independent of the WSN it is run on, we will use an abstraction from the protocols and hardware in use. This is done by counting discrete events such as number of handovers and lost packets and derive the energy consumption from these. It will not provide any exact values but it should be good enough to get a rough picture of the energy consumption.

6.3 Algorithm Parameters

It is not our goal to optimize the proposed algorithm's parameters. Instead, we will reason about reasonable values for each and use these to create a reference algorithm to be used for the evaluation.

6.3.1 Timers

The values of the timers used by the algorithm are decided by an empirical evaluation where the average values for each function in Table 4.3 are decided. Since only one MN is used for the evaluation of the algorithm, l is constant and close to zero. The evaluation showed that the times for α , β and γ are negligible. Figure 6.1 shows the average times for σ , τ and ρ .

The *handover_select* Parameter

The value is decided by Equation (4.1). Since τ has a large standard deviation we use a large safety margin; giving it the value 50 (ms). The number of candidate APs is never expected to exceed four, hence $d = 4$.

$$handover_select = 4(50 + 0) = 200\text{ms}.$$

Figure 6.1 Empirical evaluation of message timings. The chart shows the average time and (sample) standard deviation for sending a specific message type. 100 handovers were performed.

The *intra_domain_handover* Parameter

The value is decided by Equation (4.2). As for the *handover_select* parameter we set $\tau = 50$ (ms). Also $\sigma = 25$ (ms) and $\rho = 25$ (ms) get large safety margins.

$$intra_domain_handover = 25 + 0 + 200 + 50 + 25 = 300\text{ms}.$$

The *observe_frequency* Parameter

This parameter is not used.

6.3.2 Thresholds

Luckily, Atmel has done an extensive evaluation of the LQ value for the AT86RF233 2.4 GHz radio transceiver, see Figure 6.2. Our evaluations confirmed these results; the only difference was that the PER values were displaced by 20 units to the left, i.e. the PER was slightly higher for a certain LQ value. The reason for this is probably multipath propagation.

The Atmel implementation of the IEEE 802.15.4 standard made some strange calculation when deriving the LQ value, taking the PER LQ value from the transceiver in conjunction with the signal strength. We changed this to use only the PER LQ value provided by the transceiver as the LQ value.

The $LQ_{init_threshold}$ Threshold

The algorithm suggests that the handover should be initiated in the lower end of the transitional region. Therefore, it is initiated at a PER above 40 %, thus

$$LQ_{init_threshold} = 160$$

(i.e. 140 in Figure 6.2 due to the displacement).

Figure 6.2 PER vs. LQ for the AT86RF233 2.4 GHz radio transceiver [27]. We observed an offset by 20 units to the left.

The $LQ_{candidate_threshold}$ Threshold

The algorithm suggests a large hysteresis margin. Therefore, a packet error rate below 5 % is required, thus

$$LQ_{candidate_threshold} = 200$$

(i.e. 180 in Figure 6.2 due to the displacement).

6.3.3 Stability

In this section we discuss the values on the stability parameters.

The *request_stability* Parameter

As discussed in Section 4.3.1; thanks to the large hysteresis margin this parameter is superfluous, thus

$$request_stability = 0.$$

The LQ_window_size Parameter

Fotouhi et al. [24] argue that a suitable value for these kind of parameters is three, thus

$$LQ_window_size = 3.$$

6.4 Performance Metrics

This section describes the empirical evaluation of the proposed algorithm. It is evaluated by measuring some performance metrics of interest:

- Handover delay
- Number of handovers and packet loss
- Energy consumption

For each metric we also do a theoretical comparison of the proposed handover algorithm and the following existing handover algorithms for WSNs: 1) Smart-HOP [24], 2) Fotouhi et al. [4], 3) LoWMob [19], 4) S-GinMob [23] and 5) Inter-MARIO [25]. There are very few proposed handover algorithms for WSNs; the reason for selecting those just mentioned is simply because they are the ones we found (and provide enough details about the algorithm to be able to do a comparison). A fair comparison is however difficult since very few of the existing handover algorithms for WSNs have been implemented and empirically evaluated.

6.4.1 Handover Delay

The handover delay was defined in Section 4.6. However, this is not a fair definition compared with other handover algorithms. In most contexts, the handover delay defines the time a MN is non-operational due to the handover. Our proposed algorithm allows the MN to perform the application tasks (i.e. stay operational) during most of the handover (at least in the intra-domain handover). We will use our definition when measuring the handover delay but this should be remembered when comparing with other algorithms.

To measure the delay the code is instrumented as described in Section 6.2 (timing metrics). The MN is placed in close range to two APs and the handover is forced, i.e. it is not initiated as defined by the algorithm. Figure 6.3 shows the results. We included an approximation of the handover delay for a naive IEEE 802.15.4 handover solution, i.e. a handover in which all channels are scanned. Since a system (almost) always operates on one of the bands only we chose the global 2.4 GHz band which consists of sixteen channels. Each channel is scanned for 500 ms and the association takes approximately 400 ms (see the conclusion section below), resulting in $500 \cdot 16 + 400 = 8400$ ms handover delay. This is also the expected delay of the proposed handover algorithm if the neighbor channel scan during the inter-domain handover fails.

Conclusion

The results show on the importance of the protocol. The time between sending a *handover_request* message and receiving a *handover_execute* message takes less than 300 ms for an intra-domain handover (otherwise it would not be an intra-domain handover, see Section 6.3.1). This means that the association procedure takes more than 400 ms. Since the MAC protocol association procedure is used, there is nothing the algorithm can do to reduce the association time.

The comparison of our proposed handover algorithm and the existing handover algorithms for WSNs—in terms of delay—is done by analyzing the most time consuming part of

Figure 6.3 Empirical evaluation of handover delays. The chart shows the average time for performing a specific handover type (the standard deviation is negligible). Two neighbor channels were used, each channel was scanned for 500 ms and 100 handovers were performed. An approximation of the handover delay for a naive IEEE 802.15.4 handover solution is included for comparison.

the handover process: the Scan. Inter-MARIO uses a selective scan, similar to the neighbor channel concept and should therefore have roughly the same handover delay as the inter-domain handover delay in our proposed algorithm. Smart-HOP and Fotouhi et al. [4] do not provide details on the scan procedure (e.g. which channels to scan) but uses an ordinary MAC scan and hence is expected to have at least the same delay as the proposed algorithm’s inter-domain handover. S-GinMob claims to have a handover delay equal to zero. LoWMob does not explicitly evaluate the algorithm in terms of handover delay but we can conclude from the results that the handover delay is below 100 ms. The proposed algorithm will probably not be able to match S-GinMob and LoWMob even if the MAC layer protocol is optimized for handover. Hence, our proposed handover algorithm positions itself somewhere in the middle of the existing handover algorithms in terms of handover delay.

The conclusion is that the QoS in terms of handover delay provided by the proposed algorithm is expected to be good enough for most WSNs.

6.4.2 Number of Handovers and Packet Loss

These two metrics are measured by instrumentation as described in Section 6.2 (discrete event metrics). The test bed is built in a more or less empty building with dimensions 13×6 meters. As mentioned in Section 5.2.1; the MN’s and AP’s radios have a range of around 100 meters indoors. This is far too long for easily manageable test beds. We were able to disable the amplifier in the front end and lower the transceiver transmit power to -17 dBm. This resulted in a range below 10 meters. It was even possible to completely turn off the front end but this resulted in a range below 50 centimeters which is too short.

Two test cases are considered: 1) sparse deployment of APs and 2) dense deployment

of APs. Two APs and one MN are used. The APs are positioned with thirteen meters intermediate distance in the sparse deployment test case and six meters intermediate distance in the dense deployment test case. As mentioned in Section 5.6.3; the algorithm's *observe_frequency* parameter is not used. The MN only checks for connection if we explicitly tell it to, i.e. an observe is initiated manually. Points are marked on the floor (to be able to repeat the same test several times) where observations are performed. At each point three observes and one sensor data send is performed. In the sparse deployment test case the APs are in different domains while they are in the same domain in the dense test case.

Test Case: Sparse Deployment of Access Points

The test bed is shown in Figure 6.4. The MN is moved from point A to point K and back to A again, running the test described in the previous paragraph at each point. This is done five times. The APs are placed with as little overlap as possible. This test case can be considered a worst case and the results show on poor performance, see Figure 6.5. The number of reconnections measures the number of full scans (i.e. scanning all channels). If several full scans are performed before finding an AP, each counts as one reconnection. The optimal would be two inter-domain handovers (and, of course, two failed intra-domain handovers): one when moving from point A to point K and one on the way back.

Figure 6.4 A test setup for a sparse deployment of APs (test case 1). The two APs are in two different domains and are placed with a minimal overlap.

Interesting things were observed in this test case. The handover was expected to take place between the D and H points. Most handovers did take place here, see Figure 6.6, and we noticed some redundant handovers (even some ping-pong effects), as expected. However, some handovers took place at point B and J. In one of the test rounds a handover was performed from AP B to AP A at point J. This is very surprising since the AP A was supposed to be out of range at this point and definitely not have a higher LQ than AP B. A handover at point B when being associated with AP A was initiated several times.

Figure 6.5 Empirical evaluation of number of handovers and packet loss (test case 1). The chart shows the average number of handovers of a certain type and the corresponding (sample) standard deviation. It also shows the average number of packets lost.

Figure 6.6 Probability distribution of where the handovers are done (test case 1).

Further investigation of this phenomenon showed that the LQ does not fade nicely in an indoor environment even though there is a free line-of-sight to the AP. Intuition tells that the LQ should fade similar to the left part of Figure 6.7. However, the fading look much more like the right part of the same figure. The reason for this is probably due to multipath propagation. Thus, even if the MN is very close to the AP and has free line-of-sight the LQ may be very low. Furthermore, the LQ is not constant at certain position; some variations were observed. The most likely reason for this variation is interference from external sources, e.g. nearby Wi-Fi networks.

Figure 6.7 The left part shows the expected fading of the LQ for an AP in free line-of-sight. The right part shows the actual fading in an indoor environment. Darker color means higher LQ value.

Test Case: Dense Deployment of Access Points

This test case is performed in a very similar manner to the previous one. The test bed is shown in Figure 6.8. The MN is moved from point A to point I and this is repeated ten times. The results are shown in Figure 6.9 and Figure 6.10. This is a more “normal” case—in terms of coverage—than the previous one and here we see the strength of the algorithm. The optimal number of handovers is one (of the intra-domain type) and the proposed algorithm is able to match this.

Figure 6.8 A test setup for a dense deployment of APs (test case 2). The two APs are in the same domain and are placed with a large overlap.

Figure 6.9 Empirical evaluation of number of handovers and packet loss (test case 2). The chart shows the average number of handovers of a certain type and the corresponding (sample) standard deviation. It also shows the average number of packets lost.

Figure 6.10 Probability distribution of where the handovers are done (test case 2).

Conclusion

The “normal” test case (dense deployment of access points) shows how well the proposed algorithm performs. It does no redundant handovers and the number of lost packets is low. However, the worst case test case (sparse deployment of access points) shows on very poor performance. But blaming it on the algorithm would be unfair; the main reason for the reduced performance is the fading effects described above (illustrated in Figure 6.7) and will most likely cause a great reduction in performance for every handover algorithm. This shows how important a dense deployment of APs is, in order to reduce the number of (redundant) handovers. A dense deployment with overlaps between APs is necessary to cover each other’s “blind spots”. It may also be worth considering

incorporate another parameter in the LQ value, e.g. RSSI. This is what the default Atmel implementation of the IEEE 802.15.4 standard did, before we changed it. The LQ could be high, even though the PER is high, if the RSSI is large. However, this does not change the fact that the PER is high, but could make the algorithm a bit more stable, by not initiating a handover if both the PER and the RSSI is high, thus hopefully avoid redundant handovers.

The number of handovers is mainly decided by how well the Initiate and Select parts of the algorithm are designed. Smart-HOP and Fotouhi et al. [4] are the two existing handover algorithms which consider these parts in most detail. We have carefully considered the conclusions regarding handover initiation and new AP selection provided by Smart-HOP, i.e. the selection of hysteresis margin and stabilizing the LQ with a sampling window, when designing the proposed algorithm. Further, the selection of the new AP suggested by Fotouhi et al. [4] is based on the LQ value, using the F-LQE [29] LQ estimator. The proposed algorithm uses a similar (fuzzy logic) approach.

The other existing algorithms do pay less attention to the Initiate and Select parts, which most likely results in more redundant handovers than the algorithms described above (and, consequently, our proposed handover algorithm). LowMob initiates the handover based on a single read of the link quality; no stabilizing window is used. Moreover, the new AP is selected based on location only, which is a bad idea and could lead to a lot of redundant handovers. S-GinMob bases the handover initiation on the RSSI, the packet loss rate and number of lost packets—each considered individually and compared with a threshold value. The RSSI value is stabilized using a sampling window. The foundation of S-GinMob is described by Silva et al. [47]; from which we can conclude that the new AP is selected based on RSSI (it selects the AP with the highest RSSI). Inter-MARIO bases the handover initiation on movement detection but does not specify how movement is detected nor how the new AP is selected.

Neither of the existing solutions described above is expected to have high packet loss; as most of them focus on QoS aspects. And, as the empirical evaluation showed; the packet loss rate is low for the proposed handover algorithm as well. In both test cases the number of lost packets is low; indicating that the proposed algorithm provides a high QoS (in terms of lost packets). However, this could be due to the discrete nature of the test cases in addition to the fact that observes were performed before trying to send sensor data. Reversing this order may result in a different outcome with higher packet loss.

To conclude: the proposed handover algorithm is expected to be one of the best (maybe even the best) in avoiding redundant handovers and provides a high QoS in terms of packet loss. We will have more to say about the two test cases as we analyze the energy consumption in the next section.

6.4.3 Energy Consumption

As mentioned throughout the thesis; we are not interested in the algorithm's characteristics for a particular system. We therefore derive a general equation for calculating the energy (E) consumed by the algorithm

$$E = q \cdot E_{handover} + r \cdot E_{send} + s \cdot E_{observe}, \quad (6.1)$$

where q is the number of handovers performed, r is the number or retransmitted packets (due to being associated with a bad access point) and s is the number of observes

performed. Further, $E_{handover}$ is the energy spent on performing a handover, E_{send} is the energy consumed for sending a packet and finally, $E_{observe}$ is the energy spent on performing one observation. The first term can be further split

$$q \cdot E_{handover} = x \cdot E_{intra_domain} + y \cdot E_{inter_domain} + z \cdot E_{reconnect}, \quad (6.2)$$

where x , y and z are the number of intra-domain handovers, inter-domain handovers and reconnections (i.e. full scans) respectively ($x+y+z = q$). E_{intra_domain} , E_{inter_domain} and $E_{reconnect}$ are the energy spent on performing these three types of handovers, respectively. These can be further split into MAC primitives

$$E_{intra_domain} = E_{broadcast} + E_{receive}, \quad (6.3)$$

$$E_{inter_domain} = U \cdot E_{scan}, \quad (6.4)$$

$$E_{reconnect} = V \cdot E_{scan}, \quad (6.5)$$

where $E_{broadcast}$, $E_{receive}$ and E_{scan} are the energy spent on performing a broadcast, receive and channel scan, respectively. U is the average number of neighbor channels and V is the number of channels used by the system (i.e. the number of channels scanned in a full scan).

On most systems, $E_{receive}$, $E_{broadcast}$ and E_{send} are in the same order of magnitude (as we saw in Section 3.2.1). Let us replace each of them with $E_{transceive}$. Using this, Equation (6.5), Equation (6.4) and Equation (6.3) in Equation (6.2) gives us

$$q \cdot E_{handover} = 2x \cdot E_{transceive} + (yU + zV) \cdot E_{scan}. \quad (6.6)$$

Using Equation (6.6) in Equation (6.1) and $E_{send} = E_{transceive}$, gives us the final equation for the algorithm's energy consumption

$$E = (2x + r) \cdot E_{transceive} + (yU + zV) \cdot E_{scan} + s \cdot E_{observe}. \quad (6.7)$$

The calculations (i.e. the fuzzy logic) done by the Observe part of the algorithm should only require a few instructions, thus being negligible compared to the MAC synchronize in terms of energy consumption (see Section 3.2.1). Moreover, s is decided by the application's requirements on the QoS. This means that the proposed algorithm has minimized the $s \cdot E_{observe}$ term as much as possible; reducing it further requires changes to either the MAC protocol or the application's QoS requirements.

Test Case: Sparse Deployment of Access Points

This test case is described in Section 6.4.2. By using Equation (6.7) we can calculate the energy consumption (x , y , z and r are given by Figure 6.5, $s = 66$, $U = 2$ and $V = 15$)

$$E = 7.2 \cdot E_{transceive} + 41.6 \cdot E_{scan} + 66 \cdot E_{observe}.$$

Test Case: Dense Deployment of Access Points

This test case is described in Section 6.4.2. By using Equation (6.7) we can calculate the energy consumption (x , y , z and r are given by Figure 6.9, $s = 27$, $U = 2$ and $V = 15$)

$$E = 2.6 \cdot E_{transceive} + 0 \cdot E_{scan} + 27 \cdot E_{observe}.$$

Conclusion

Measuring the energy consumption further shows how well the proposed algorithm performs in the “normal” test case (dense deployment of access points). Not a single scan was needed and very few transmissions and receptions were performed. The existing handover algorithms for WSNs often perform several scans (Inter-MARIO at least uses a selective scan, similar to the neighbor channel concept). S-GinMob scans actively for a better AP even if the current one is good enough; clearly not acceptable from an energy point of view. Since $E_{scan} \gg E_{transcive}$ on most systems¹—often several orders of magnitude larger—it is proven how little energy our proposed algorithm actually uses, especially compared with existing handover algorithms for WSNs. LoWMob is the only one expected to have roughly the same energy consumption as our proposed algorithm. LoWMob uses a network controlled handover; similar to our proposed handover algorithm, but it is a location based handover which is—as we saw in Section 4.7.4—not feasible due to the great risk of selecting a bad access point since the LQ is not considered. Thus, LoWMob is expected to have higher energy consumption than the proposed algorithm since it will most likely perform more redundant handovers.

Looking at the energy consumption further emphasizes how important a dense deployment of APs is. The intuition tells that 4 channel scans (2 neighbor channel scans) would be needed in the sparse deployment test case (since two neighbor channels were used) but in reality 41.6 channel scans (2.8 neighbor channel scans and 2.4 full scans) were performed (on average). This clearly is not acceptable. As discussed in Section 6.4.2; most (probably all) handover algorithms will get a reduced performance in sparse networks. However, the proposed algorithm can surely be improved to handle this case better; the most obvious change being to perform a sleep between each consecutive full scan, as discussed in Section 4.4.2.

The results are fairly preliminary; more test cases and tests are required in order to tell something for sure but some preliminary trends have been observed, sketched in Figure 6.11. The total number of handovers seems to increase as the coverage decreases. The main reason for this is that the MN is often forced to handover to an AP with bad LQ, due to the absence of better in a sparse deployment. Moreover, a sparse deployment results in a lot of “blind spots” (see Figure 6.7) where no AP is found, leading to a lot of MN scans when being in these spots. The number of intra-domain handovers (x) and the number of inter-domain handovers (y) is expected to increase while the coverage decreases. Since the proposed algorithm requires a certain threshold for an AP to be a candidate, x and y only increases to a certain limit; when the coverage is very low the MN will stay in the handover and only select a new AP when a good enough is found. This is really severe from an energy point-of-view as this means that the MN will perform several full scans (z), resulting in a very high energy consumption. The aim should be to have enough coverage to keep $z = 0$; a neighbor channel scan should be enough. As noticed earlier: the number of lost packets (r) seems to be constant (and low) as the coverage varies (i.e. independent of the coverage); the handover thus does its job providing a high QoS while not wasting a lot of energy on retransmitting packets.

To conclude: the proposed handover algorithm reduces the energy consumption by several orders of magnitude—compared to existing handover algorithms for WSNs. Moreover, the coverage is shown to be of great importance for the handover performance: in order

¹Since the radio has to be turned on for much longer during a scan than for transmitting or receiving a message.

Figure 6.11 A rough sketch of the expected relation between AP coverage and the number of intra-domain handovers (x), number of inter-domain handovers (y), number of full scans (z) and number of packet loss (r).

to reduce the energy consumption on the MN good coverage is very important. The coverage should be good enough for a MN to always have at least one AP with good LQ in range.

Discussion and Conclusion

WE begin this concluding chapter by summarizing the lessons learned regarding handover in WSNs with focus on low power. The proposed algorithm is then analyzed in terms of energy efficiency. We also take a brief look at possible future work. The chapter—and the thesis—are concluded by some final words.

7.1 Handover in Wireless Sensor Networks

Handover in WSNs is an unexplored area in the field of communication systems. Especially from an energy perspective; almost all of the existing research focuses on QoS. However—as we have seen throughout the thesis—energy is often a much more important parameter to WSNs than QoS. Moreover, the unique characteristics of WSNs make existing handover techniques used in traditional communication systems unsuitable, see Section 3.1.

Our main question at issue (given in Section 1.2) was:

How to keep the energy consumption at a minimum when doing a handover?

The simple answer would be to put all the work on the network, i.e. the APs (assuming they are mains powered). The answer is however not this simple. For example, the parameters used to decide whether a handover should be initiated or not often has to be gathered on the MN. In order for the network to make a handover decision these would have to be communicated, i.e. using a great amount of power. Therefore the observation and handover initiation decision should be the MN's responsibility. The Scan and Select parts, however, can—and should—be put on the network. We concluded that the Scan is the most energy consuming part of a handover. Making it the network's responsibility will reduce the energy consumption on the MN by several orders of magnitude.

Further, in order to keep the energy consumption at a minimum there are several other things to consider, affecting the algorithm (as we saw in Section 3.2); ranging from protocols and network structure to QoS and application requirements. Trade-offs must be made between contradictory conditions. Therefore a general algorithm for handover will (at its best) be a Jack of all trades, *master of none*. Designing this kind of algorithm

is very difficult since there are so many differences between WSN systems and their requirements.

Moreover, the objective was to also have high energy efficiency, i.e. reducing the energy spent on handover should not severely decrease the QoS. However, generally speaking, higher QoS requires higher energy consumption. Taking connectivity as an example, in order to ensure connectivity to the network some sort of polling action is needed, e.g. listening for beacons at regular intervals. This will obviously increase the energy consumption as the radio has to be active.

7.2 Algorithm Analysis

In Chapter 4 we proposed an energy-efficient handover algorithm for WSNs. In this section it is analyzed further in terms of energy consumption and QoS—to be able to analyze the energy efficiency of the algorithm.

7.2.1 Energy Consumption

As we have seen throughout the thesis; the algorithm has an extremely low energy usage on the MN. The main contribution to this is the optimization of the most energy consuming part of the algorithm: the Scan part. Instead of having the radio on and listening for beacons by nearby APs, the MN broadcasts a handover request message to all nearby APs. With this approach the radio has to be on only for a couple of milliseconds (sending the broadcast message and receiving the handover execute message), instead of hundreds or even thousands of milliseconds if a MAC scan approach would have been used. This saves a tremendous amount of energy.

Let us look at an example. The IEEE 802.15 standard does not define how long a scan on each channel should be but Staehle [42] argues that 0.26 seconds per channel is a reasonable time. Assume that the global band is used, hence up to sixteen channels. This results in a radio-on time for $260 \cdot 16 = 4000$ ms. Our empirical evaluation in Section 6.3.1 showed that broadcasting a message took 2.4 ms on average and receiving a handover execute message took 3.2 ms. This results in a radio on time for $2.4 + 3.2 = 5.6$ ms. This means that a traditional scan uses $4000/5.6 \approx 714$ times more energy¹ than our proposed algorithm. Even if the reception of the handover execute message is not optimal, e.g. the MN has to periodically turn the radio on to listen for it, a great amount of energy is still saved.

If the intra-domain handover fails the algorithm uses the neighbor channels concept. This reduces the energy consumption by a factor Ψ/Ω ², where Ψ is the total number of channels used and Ω is the average number of neighbor channels.

The proposed algorithm also makes an effort in avoiding redundant handovers. It uses position knowledge to its advantage and takes several parameters into consideration—by a fuzzy logic approach—when making a handover initiation and selection decision. The unreliable LQ parameters are stabilized. Further, the hysteresis margin is large which also helps in the avoidance of redundant handovers. Finally, the algorithm suggests

¹Energy consumed by the transceiver operating mode switches is not included in the calculation.

²Actually slightly less, due to the fact that an intra-domain handover try is performed before an inter-domain handover.

that context data needed by the new AP (and possessed by the previous AP) should be transferred between the APs instead of involving the MN.

To summarize, these are the main features of the algorithm:

- Avoiding redundant handovers by considering several parameters at the same time by a fuzzy logic approach as well as stabilizing the most unstable ones.
- Highly optimized scan procedure (it totally avoids the MAC scan in an intra-domain handover and optimizes it through the neighbor channel concept in an inter-domain handover).
- Built-in load balancing.
- Context data transfers between APs.

7.2.2 Quality of Service

One obvious advantage with the proposed algorithm is that a MN can keep transmitting and receiving data packets with its current AP while the network selects the new AP. Thanks to this only a short break in the connection is needed when the MN receives the handover execute message. This results in the handover being very seamless from the MN point of view. The algorithm also has built-in load balancing which should help in improving overall QoS in the network. Converting the QoS to a figure is not an easy task since it comprises several parameters. Moreover, the QoS is application dependent; a QoS that is high for one application might be low for another.

7.2.3 Energy Efficiency

As Section 7.2.1 pointed out; we have seen throughout the thesis that our proposed algorithm has a remarkable low energy consumption compared to traditional handover algorithms as well as existing handover algorithms for WSNs. Moreover, the QoS is not expected to be much lower than for the existing handover algorithms for WSNs. In fact, the proposed algorithm can actually provide a higher QoS than some of the existing ones since the MN can stay operational during the Scan. Hence, the proposed algorithm is very energy efficient since it uses much less energy to provide (roughly) the same QoS.

7.2.4 Weaknesses

The greatest limitation of the proposed algorithm is the restriction to mains powered APs. Our algorithm is not designed for wireless ad hoc networks where all the nodes are battery powered. The network solution used by the proposed algorithm should work well for most types of WSNs, however, if the network has very high load the algorithm will reduce the performance. The solution also increases the complexity of the network. Moreover, the proposed algorithm will scale badly if only the number of MNs increases while the number of APs remains constant (since the intra-domain handover part of the algorithm is network controlled). If a mobile controlled handover is chosen also as the intra-domain handover strategy, at least three things should be taken from the proposed algorithm:

- Avoiding redundant handovers by considering several parameters and stabilize the unstable ones.
- Smart channel selection for the scan, e.g. via neighbor channels.
- Context switch, i.e. transfer information from the previous AP to the new one.

The proposed algorithm requires the implementation to provide some functions (see Appendix A) but as we saw in the implementation (Chapter 5); these are often simple. They are not provided by the algorithm because they depend on the data structures used by the implementation, e.g. for representing the associated MNs at an AP.

Depending on the MAC protocol there might be some issues to solve. We did for example notice that it is difficult for the new AP to communicate with the MN (when sending the handover execute message) since the MN synchronizes with the beacon frames send by the previous AP. This was solved by changing a setting in the MAC protocol telling the mobile node to not synchronize with beacon frames and have to radio always on. This solution is of course not feasible in a real system since it would result in large energy consumption, similar to the ordinary scan. A solution might be to let the MN periodically broadcast some sort of message asking the network if a new AP has been found.

The communication issue described above should not be a problem in the low power duty cycle MAC protocols since it is easy for the new AP to communicate with the MN knowing the MN's slot. However, in these types of protocols a broadcast might be problematic, since the algorithm requires the energy used by a broadcast to be in the same order of magnitude as the send and receive. This is not the case if a broadcast requires sending the data multiple times (one time in each slot). A solution might be to use a slot dedicated for broadcast and use e.g. CSMA in this slot.

7.3 Future Work

There is still a lot of work to be done. One example is to implement the algorithm in a more realistic WSN, this to be able to perform an empirical evaluation in a more realistic scenario: with people carrying MNs across different rooms. This will test the Observe part of the algorithm more thoroughly. It should also be tested alongside an application and with significant load on the network, i.e. using several MNs.

To really prove its low power characteristics the WSN should be focused on low power, i.e. use some sort of duty cycle scheme, e.g. TDMA. The algorithm should also be tailored for this particular system, i.e. optimize the parameters for this particular WSN and incorporate handover in the protocol stack. This will enable an investigation of how big an impact the handover has on the energy consumption, by comparing the proposed algorithm with a traditional algorithm in such system. It should also be tested in different kinds of WSNs to prove that it indeed is a generic algorithm. Further, it would be very interesting to test the algorithm in a system that provides location data. Another thing left to be done is to adapt the algorithm to be run in a simulation tool. Doing both an empirical evaluation and a simulation is the best way to evaluate the algorithm. It would be interesting to investigate the fading effects described in Section 6.4.2 further to be able to make the algorithm better at handling this phenomenon. Finally, mobility pattern awareness may be added to the algorithm if the WSN has very predictable patterns.

7.4 Final Words

This thesis has been a long but pleasant journey in the topic of handover for WSNs. Solving the handover problem within individual WSNs is just the beginning; the new IoT paradigm will require handover algorithms to handle handovers between different WSNs. This will arguably be a great challenge due to the differences between these systems.

I would like to conclude the thesis with a quote:

*"Now this is not the end. It is not even the
beginning of the end. But it is, perhaps, the end of
the beginning."*

— Sir Winston Churchill

The Proposed Algorithm: Parameters and Functions

In this appendix we list and describe the parameters and functions used by the proposed algorithm. It is up to the implementation of the algorithm to decide the values of the parameters as well as define the functions listed here. The functions are not defined by the algorithm due to their dependences on things like specific protocol details and/or data structures.

A.1 Algorithm Parameters

Table A.1 lists the parameters used by the algorithm.

Table A.1 Parameters used by the proposed algorithm.

Parameter	Description	Unit
<i>observe_frequency</i>	Defines how often an observation shall take place.	MN
<i>LQ_window_size</i>	Defines how many observations to consider when deciding if a handover should be initiated or not.	MN
<i>request_stability</i>	Defines the length of the stability data field in the <i>handover_request</i> message.	MN
<i>handover_select</i>	Defines the time from receiving the first <i>handover_notify</i> message to the selection of AP.	AP
<i>intra_domain_handover</i>	Defines the time from sending a <i>handover_request</i> message to the switch to an inter-domain handover (if a <i>handover_execute</i> message is not received).	MN

A.2 Access Point and Mobile Node: Common Functions

Table A.2 lists the functions used by both the AP and MN.

Table A.2 Functions required by both the AP and MN by the proposed algorithm.

Function	Description
CREATE_MESSAGE (<i>message_type</i>)	Creates the message type (see Figure 4.2) decided by the <i>message_type</i> argument to be sent with either the MAC protocol's send or the network protocol's send.
CALC_RELATIVE_VELOCITY (<i>A</i> , <i>B</i> , <i>B'</i> , <i>dt</i>)	Calculates the relative velocity, see Section 4.2.1. (Only needed if location data is available.)

A.3 Access Point: Functions

Table A.3 lists the functions used by the AP (only).

Table A.3 Functions required by the AP by the proposed algorithm.

Function	Description
FUZZY_EVAL_CANDIDATE (<i>LQ</i> , [<i>relative_velocity</i>])	Maps the arguments to their respective fuzzy set and makes a fuzzy decision in accordance with Table 4.2. The <i>LQ</i> is mapped in accordance with Figure 4.4.
HANDLE_HANOVER (<i>MN</i> , <i>candidateAP</i>)	Initiates a handover for a MN by (at least) starting the <i>handover_select</i> timer (see Section 4.6.1) for this MN, if a handover is not already initiated. If it is, it checks the <i>candidateAP</i> and if this is better than the current best AP this is updated. The candidates are compared as described in Section 4.3.2.
DISASSOCIATE (<i>MN</i>)	Disassociates a MN.

A.4 Mobile Node: Functions

Table A.4 lists the functions used by the MN (only).

Table A.4 Functions required by the MN by the proposed algorithm.

Function	Description
ADD_LQ_TO_WINDOW (LQ , LQ_window)	Adds a LQ value to the sampling window for a MN and returns the average LQ for the values currently in the window.
FUZZY_EVAL_INIT (LQ , [$relative_velocity$])	Same as FUZZY_EVAL_CANDIDATE (see Table A.3) but uses the fuzzy logic matrix given in Table 4.1 instead.
FIND_BEST_AP (AP_list)	Finds the AP with the highest LQ given a list of APs.

The Proposed Algorithm: Flowchart

This appendix shows the proposed algorithm in flowchart form. The Idle state means “continue with whatever task a unit was doing before the flowchart was entered”.

B.1 Monitor

The Monitor part, i.e. Observe and Initiate parts of the proposed algorithm is illustrated by Figure B.1. This figure also shows the mobile node's handling of a received neighbor channels message.

Figure B.1 The left flowchart shows the mobile node's Monitor part (Observe and Initiate parts) of the proposed algorithm. The right flowchart shows the mobile node's handling of a received neighbor channels message.

B.2 Selection and Switch: Mobile Node

Figure B.2 illustrates the mobile node's Selection part, i.e. the scan and select parts, as well as the data link layer switch of the proposed algorithm.

Figure B.2 The mobile node's flowchart of the Selection (Scan and Select parts) and Switch part (Data link layer switch part) of the proposed handover algorithm.

B.3 Selection: Access Point

The access point's handling of the Selection part, i.e. the Scan and Select parts of the proposed algorithm is illustrated in Figure B.3.

Figure B.3 The access point's flowcharts of the Selection part (Scan and Select parts) of the proposed handover algorithm. The left flowchart shows the handling of a received *handover_request* message, the right part shows ditto for a received *handover_notify* message.

B.4 Switch: Access Point

Figure B.4 illustrates the access point's Switch part, i.e. the Data link layer and Network layer switch parts of the proposed algorithm.

Figure B.4 The access point's Switch part (Data link layer and Network layer switch parts) flowcharts of the proposed algorithm. The left flowchart shows the handling of a received *handover_select* message and the right flowcharts shows how a network layer switch is performed.

Command Line Interface

Figure C.1 shows the mobile node's command line interface.

```
=====

*****
Mobile Node Handover Evaluation Application, Aug 26 2013
*****

==== STATUS =====

RFX2411 on = 1
Tx power = -17 dBm
My long address: 0xDCDFEED30D4D4CDB
Channel = 20
PAN-ID = 0xBEEF
My short address = 0x0101
Access point short address = 0x0100
Neighbor channels = 0x02100000
Average LQI = 242
State = CONNECTED

==== EVALUATION METRICS =====

Number of handovers: 10
Number of intra handovers: 6
Number of (re-)connections: 0
Number of packets: 109
Number of failed packets: 0
Handover delay:
Mean: 1200.099999
Variance: 314325.888888

==== MENU =====

(S) : Send sensor data
(U) : Synchronize with access point
(M) : Initiate handover
(D) : Display this menu
(O) : Run handover test
(Y) : Run evaluation test
(B) : Reset evaluation metrics
(R) : Reset

*****
=====
```

Figure C.1 The mobile node's command line interface.

Bibliography

- [1] Chipara, O., Lu, C., Bailey, T. C., and Roman, G. C., *Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit*, Proc. Conference on Embedded Networked Sensor Systems, ACM, 2010.
- [2] Lee, S. H., Lee, S., Song, H., and Lee, H. S., *Wireless sensor network design for tactical military applications: remote large-scale environments*, Proc. Military Communications Conference, IEEE, 2009.
- [3] Dyo, V., Ellwood, S. A., Macdonald, D. W., Markham, A., Mascolo, C., Pásztor, B., and Yousef, K., *Evolution and sustainability of a wildlife monitoring sensor network*, Proc. Conference on Embedded Networked Sensor Systems, ACM, 2010.
- [4] Fotouhi, H., Alves, M., Koubaa, A., and Baccour, N., *On a reliable handoff procedure for supporting mobility in wireless sensor networks*, Proc. International Workshop on Real-Time Networks in conjunction with the Euromicro International Conference on Real-Time Systems, 2010.
- [5] Dargie, W., and Poellabauer, C., *Fundamentals of wireless sensor networks: theory and practice*, Wiley Publishing, 2010.
- [6] Pollini, G. P., *Trends in handover design*, Communications Magazine, IEEE, vol. 34, pp. 82-90, 1996.
- [7] Akyildiz, I. F., Weilian Su, Sankarasubramaniam, Y., Cayirci, E., *A survey on sensor networks*, Communications magazine, IEEE, vol. 40, pp. 102-114, 2002.
- [8] García Castaño, J., *Algorithms and protocols enhancing mobility support for wireless sensor networks based on Bluetooth and Zigbee*, PhD Thesis, Mälardalen University, 2006.
- [9] Zuniga, M., and Krishnamachari, B., *Analyzing the transitional region in low power wireless links*, Proc. Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE, 2004.
- [10] Raghunathan, V., Schurgers, C., Park, S., Srivastava, M. B., *Energy-aware wireless microsensor networks*, Signal Processing Magazine, IEEE, vol. 19, pp. 40-50, 2002.
- [11] Culler, D. E., Dutta, P., Ee, C. T., Fonseca, R., Hui, J., Levis, P., and Zhao, J., *Towards a sensor network architecture: Lowering the waistline*, Proc. Hot Topics in Operating Systems Symposium, 2005.
- [12] Atzori, L., Iera, A., and Morabito, G., *The internet of things: A survey*, Computer Networks, vol. 54, pp. 2787-2805, 2010.

- [13] The Internet of Things - Architecture website, <http://www.iot-a.eu/public>, fetched 2013-06-18.
- [14] Dong, Q., and Dargie, W., *A survey on mobility and mobility-aware MAC protocols in wireless sensor networks*, Communications Surveys and Tutorials, IEEE, vol. 15, pp. 88-100, 2012.
- [15] Silva, R., Silva, J. S., and Boavida, F., *MIPv6 Soft Hand-off for Multi-Sink Wireless Sensor Networks*, Smart Wireless Sensor Networks, Intech, pp. 147-155, 2010.
- [16] Soliman, H., Bellier, L., and Malki, K. E., *Hierarchical mobile IPv6 mobility management (HMIPv6)*, RFC 5380, 2008.
- [17] Abdelwahed, B. and Youssef, H., *A fast handover scheme for proxy-based mobility in wireless sensor networks*, Proc. International Symposium on Mobility Management and Wireless Access, ACM, 2012.
- [18] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and Patil, B., *Proxy mobile ipv6*, RFC 5213, 2008.
- [19] Bag, G., Raza, M. T., Kim, K. H., and Yoo, S. W., *LoWMob: Intra-PAN mobility support schemes for 6LoWPAN*, Sensors, vol. 9, pp. 5844-5877, 2009.
- [20] Ali, M., Suleman, T., and Uzmi, Z. A., *MMAC: A mobility-adaptive, collision-free mac protocol for wireless sensor networks*, Proc. International Performance, Computing, and Communications Conference, IEEE, 2005.
- [21] Dargie, W., *A medium access control protocol that supports a seamless handover in wireless sensor networks*, Journal of Network and Computer Applications, vol. 35, pp. 778-786, 2012.
- [22] Pham, H., and Jha, S., *An adaptive mobility-aware MAC protocol for sensor networks (MS-MAC)*, Proc. International Conference on Mobile Ad-hoc and Sensor Systems, IEEE, 2004.
- [23] Zinonos, Z., and Vassiliou, V., *S-ginmob: Soft-handoff solution for mobile users in industrial environments*, Proc. International Conference on Distributed Computing in Sensor Systems and Workshops, IEEE, 2011.
- [24] Fotouhi, H., Zúñiga, M., Alves, M., Koubâa, A., and Marrón, P., *Smart-hop: a reliable handoff mechanism for mobile wireless sensor networks*, Proc. European Conference on Wireless Sensor Networks, 2012.
- [25] Ha, M., Kim, D., Kim, S. H., and Hong, S., *Inter-MARIO: a fast and seamless mobility protocol to support Inter-PAN handover in 6LoWPAN*, Proc. Global Telecommunications Conference, IEEE, 2010.
- [26] Shih, E., Cho, S. H., Ickes, N., Min, R., Sinha, A., Wang, A., and Chandrakasan, A., *Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks*, Proc. International Conference on Mobile Computing and Networking, ACM, 2001.
- [27] Atmel, *AT86RF233 datasheet*, http://www.atmel.com/Images/Atmel-8351-MCU_Wireless-AT86RF233_Datasheet.pdf, fetched 2013-08-16.

-
- [28] Xu, Y., Heidemann, J., and Estrin, D., *Geography-informed energy conservation for ad hoc routing*, Proc. International Conference on Mobile Computing and Networking, ACM, 2001.
- [29] Baccour, N., Koubâa, A., Youssef, H., Jamâa, M. B., do Rosário, D., Alves, M., and Becker, L. B., *F-lqe: A fuzzy link quality estimator for wireless sensor networks*, Proc. European Conference on Wireless Sensor Networks, 2010.
- [30] The 6LoWPAN website, <http://datatracker.ietf.org/wg/6lowpan/charter/>, fetched 2013-08-09.
- [31] Park, S. H., Kim, H. S., Park, C. S., Kim, J. W., and Ko, S. J., *Selective channel scanning for fast handoff in wireless LAN using neighbor graph*, Proc. International Technical Conference on Circuits/Systems, Computers and Communications, 2004.
- [32] Atmel, *AT91SAM ARM-based Flash MCU datasheet*, http://www.atmel.com/Images/Atmel_6500_32-bit-Cortex-M3-Microcontroller_SAM3S_Datasheet.pdf, fetched 2013-08-16.
- [33] Skyworks, *SE2431L: 2.4 GHz ZigBee/802.15.4 Front-End Module datasheet*, http://www.skyworksinc.com/uploads/documents/SE2431L_202410D.pdf, fetched 2013-08-16.
- [34] Ams AG, *AS1375 datasheet*, <http://www.ams.com/eng/Products/Power-Management/LDOs/AS1375>, fetched 2013-08-16.
- [35] Rfaxis, *RFX2411 datasheet*, https://www.rfmw.com/data/RFaxis_RFX2411_DS_R1p4.pdf, fetched 2013-08-16.
- [36] Atmel, *AT91SAM9G20 datasheet*, <http://www.atmel.com/Images/doc6384.pdf>, fetched 2013-08-16.
- [37] The Contiki website, <http://www.contiki-os.org/>, fetched 2013-08-10.
- [38] IPSO Alliance, *RPL: The IP routing protocol designed for low power and lossy networks*, <http://www.cs.berkeley.edu/~jwhui/6lowpan/IPSO-WP-7.pdf>, fetched 2013-08-12.
- [39] Atmel, *Atmel AVR2025: IEEE 802.15.4 MAC Software Package - User Guide*, <http://www.atmel.com/Images/doc8412.pdf>, fetched 2013-08-12.
- [40] Institute of Electrical and Electronics Engineers, Inc., *IEEE Std. 802.15.4-2006: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, New York, IEEE Press, 2006.
- [41] The ZigBee website, <http://www.zigbee.org/>, fetched 2013-08-12.
- [42] Staehle, B., *Optimizing the Association Procedure for Low-Power 802.15.4 Nonbeacon Sensor Networks*, Proc. International IFIP-TC 6 Networking Conference, 2009.
- [43] Garcia-Sanchez, A. J., Garcia-Sanchez, F., and Garcia-Haro, J., *Optimized orphan algorithm for IEEE 802.15.4 networks*, Proc. International Symposium on Wireless Communication Systems, 2009.

- [44] Zuniga, M., and Krishnamachari, B., *Analyzing the transitional region in low power wireless links*, Proc. Sensor and Ad Hoc Communications and Networks, IEEE, 2004.
- [45] Osterlind, F., Dunkels, A., Eriksson, J., Finne, N., and Voigt, T. *Cross-level sensor network simulation with cooja*, Proc. Conference on Local Computer Networks, IEEE, 2006.
- [46] Dunkels, A., Eriksson, J., Finne, N., and Tsiftes, N., *Powertrace: Network-level power profiling for low-power wireless networks*, Technical Report, 2011.
- [47] Silva, R., Zinonos, Z., Sa Silva, J., and Vassiliou, V., *Mobility in WSNs for critical applications*, Proc. Symposium on Computers and Communications, IEEE, 2011.

Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>

Copyright

The publishers will keep this document online on the Internet – or its possible replacement – from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: <http://www.ep.liu.se/>.